AD-A135 648

AD A135 648

50272-101

1

REPORT DOCUMENTATION PAGE

1. REPORT NO. DOD/DF-84/002a

2.

3. Recipient's Accession No.

4. Title and Subtitle

Computer Directed Training System (CDTS),
User's Manual

5. Report Date July 1, 1983

6.

7. Author(s)

8. Performing Organization Rept. No. AF Manual 50-752

9. Performing Organization Name and Address

Department of the Air Force
Headquarters US Air Force
Washington, DC. 20330

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.

(C)

(G)

12. Sponsoring Organization Name and Address

(same as above)

13. Type of Report & Period Covered

14.

15. Supplementary Notes

For magnetic tape package, see
This document supersedes PB81-210346.

16. Abstract (Limit: 200 words)

This manual provides information about how to use the Computer Directed Training System for the H6000 series computer. The CDTS is a general purpose computer assisted instruction package that uses the capability of the computer to present instructional materials to trainees via remote terminals. The CDTS programs, among other things, control the presentation of lesson material, read and evaluate student responses, record student activities and produce an activity summary containing information of interest to local training monitors.

DEC 13 1983

A

Accession For

NTIS
$ 68.50

Dist

A | 21

17. Document Analysis a. Descriptors

Computer Directed Training
Computer Assisted Instruction
Coursewriter

b. Identifiers/Open-Ended Terms

DTIC
COPY
INSPECTED
3

c. COSATI Field/Group

18. Availability Statement:

The source agency has restricted sales of this item to federal, state and local governments.

19. Security Class (This Report)

20. Security Class (This Page)

21. No. of Pages

22. Price

(See ANSI-Z39.18)

See Instructions on Reverse

83 12 13 068

DO NOT PRINT THESE INSTRUCTIONS AS A PAGE IN A REPORT

INSTRUCTIONS

Optional Form 272, Report Documentation Page is based on Guidelines for Format and Production of Scientific and Technical Reports, ANSI Z39.18–1974 available from American National Standards Institute, 1430 Broadway, New York, New York 10018. Each separately bound report—for example, each volume in a multivolume set—shall have its unique Report Documentation Page.

1. Report Number. Each individually bound report shall carry a unique alphanumeric designation assigned by the performing organization or provided by the sponsoring organization in accordance with American National Standard ANSI Z39.23–1974, Technical Report Number (STRN). For registration of report code, contact NTIS Report Number Clearinghouse, Springfield, VA 22161. Use uppercase letters, Arabic numerals, slashes, and hyphens only, as in the following examples: FASEB/NS–75/87 and FAA/RD–75/09.
2. Leave blank.
3. Recipient's Accession Number. Reserved for use by each report recipient.
4. Title and Subtitle. Title should indicate clearly and briefly the subject coverage of the report, subordinate subtitle to the main title. When a report is prepared in more than one volume, repeat the primary title, add volume number and include subtitle for the specific volume.
5. Report Date. Each report shall carry a date indicating at least month and year. Indicate the basis on which it was selected (e.g., date of issue, date of approval, date of preparation, date published).
6. Sponsoring Agency Code. Leave blank.
7. Author(s). Give name(s) in conventional order (e.g., John R. Doe, or J. Robert Doe). List author's affiliation if it differs from the performing organization.
8. Performing Organization Report Number. Insert if performing organization wishes to assign this number.
9. Performing Organization Name and Mailing Address. Give name, street, city, state, and ZIP code. List no more than two levels of an organizational hierarchy. Display the name of the organization exactly as it should appear in Government indexes such as Government Reports Announcements & Index (GRA & I).
10. Project/Task/Work Unit Number. Use the project, task and work unit numbers under which the report was prepared.
11. Contract/Grant Number. Insert contract or grant number under which report was prepared.
12. Sponsoring Agency Name and Mailing Address. Include ZIP code. Cite main sponsors.
13. Type of Report and Period Covered. State interim, final, etc., and, if applicable, inclusive dates.
14. Performing Organization Code. Leave blank.
15. Supplementary Notes. Enter information not included elsewhere but useful, such as: Prepared in cooperation with . . . Translation of . . . Presented at conference of . . . To be published in . . . When a report is revised, include a statement whether the new report supersedes or supplements the older report.
16. Abstract. Include a brief (200 words or less) factual summary of the most significant information contained in the report. If the report contains a significant bibliography or literature survey, mention it here.
17. Document Analysis. (a). Descriptors. Select from the Thesaurus of Engineering and Scientific Terms the proper authorized terms that identify the major concept of the research and are sufficiently specific and precise to be used as index entries for cataloging.

    (b). Identifiers and Open-Ended Terms. Use identifiers for project names, code names, equipment designators, etc. Use open-ended terms written in descriptor form for those subjects for which no descriptor exists.

    (c). COSATI Field/Group. Field and Group assignments are to be taken from the 1964 COSATI Subject Category List. Since the majority of documents are multidisciplinary in nature, the primary Field/Group assignment(s) will be the specific discipline, area of human endeavor, or type of physical object. The application(s) will be cross-referenced with secondary Field/Group assignments that will follow the primary posting(s).
18. Distribution Statement. Denote public releasability, for example "Release unlimited", or limitation for reasons other than security. Cite any availability to the public, with address, order number and price, if known.

19. & 20. Security Classification. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED).

21. Number of pages. Insert the total number of pages, including introductory pages, but excluding distribution list, if any.
22. Price. Enter price in paper copy (PC) and/or microfiche (MF) if known.

☆U.S. Government Printing Office: 1978—261-647/3308

DEPARTMENT OF THE AIR FORCE
Headquarters US Air Force
Washington DC 20330

AF MANUAL 50-752

1 July 1983

DISTRIBUTION LIMITED TO DOD,
REFER OTHER REQUESTS TO THE ADPS MANAGER

Training

COMPUTER DIRECTED TRAINING SYSTEM (CDTS): E776A/VC

USERS MANUAL

This manual provides information about how to use the Computer Directed Training System (CDTS) for the H6000 series computer.

This manual is affected by the Privacy Act of 1974. Each student will be informed of this fact when logging on to this system. The student will be told that disclosure is voluntary, and will be referred to AFM 171-752, figure 2-18, for more information.

NOTE: This manual has been written to be easily understood by its primary audience. However, there are certain words that are commonly used in performing the functions described here. These are listed in the section titled "Terms and Abbreviations." We advise you to review them before reading this document.

TABLE OF CONTENTS

Supersedes AFM 50-752, 1 June 1979. (See signature page for summary of changes.)
No. of Printed Pages: 1042
OPR: KTTC (Mr J. V. Johnson)
ADPS Manager: AFDSDC/DM
Approved by: 3300 TCHTW/CC (Col H. W. Boardman)
Distribution: L

83 12 13 068

LIST OF ATTACHMENTS

SECTION 1. GENERAL

1.1 Purpose of the Users Manual. The objective of the Users Manual for the Computer Directed Training System (CDTS): E776A/VC is to provide the user's non-EDP personnel (Training System Manager, Training Monitor, and Trainee) with the information necessary to effectively use the system.

1.2 Project References.

a. MJCS-230-73, 16 July 1973, Standardization Task Assignment, Computer Directed Training System.

b. Data Project Directive (DPD) HAF-P-74-14, Computer Directed Training System (CDTS) for Honeywell 6000 series computers (H6000) Worldwide Military Command and Control System, (WWMCCS), 24 September 1974.

c. JCS Publication 14, Computer System and Program Catalog for WWMCCS.

d. JCS Publication 17, Management Procedures for the Worldwide Military Command System.

e. AFM 171-752, Automatic Data Processing Systems and Procedures, Computer Directed Training System, CDTS: 776A/VC, Computer Operations Manual, 1 June 1979.

1.3 Terms and Abbreviations.

CDT(S) - Computer Directed Training (System)
COURSEWARE - Computer prepared lessons and associated doucmentation
COURSEWRITER - Author of CDT courseware
DASH (-) - Denotes locally determined
DEVELOPMENT CENTER - A site which prepares courseware
DPI - Data Processing Installation
LESSON - A logical segment of a subject matter course
TM - Training Monitor
TSM - Training System Manager
TTY - Remote Teletype
UMC - User Master Catalog
VIP - Visual Information Projection

1.4 Security and Privacy.

a. This system maintains records under provisions of 10 USC 8012 and Executive Order 9397, 22 November 1943.

b. Anyone logging onto the CDT System will be advised that input of their Social Security Account Number is voluntary and will be referred to AFM 50-752, Attachment 2, for more information.

c. Disposal of reports covered by the Privacy Act of 1974 may be accomplished by turning them over to the local Defense Property Disposal Office.

1.5 Additional Required Materials. As far as practical, all off-line information necessary for the successful completion of CDT lessons will be available in the applicable attachment to AFM 50-752. Additional recommended reference materials must be acquired by the user. Assistance in locating this material can be obtained from the TSM/TM or Attachment 2 of AFM 50-752.

SECTION 2. SYSTEM SUMMARY

2.1 System Application. The Computer Directed Training System is used to prepare and present lessons that supplement local on-the-job training programs. The system utilizes Computer Assisted Instruction (CAI) techniques and provides some of the features which have long been desirable for training:

a. Course material is tailored to the needs of the individual trainee.

b. Training is accomplished at the trainee's convenience without long periods away from the job.

2.2 System Operation. Any trainee having a need for any existing CDTS lesson is provided the capability to take that lesson via a remote teletype (TTY) or visual information projection (VIP) system. The trainees' progress through lessons are maintained during lesson execution. In this way, trainees are able to stop in the middle of a lesson; when they wish to resume training, they are able to continue from the point at which they stopped.

2.3 System Configuration. The CDT System may be processed on any Honeywell 6000 series computer and on compatible GE 600 series computers. It may use any of the standard Honeywell terminals (Teletype Models 33, 35, and 37 and VIP Models 786 and 7700 series) or any other H6000 compatible terminal.

2.4 System Organization.

2.4.1 CDTS Courseware Load. The CDTS release tapes are FILSYS user save tapes of User Master Catalog (UMC) CDTSCDTS and CDTSCDTS/DISK.THREE. The entire CDT system can be loaded by doing a FILSYS RESTORE activity for UMC CDTSCDTS and another FILSYS RESTORE activity for CDTSCDTS/DISK.THREE. A selective restore is also possible if only certain courses, data files, or an object program library are needed. Reference Honeywell's File Management Supervisor Manual and Attachment 1 of this manual for more information.

2.4.2 CDTS Files Creation. The following files must be created by FILSYS to allow normal processing of the system.

a. Student File--Unique file for each student. Contains information about the student and the lesson being taken.

b. Student Record Accumulator File--FVCXASJ3U. Information about the lesson a student takes is copied from the student file to this file.

c. Executive Object File--CMDLIB/CDTS (VCXASJ). This file is located in the Time-sharing Command Library. It contains an executable (H*) version of the CDT Executive Program.

2.4.3 CDTS Executive (VCXASJ). The CDTS Executive runs as a subsystem of time-sharing. It controls the presentation of lesson material to students, builds temporary files of student records, and spawns a locally created batch program to update the Student Record Accumulator File. The Executive is loaded in the Time-sharing Command Library. It is placed into execution by entering CDTS in response to the time-sharing prompt * .

2.4.4 CDTS Student Records Summary (VCSRSJ). This program produces a report of student activity on CDTS and maintains a file of student responses to lesson material. This file (FVCSRSJTU) is called a Trace Tape and must be sent to the CDT section each quarter. Reference Attachment 5 of this manual.

2.4.5 CDTS Compiler (VCCMSJ). This program converts the source program into an object file which the Utility Program can load into a lesson file. This program is for use by those sites which prepare site-unique lessons using the CDT Coursewriter Course. Reference Attachment 8 of this manual.

2.4.6 CDTS Utility Program (VCUTSJ). This program is used to load and delete object lessons from lesson files. Reference Attachment 8 of this manual.

2.4.7 CDTS Development Center Programs. These programs are available for those sites which write their own CDT lessons. They allow the site to gather statistical and course validation data. Reference Attachment 9, paragraph 8, of this manual.

2.4.8 Hard Copies of CDTS Text Material. This program allows Training Monitors to print copies of text material of CDT lessons. Refer to Attachment 7 for instructions on using this program.

2.5 Performance. The CDT System is run as a subsystem of time-sharing. Each user uses approximately 0.6 percent of the available time-sharing resources or about 0.2 percent of the total available resources of the computer (based on H6060 computer). The students may use any type of standard or compatible remote terminal. Each student must have a unique student file of at least two random blocks. The number of lessons in which a student may be enrolled is equal to the number of blocks (LLINKS) in the student file minus the one block reserved for student identification data.

2.6 Data Base.

2.6.1 Lesson Material. All lessons described herein are developed or controlled by Air Training Command and released through the Air Force Data Systems Design Center. A FILSYS user save tape containing all current lessons, data files, hard copy files, and object program libraries will be maintained by Data Automation at each site.

2.6.2 UMC Creation. The tape mentioned in paragraph 2.6.1 is a user save tape of UMC CDTSCDTS. Prior to loading the CDT system at a site for the first time, a System Master Catalog (SMC) entry entitled CDTSCDTS must be created with the following parameters: size of 700 links with CARDIN permission. For system integrity, this UMC is designed for READ ONLY permission.

2.6.3 UMC Contents. The UMC structure as shown on the next page is accomplished by a FILSYS RESTORE. Catalog/file strings are provided for student access to courses, data files, and data bases.

UMC SUBCAT FILE	FILE SIZE/TYPE	FILE CONTENT
CDTSCDTS		
RSTARLIB	110 blocks/SEQ	Object Program Library containing the following four programs: VCXASJ - Controls lesson material presentation to students. VCUTSJ - Obtains lessons from courseware master file. VCCMSJ - This is the CDTS lesson compiler. VCSRSJ - Student Record Summary program.
RSTARTWO	210 blocks/SEQ	Object Program Library containing the following Development Center programs: VCCØSJ VCS3SJ VCT3SJ VCCRSJ VCS5SJ VCT4SJ VCSØSJ VCSMSJ VCT5SJ VCS1SJ VCTØSJ VCTSSJ VCS2SJ VCT2SJ VCTTSJ
CDTSCDTS		
CBL		
LESSONS	380 blocks/RAN	Lesson file of the 16 COBOL lessons.
CAI001	2 blocks/SEQ	File for COBOL lesson CBL120.
CAI002	6 blocks/RAN	Data file for COBOL lesson CBL120.
CDTSCDTS		
CDT		
LESSONS	100 blocks/RAN	Lesson file of the 6 CDT Coursewriter lessons.
SKELETON	25 blocks/SEQ	Skeleton file used to create a CDT Course lesson.
CDTSCDTS		
FTN		
LESSONS	400 blocks/RAN	Lesson file of the 12 FORTRAN lessons.
CDTSCDTS		
GIP		
LESSONS	250 blocks/RAN	Lesson file of the 12 GIP lessons.
CDTSCDTS		
GMP		
LESSONS	600 blocks/RAN	Lesson file of the 22 GMAP lessons.

UMC SUBCAT FILE	FILE SIZE/TYPE	FILE CONTENT
CDTSCDTS		
IDP		
LESSONS	110 blocks/RAN	Lesson file of the 4 IDP lessons.
CDTSCDTS		
IDS		
LESSONS	380 blocks/RAN	Lesson file of the 14 IDS lessons.
LOADDATA	24 blocks/SEQ	Data file for the lesson IDS043.
CDTSCDTS		
INT		
LESSONS	225 blocks/RAN	Lesson file of the 6 INTRO H6000 lessons.
COBOL061	4 blocks/SEQ	COBOL program for lesson INT061.
DATA061	1 block/SEQ	Data file for lesson INT061.
TAPE1	10 blocks/SEQ	Data file for lessons INT070 and INT080.
TAPE2	10 blocks/SEQ	Data file for lesson INT070.
CDTSCDTS		
ISP		
LESSONS	125 blocks/RAN	Lesson file of the 5 Indexed Sequential Processor Lessons.
DATA020	5 blocks/SEQ	Data file for lesson ISP020.
DATA030	1 block/SEQ	Data file for lesson ISP030.
DATA040	1 block/SEQ	Data file for lesson ISP040.
CDTSCDTS		
JCL		
LOAD	(2 blocks/SEQ)	JCL for Programs VCCMSJ and VCUTSJ.
LOADCDTS	(1 block/SEQ	JCL to load the CDT Executive VCXASJ) to keep records.
RUN-SR	(1 Block/SEQ)	JCL for Program VCSRSJ.

2.7 General Description of Inputs, Processing, Outputs.

2.7.1 Inputs.

a. Executive: All input to the Executive Program is in the form of student responses to questions initiated by the Executive.

b. Student Records Summary: Input to the Student Summary Program consists of a consolidated file of student records (Accumulator File--FVCXASJ3U) output by the CDTS Executive, and the last cycle tape (Trace Tape--FVCSRSJTU) output from the Student Records Summary Program.

c. Compiler: Input to the CDTS Compiler is a source lesson file prepared by a coursewriter.

d. Utility Program: Input to the Utility Program consists of a Utility Control Cards file (CVCUTSJCU--PCN: SE776A001) and a lesson Object File (FVCCMSJOU) located on a disk or a tape.

2.7.2 Processing.

a. Executive: Reference Attachments 3.

b. Student Records Summary: Reference Attachments 4 and 5 of this manual.

c. Compiler: Reference Attachment 8 of this manual.

d. Utility Program: Reference Attachment 8 of this manual.

2.7.3 Outputs.

a. Executive: All output from the Executive Program consists of lesson material presented to the student, information/error messages to the student, and a record of the student's activity during the time-sharing session (Accumulator File--FVCXASJ3U).

b. Student Record Summary: Output from the Student Record Summary consists of a report of student activity (PCN: SE776A003) since the last report and a file tracing student responses to lesson material for courseware validation. The file produced is in the form of a Trace Tape. Trace Tape instructions are found in Attachment 5 of this manual.

c. Compiler: The CDTS compiler produces a diagnostic source listing (PCN: SE776A002) showing the input source lesson and any applicable error messages. If the CDTS Compiler finds no errors, an object file is produced which will be used by the Utility Program to load the lesson to a lesson file. The Compiler is only used for generating site unique lessons.

d. Utility Program: Outputs from the Utility Program consist of a Lesson File Inventory produced when the lesson output medium is disk and a file containing lesson data in either tape or disk format. The primary output option is a Lesson File in a format usable by the Executive. The Utility Program is only used on site unique lessons.

2.8 Field Assistance. All inquiries and feedback on CDTLBS should be forwarded to the data automation's system monitor for assistance in determining the need for the preparation of a Difficulty Report (DIREP) and/or Incident Report.

SECTION 3. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

3.1 Initiation Procedures. See Attachment 13 of this manual.

3.2 Staff Input Requirements.

3.2.1 Input Formats.

a. Student Record Summary: Input formats for the Student Record Summary are automatically generated by the CDTS Executive or a previous execution of the Summary Program.

b. Compiler: Input formats (language specifications) for the compiler are documented in the CDTS Coursewriter Course (CDT) and Attachment 8 of this manual.

c. Utility Program: The format for program control cards for the Utility Program (VCUTSJ) is documented in Attachment 8.

3.2.2 Composition Rules. NA

3.2.3 Input Vocabulary. NA

3.2.4 Sample Inputs. See Attachments 3, 5, 7 and 8.

3.3 Output Requirements.

3.3.1 Output Formats. NA

3.3.2 Sample Outputs. See Attachments 5 and 6.

3.3.3 Output Vocabulary. NA

3.4 Utilization of System Outputs. See Attachment 2.

3.5 Recovery and Error Correction Procedures.

a. Error Messages: See Attachment 4.

b. Incident Reporting: Incident reporting for this system will be performed in accordance with JCS Pub 17, section 3, and annex F, chapter III. Two copies of the incident report will be forwarded to:

AFDSDC/SCCA
Gunter AFS, AL 36114

For errors in lesson content, contact the Keesler CDT Section by phoning AUTOVON 868-2985, commercial (601) 377-2985, or write to:

3300 TCHTW/TTGXZ
Attn: WWMCCS CDT
Keesler AFB, MS 39534

3.6 Training System Manager (TSM). Each site will have an individual designated to perform this additional duty. The duties of the TSM are described in Attachment 2 of this manual.

3.7 Training Monitor (TM). The TSM at each site will appoint technically qualified individuals for the additional duty of training monitor (TM). (These appointments may be by CDTS course, functional area, work center, or any other division considered appropriate by the TSM.) The duties of the TM are described in Attachment 2 of this manual.

SECTION 4. CDTS COURSE DESCRIPTIONS

4.1 System Query Capabilities. The H6000 Computer Directed Training System (CDTS) is a general-purpose Computer Directed Training (CDT) system that uses the capabilities of the computer to present instructional materials to trainees who interact with the computer via remote terminals. The CDTS system presents the requested lesson material at the remote terminal for individual online training sessions. The computer acts as a medium for the presentation of expertly prepared instruction to the trainee, who reads the material and responds via the terminal when questions are asked. The computer constantly evaluates these responses and, based on how well training objectives are being met, selects appropriate material for the next presentation--a sort of tutor-computer arrangement. When applicable, the instructional materials presented by the system are augmented by off-line exhibits and work projects. The application of the computer to training offers many advantages to the user. The system will accept a wide variety of material from course designers and retain it indefinitely. This material can be presented to many trainees simultaneously, with each progressing at their own individual rates. The trainee uses job-related equipment and can remain on the job during the training period. When not working, the trainee can be learning and vice versa. Instruction is tailored to the needs of each individual; if the trainee must leave unexpectedly or training time expires, CDTS will retain the position in the lesson. When the trainee returns, the system will start the instruction exactly where the training was previously interrupted. However, the conscientious scheduling of lessons, trainees, and remote terminals is critical to efficient operation.

4.2 Data Base Format. The instructional materials that are provided through CDTS are divided into courses, and these courses are divided into lessons. Each lesson is further subdivided into frames. The H6000 CDT system currently consists of 19 courses.

(1) A 12 lesson Terminal User (H6000-CDT) course.

(2) A 16 lesson COBOL programming (H6000-CDT) course.

(3) A 12 lesson FORTRAN programming (H6000-CDT) course.

(4) A 33 lesson Joint Operational Planning System (JOPS) III User Programming (H6000-CDT) course.

(5) A six lesson Introduction to the H6000 Series Computer Programming (H6000-CDT) course.

(6) A 20 lesson Integrated Data Store (IDS) (H6000-CDT) course.

(7) A 13 lesson H6000 Computer Operator (H6000-CDT) course.

(8) A six lesson Computer Directed Training System (CDTS) Coursewriter (H6000-CDT) course.

(9) A 10 lesson PDP-11 Computer Operator (H6000-CDT) course.

(10) A seven lesson Selective Inquiry System (SIS) (H6000-CDT) course.

(11) A three lesson H6000 Total On-Line Test System (TOLTS) (H6000-CDT) course.

(12) An eight lesson Management Data Query (MDQS) Basic User (H6000-CDT) course.

(13) A five lesson Indexed Sequential Processor (ISP) (H6000-CDT) course.

(14) A 22 lesson General Macro Assembly Programming (GMAP) (H6000-CDT) course.

(15) A one lesson H700 Test and Diagnostics (H6000-CDT) course.

(16) A five lesson Introduction to Data Processing (H6000-CDT) course.

(17) A 15 lesson WWMCCS Intercomputer Network (WIN) User (H6000-CDT) course.

(18) A 13 lesson Graphics Information Presentation System (GIPSY) (H6000-CDT) course.

(19) A six lesson Basic Language Programming (BSC) (H6000-CDT) course.

Each lesson is given a unique six-character alphanumeric identifier; for example, the first lesson in the Terminal User course is TUC010.

4.2.1 Terminal User (H6000-CDT) Course.

4.2.1.1 Course Number, Title, and PDS Code. This course is number ECDTS5116 000; it is titled "Terminal User (H6000-CDT)"; and it bears PDS Code S7M.

4.2.1.2 Purpose. The Terminal User course is designed to provide WWMCCS staff personnel with an introduction to the H6000 time-sharing system. The lessons should be used to instruct those who have limited automatic data processing experience and those who are unfamiliar with the time-sharing system.

4.2.1.3 Terminal User (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the Terminal User course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB MS 39534.

4.2.1.4 Prerequisites. None.

4.2.1.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 13 of this manual. The trainee will be directed by the lesson to the appropriate exhibit. Prior to the start of lesson TUC010, the trainee should study Exhibits TUC010-1 through TUC010-4; prior to the start of lesson TUC011 the trainee should study Exhibits TUC011-1 through TUC011-4; and prior to the start of lesson TUC012, the trainee should study Exhibits TUC012-1 through TUC012-4. Trainees in the first lesson of the Terminal User course will also be required to use Attachment 5, CDTS Executive Error Messages.

4.2.1.6 Pattern of the Course. The Terminal User course consists of at least six lessons which should be executed consecutively:

(1) TUC010—Operation of the Remote Teletype Terminal
TUC011—Operation of the VIP786 Terminal
TUC012—Operation of the VIP7705 Terminal
TUC013—Operation of the LA36/LA35 Decwriter II
TUC014—Operation of the Execuport 300
(2) TUC020—Introduction to the Time-Sharing System
(3) TUC030—File System Security and Access
(4) TUC035—Fundamentals of File System Software (Optional Lesson)
(5) TUC040—CARDIN and Conversion Subsystems
(6) TUC041—File Maintenance Subsystems
(7) TUC050—Text Editor
(8) TUC060—RUNOFF (Optional Lesson)

The instruction in this source is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Twenty-two (22) hours are used for course length and diploma credit. For course credit, the trainee must successfully complete either TUC010, TUC011, TUC012, TUC013, or TUC014; trainee must also complete TUC020, TUC030, and TUC040 through TUC050.

4.2.1.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.1.8 Lessons Available. Following is a list of available lessons, together with an estimate of the time required for completion.

a. TUC010. Instruction in this lesson focuses on two major subject matter areas: (1) the operation of the remote teletype terminal, to include turning the device on and off, correcting typographical errors, cancelling a line of type prior to transmission, and logging on/off; and (2) interaction with the Computer Directed Training System (CDTS) to include how to interact with the lesson, terminate a training session, and interpret CDTS messages to a remote terminal. NOTE: This lesson should be executed on the teletype terminal.

TIME: Approximately 1.0 hour of on-line time is required to complete TUC010.

b. TUC011. This lesson teaches operation of the VIP786. This lesson also focuses on two major subject matter areas: (1) operation of the device, to include turning the VIP on and off, correcting typographical errors, cancelling messages prior to transmission, and logging on/off; and (2) interaction with the Computer Directed Training System (CDTS), to include how to access a CDT lesson, terminate a training session, and interpret CDTS messages to the remove device. NOTE: This lesson should be executed on the VIP786.

TIME: Approximately 1.0 hour of on-line time is required to complete TUC011.

c. TUC012. This lesson teaches operation of the VIP7705, one of the Honeywell 7700 series Visual Information Projection (VIP) devices. This lesson focuses on two major subject matter areas: (1) operation of the device—that is, turning it on/off, correcting typographical errors or cancelling messages, and logging on/off; and (2) interaction with the Computer

Directed Training System (CDTS), including how to get a lesson, interact with a lesson, and terminate a training session. NOTE: This lesson should be executed on the VIP7705 or other 7700 series VIP.

TIME: Approximately 1.0 hour of on-line time is required to complete TUC012.

d. TUC013. This optional lesson teaches operation of the LA36/LA35 Decwriter II. This lesson focuses on two major subject matter areas: (1) operation of the device, and (2) interaction with the Computer Directed Training System(CDTS), including how to get a lesson, interaction with a lesson, and terminate a training session. NOTE: This lesson should be executed on the Decwriter.

TIME: Approximately 1.0 hour of on-line time is required to complete TUC013.

e. TUC014. This optional lesson teaches the operation of the Executport 300. This lesson focuses on two major subject matter areas: (1) operation of the device, and (2) interaction with the Computer Directed Training System (CDTS), including how to get a lesson, interact with a lesson, and terminate a training session. NOTE: This lesson should be executed on the Execuport.

TIME: Approximately 1.0 hour of on-line time is required to complete TUC014.

f. TUC020. This is the introductory lesson to the time-sharing system (TS1). It is designed to familiarize the trainee with the Time-sharing Executive and its relationship with various subsystems. It provides instruction on how the command language controls the operational time-sharing subsystems, and it identifies those commands given in response to questions from the system and those that are initiated by the user.

TIME: Approximately 2.0 hours of on-line time are required to complete TUC020.

g. TUC030. This lesson includes instruction pertaining to file system security and access. It covers the construction of the User Master Catalog and sub-catalogs and files created by the user. It gives instruction on assigning passwords and giving permissions to other users. This lesson also explains the ACCESS subsystem and its functions. The trainee will use the ACCESS subsystem to create and modify a catalog and a file.

TIME: Approximately 3.0 hours of on-line time are required to complete TUC030.

h. TUC035. This optional lesson teaches the fundamentals of the file system software. The trainee will learn rudiments of catalog and file creation. The subject matter is divided into three areas -- that is, (1) user communication media (how to communicate with the file system), (2) file structure (files and catalogs), and (3) file system security (catalog/file strings, password, and permissions).

TIME: Approximately 2.0 hours of on-line time are required to complete TUC035.

i. TUC040. This lesson instructs the trainee on use of the CARDIN and Conversion subsystems to create a card image file. It teaches the capabilities of the terminal/batch interface facility which allows the user to exercise conversion techniques such as ASCII time-sharing format to BCD. The associated command language is also taught.

TIME: Approximately 2.0 hours of on-line time are required to complete TUC040.

j. TUC041. This lesson, titled "File Manipulation Subsystems," teaches the BPUNCH, BPRINT, RUN, and PRINT commands; it also teaches BCDASC and ASCBCD conversions. A simulation of the dialogue within the file manipulation subsystems is used to develop the knowledges and skills needed to effectively use the various commands. During the lesson, the trainee is required to demonstrate an ability to print, punch, convert, and execute a disk file.

TIME: Approximately 9.0 hours of on-line time are required to complete TUC041.

k. TUC050. This lesson is designed to instruct the trainee on the Text Editor subsystem, a major feature of the time-sharing system. The trainee will learn how files consisting of text are entered, edited, stored, and retrieved. Numerous time-sharing editing commands are also covered in this lesson. This lesson assigns PROJECT 1.

TIME: Approximately 3.5 hours of on-line and 1.5 hours of off-line time are required to complete TUC050.

l. TUC060. The RUNOFF subsystem is the subject of this optional lesson. It shows how this subsystem works in conjunction with the Editor subsystem to permit the user to specify the format in which text is to be reproduced. The trainee learns how to specify the length of each line of text, the number of lines on a page, and the size of page margins. RUNOFF commands and control words are also included in this lesson. This lesson assigns an optional project.

TIME: Approximately 2.0 hours of on-line time are required to complete TUC060.

4.2.2 COBOL Programming (H6000-CDT) Course.

4.2.2.1 Course Number, Title, and PDS Code. This course is numbered ECDTS51131B 001; it is titled "COBOL Programming (H6000-CDT)"; and it bears PDS Code SZ8.

4.2.2.2 Purpose. This course is designed for use by computer programmers and systems analysts working in a Honeywell Series 6000 computer environment. The course can be used to provide a working knowledge of COBOL-68 to trainees who are totally unfamiliar with the language, or it can be used by COBOL-experienced programmers who wish to make a transition to the H6000 Series computer.

4.2.2.3 COBOL Programming (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the COBOL Programming course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.2.4 Prerequisites. In terms of training, the prerequisites of this course are successful completion of Introduction to Data Processing or equivalent training and a knowledge of applicable control card formats. Prior to starting this course, the trainee should also complete either lesson TUC010, TUC011, or TUC012.

4.2.2.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 14 of this manual. The trainee will be directed by the computer to the appropriate exhibits for each lesson.

4.2.2.6 Pattern of the Course. The COBOL Programming course consists of 16 lessons which should be executed consecutively:

(1) CBL010—Introduction to COBOL
(2) CBL020—Basic Grammar of COBOL, Part 1
(3) CBL030—Basic Grammar of COBOL, Part 2
(4) CBL040—Identification and Environment Divisions
(5) CBL050—Basic Options of the Data Division
(6) CBL060—Working Storage Section and Redefinition
(7) CBL070—Procedure Division and Basic Options
(8) CBL080—Additional Options (Data and Procedure Divisions)
(9) CBL090—Decision Instructions (IF Verb)
(10) CBL100—Processing Magnetic Tape
(11) CBL110—Perform Verb
(12) CBL120—Processing Magnetic Disk
(13) CBL130—Tables, Occures Clause, Subscripts, and Perform Varying
(14) CBL140—Sort and Merge Options
(15) CBL150—Report Writer (optional lesson)
(16) CBL160—Converting from COBOL-68 to COBOL-74 (optional lesson)

The instruction in this course is self-paced, actual completion time may vary. However all lessons and lesson projects not indicated as optional are required. Forty (40) hours are used for course length and diploma credit.

4.2.2.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.2.8 Lessons Available. Following is a list of the available COBOL-68 lessons, together with an estimate of the time required for completion.

a. CBL010. This lesson, titled "Introduction to COBOL," provides a general overview of the COBOL system. The trainee is exposed to the functions of the COBOL compiler, including object program generation and validity (syntax) checking. The programmer is introduced to the COBOL coding form and the associated coding formats. The lesson touches on the functions of the four divisions that comprise a COBOL program. After completing this lesson, the trainee should be able to describe the functions of the COBOL compiler, identify the six areas of the COBOL coding form, and name the COBOL compiler division that performs each function.

TIME: Approximately 1.5 hours of on-line time are required to complete CBL010.

b. CBL020. This lesson focuses on the basic grammar of COBOL. It is the first half of a two-part lesson. The trainee will learn about the formation of COBOL words (reserved words and programmer-supplied words), including data names and procedure names. The formation and use of numeric and alphanumeric literals and figurative constants are also discussed.

TIME: Approximately 3.0 hours of on-line time are required to complete CBL020.

c. CBL030. This is a continuation of CBL020, the basic grammar of COBOL. Two major subjects are taught--(1) the use of sentences and paragraphs and (2) the rules of utilizing COBOL formats. Instruction includes definition of syntax, required words, optional words, programmer-supplied entries, braces, brackets, and ellipses. Given a list of grammar elements, syntax format elements, and a list of descriptors, the trainee should be able to match each element with its description upon completion of this lesson.

TIME: Approximately 1.5 hours of on-line time are required to complete CBL030.

d. CBL040. This lesson develops knowledge and skills relative to the Identification and Environment Divisions. After completing this lesson, the trainee, when given a COBOL Reference Manual and a narrative description of the required documentation entries, should be able to encode statements in the Identification and Environment Divisions. The following subjects are covered in the Identification Division: (1) division header and (2) paragraphs (program ID, author, installation, date written, date compiled, security, and remarks). The following three subjects are covered in the Environment Division: (1) division header, (2) configuration section, and (3) input-output section.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL040.

e. CBL050. This lesson is designed to teach the basic options of the Data Division. File description (FD file name, label records clause, and value of ID) is taught, together with the record description (01 record name, level number concept, data name, filler, and picture clause). Upon completing this lesson, the trainee should be able to encode statements required for the Data Division of a COBOL program.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL050.

f. CBL060. Instruction in this lesson centers around the working-storage section and redefinition. The following subjects are discussed: level entries (independent items and records), clauses, and both implicit and explicit redefinition. The trainee is taught to code the value and usage clauses that apply to problem parameters. To successfully complete the lesson, the trainee must correctly complete unfinished statements pertaining to the working-storage area and redefinition. This lesson assigns COBOL

PROJECT 1.

TIME: Approximately 2.0 hours of on-line and 3.0 hours off-line time are required to complete CBL060.

g. CBL070. Upon completion of this lesson, the trainee will be able to encode the Procedure Division. The trainee is asked to identify true statements pertaining to the structure of the division, and is asked to match verbs (open, read, move, write, go to, close, stop, run) with their functions. In summary, the trainee is taught the structure of the Procedure Division and the applicable verbs. This lesson assigns COBOL PROJECT 2.

TIME: Approximately 2.5 hours of on-line and 3.0 hours off-line time are required to complete CBL070.

h. CBL080. This lesson is titled "Additional Options of the Data and Procedure Divisions," and it is a sequel to CBL070. Two subjects are taught: editing characters in the picture clause and arithmetic verbs (add, subtract, multiply, divide, and compute). The trainee will encode picture clauses and arithmetic instructions to solve a variety of problems.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL080.

i. CBL090. Within this lesson on decision instruction, two subjects are taught: simple conditions (relational tests, class tests, and condition-name tests) and compound conditions (AND/OR conditions). The trainee is given a program segment that contains decision instructions plus a list of data parameters, and precisely determines the final result of selected instructions.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL090.

j. CBL100. This lesson is titled "Processing Magnetic Tape." The first half of this lesson explores the characteristics of magnetic tape; the trainee is exposed to recording features, blocked records, labels, and file organization. The last half of the lesson teaches various clauses and several verbs (alter, open, close, and copy). This lesson assigns an OPTIONAL PROJECT 3.

TIME: Approximately 1.5 hours of on-line time are required to complete CBL100.

k. CBL110. This lesson covers the basic, times, and until options of the PERFORM verb. The trainee needs the exhibits in Attachment 14 to complete this lesson. During this lesson, the trainee is shown segments of COBOL programs that utilize PERFORM instructions, and is required to determine the results of their execution.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL110.

l. CBL120. This lesson on processing by use of the magnetic disk subsystem begins with a discussion of magnetic disk characteristics; later the trainee is introduced to the control cards and program requirements for disk file space. There is a review of format notation; and coverage of environment division statements, data division statements, and procedure division statements. After completing this lesson, the trainee can program a solution to a problem that requires magnetic disk application. This lesson assigns COBOL PROJECT 4.

TIME: Approximately 2.0 hours of on-line and 3.0 hours off-line time are required to complete CBL120.

m. CBL130. CBL130 focuses on tables and table usage. Table creation--both one-dimensional and two-dimensional--is covered, together with table access using the perform verb (varying option).

TIME: Approximately 2.0 hours of on-line time are required to complete CBL130.

n. CBL140. This lesson defines the terms SORT and MERGE and explains the capabilities of these two options. Included is instruction pertaining to Environment Division requirements (file control and I/O control), Data Division requirements (fixed and variable length records), and Procedure Division requirements. The trainee is taught the unique coding required in each division, and learns the steps required in the development of a sort or merge program. Finally, the trainee must design, code, and debug a sort program. This lesson assigns COBOL PROJECT 5.

TIME: Approximately 2.0 hours of on-line and 3.0 hours off-line time are required to complete CBL140.

o. CBL150. This optional lesson is titled "Report Writer." Here the trainee learns the procedure verbs which control the production of a report. Instruction centers around definition of the various report groups (report heading, page heading, report footing, overflow footing, etc.). The role of the compiler, together with the Report Writer's table of constraints, is also discussed. To successfully complete the lesson, the trainee is required to solve a problem that uses Report Writer for the production of a report. The compilation, execution, format, data, and solution must be error-free. This lesson assigns OPTIONAL COBOL PROJECT 6.

TIME: Approximately 2.0 hours of on-line and 3.0 hours off-line time are required to complete CBL150.

p. CBL160. This optional lesson entitled "Converting from COBOL-68 to COBOL-74" is an aid to experienced programmers who are converting from the COBOL-68 compiler to the COBOL-74 compiler. This lesson provides a cross-reference between the Honeywell COBOL-68 User's Guide, the major changes in each of the five COBOL divisions, and information about the Honeywell optional Control Division which has been added to the compiler.

TIME: Approximately 2.0 hours of on-line time are required to complete CBL160.

4.2.3 FORTRAN Programming (H6000-CDT) Course.

4.2.3.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5131B 000; it is titled "FORTRAN Programming (H6000-CDT)"; and it bears PDS Code SZ9.

4.2.3.2 Purpose. This course is designed for use by computer programmers and systems analysts working in a Honeywell Series 6000 computer environment. The course can be used to provide a working knowledge of FORTRAN to trainees who are totally unfamiliar with the language, or it can be used by FORTRAN-experienced programmers who wish to make a transition to the H6000 Series computer.

4.2.3.3 FORTRAN Programming (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the FORTRAN Programming course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.3.4 Prerequisites. In terms of training, the prerequisites for this course are successful completion of Introduction to Data Processing or equivalent training to include system software and file systems concepts and knowledge of applicable control card formats. Prior to starting this course, the trainee should also complete either lesson TUC010, TUC011, or TUC012.

4.2.3.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 15 of this manual. The trainee will be directed by the computer to the appropriate exhibits for each lesson.

4.2.3.6 Pattern of the Course. The FORTRAN Programming course consists of 12 lessons which should be executed consecutively:

(1) FTN010--Introduction to FORTRAN
(2) FTN020--Basic Input-Output Statements
(3) FTN030--Computations
(4) FTN040--Additional Mode Types
(5) FTN041--Arithmetic Statements and Expressions
(6) FTN042--Additional Input-Output Information
(7) FTN050--Conditional Statements
(8) FTN060--DO Loops and Arrays
(9) FTN070--Subprograms
(10) FTN080--Subroutines
(11) FTN090--Additional Input-Output Media
(12) FTN100--Time-Sharing FORTRAN (Optional)

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Forty (40) hours are used for course length and diploma credit.

4.2.3.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.3.8 Lessons Available. Following is a list of the available lessons, together with an estimate of the time required for completion.

a. FTN010. This lesson, titled "Introduction to FORTRAN," provides a general overview of the FORTRAN programming language. The trainee is exposed to the purpose of FORTRAN, a brief history, and then a general overview of the FORTRAN system itself. This overview includes the FORTRAN character set, definitions of a source program and an object program, general types of FORTRAN statements, and the FORTRAN coding form. This lesson assigns FORTRAN PROJECT 1.

TIME: Approximately 1.5 hours of on-line and .5 hour off-line time are required to complete FTN010.

b. FTN020. This lesson focuses on the integer mode of data and introduces the concept of variables and constants. It goes on to explain input and output of this type of data. Upon completion of this lesson, the trainee should try to run one of the examples illustrated in the lesson. This lesson assigns FORTRAN PROJECT 2.

TIME: Approximately 2.5 hours of on-line and 1.0 hour off-line time are required to complete FTN020.

c. FTN030. In this lesson, the trainee is exposed to usage of the integer mode in different types of assignment statements, arithmetic operators, and the hierarchy of these operators. This lesson assigns FORTRAN PROJECT 3.

TIME: Approximately 2.0 hours of on-line and 1.0 hour off-line time are required to complete FTN030.

d. FTN040. In this lesson, the trainee is exposed to the five additional types of modes that data may have in the FORTRAN programming language. These modes include: real, complex, double precision, alphanumeric, and octal. This lesson assigns FORTRAN PROJECT 4.

TIME: Approximately 3.0 hours of on-line and 1.5 hour off-line time are required to complete FTN040.

e. FTN041. In FTN041, the trainee is taught to use some of the additional modes of data, and learns when these types of data may be mixed in arithmetic expressions. The lesson also discusses what the programmer must be aware of so that his programs will be logically correct in computing answers. This lesson assigns FORTRAN PROJECT 5.

TIME: Approximately 2.0 hours of on-line and 1.5 hours off-line time are required to complete FTN041.

f. FTN042. This lesson takes the trainee a big step further in explaining how to manipulate data other than that in the integer mode. It explains each of the different types of formal descriptors for each mode of data and discusses an extra carriage control feature. This lesson assigns FORTRAN PROJECT 6.

TIME: Approximately 2.0 hours of on-line and 1.5 hours off-line time are required to complete FTN042.

g. FTN050. In this lesson, the trainee is introduced to the entire set of conditional type statements, including: logical IF, arithmetic IF, computed GO TO, and assigned GO TO. The trainee is also introduced to the data statement which allows the use of different modes of data. This lesson assigns FORTRAN PROJECT 7.

TIME: Approximately 3.0 hours of on-line and 2.0 hours off-line time are required to complete FTN050.

h. FTN060. This lesson exposes the trainee to DO loops and arrays. In this lesson, the trainee is shown how to loop through a series of executable statements a certain number of times without coding the loop. Also, arrays are discussed in some detail and the trainee is shown how to reference each element of the array. This lesson assigns FORTRAN PROJECT 8.

TIME: Approximately 3.0 hours of on-line and 2.0 hours off-line time are required to complete FTN060.

i. FTN070. This lesson explains the use of intrinsic functions, statement functions, and types of external functions. The trainee learns how to use these functions and parameters for the subprograms.

TIME: Approximately 2.0 hours of on-line time are required to complete FTN070.

j. FTN080. In this lesson, the trainee is taken one step further and is shown subroutines that are a form of subprograms. The trainee is exposed to the CALL statement, the COMMON statement, the EQUIVALENCE statement, and the RETURN statement. This lesson assigns FORTRAN PROJECT 9.

TIME: Approximately 2.5 hours of on-line and 2.0 hours off-line time are required to complete FTN080.

k. FTN090. This lesson explains the use of additional input and output media. It covers unformatted I/O and formatted I/O, FLGEOF Subroutine, FLGERR Subroutine, and link-overlay batch mode. This lesson assigns FORTRAN PROJECT 10.

TIME: Approximately 2.0 hours of on-line and 2.0 hours off-line time are required to complete FTN090.

l. FTN100. This lesson includes instruction on the general time-sharing commands used in YFORTRAN, formatting FORTRAN statements in YFORTRAN, the RUN command for YFORTRAN, link-overlay in time-sharing FORTRAN (YFORT), and the RUNL command.

TIME: Approximately 2.0 hours of on-line time are required to complete FTN100.

4.2.4 Joint Operation Planning System (JOPS) III User Programming (H6000-CDT) Course.

4.2.4.1 Course Number, Title, and PDS Code. Thir course is numbered ECDTS5116 001; it is titled "Joint Operation Planning System (JOPS) III User (H6000-CDT)"; and it bears PDS Code T4U.

4.2.4.2 Purpose. This course is designed for planners who will be using the Joint Operation Planning System (JOPS) for OPLAN development. Through the use of simulation, it provides practice in the use of the various modules within JOPS. Trainees will learn how to interact with the automated portion of JOPS. They will have the opportunity to interact with simulations of JOPS III (ADP support) to develop a hypothetical OPLAN. The course can be used to establish a working knowledge of JOPS III; however, it can also be used as a refresher for those who already know JOPS III but wish to practice before using the system.

4.2.4.3 Joint Operation Planning System (JOPS) III User (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the JOPS III User course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.4.4 Prerequisites. In terms of training and experience, the prerequisites for this course are completion of lesson TUC011 or TUC012 in the Terminal User course, superficial knowledge of the JOPS Reporting System (JOPSREP), and basic experience in operation planning.

4.2.4.5 Use of Exhibits. This course cannot be successfully completed without the exhibits found in Attachment 16 of this manual. The computer will direct the trainee to the appropriate exhibits for each lesson.

4.2.4.6 Pattern of the Course. The Joint Operation Planning System (JOPS) III User course is currently under major revision. When completed, the course will consist of 33 lessons which should be executed consecutively:

(1) JOP005--The Planning Process - Introduction
(2) JOP010--The Planning Process - Planning Systems
(3) JOP015--The Planning Process - Joint Operation Planning System (JOPS)
(4) JOP020--The Planning Process - JOP III Automatic Data Processing Support
(5) JOP100--Force Requirement Generator (FRG) - Introduction
(6) JOP105--FRG Subsystem - Force Structuring Techniques
(7) JOP110--FRG Subsystem - Standard Force Considerations
(8) JOP115--FRG Subsystem - Nonstandard and Parent Force Entries
(9) JOP120--FRG Subsystem - Force Tailoring - Individual Force Update and Fragmentation
(10) JOP125--FRG Subsystem - - Force Tailoring - Split Shipments, FRN Assignment, and Force List Resequence
(11) JOP130--FRG Subsystem - Force Tailoring - Modification by Data Column Reference Codes and Force Deletion
(12) JOP135--FRG Subsystem - - Force Analysis
(13) JOP140--FRG Subsystem - Force Publications - TPFDD Audit
(14) JOP145--FRG Subsystem - Force Publications - TPFDL Generation
(15) JOP150--FRG Subsystem - The Force Package Option
(16) JOP155--FRG Subsystem - Nonunit-Related Cargo and Personnel
(17) JOP160--FRG Subsystem - Special FRG File Handling Techniques
(18) JOP200--Movement Requirements Generator (MRG) Subsystem - Introduction
(19) JOP205--MRG Subsystem - MRG Data Base Structuring
(20) JOP210--MRG Subsystem - MRG Data Base Update

(21) JOP215--MRG Subsystem - Resupply/Replacement Record Generation and Aggregation

(22) JOP300--Transportation Feasibility Estimate (TFE) Subsystem - Introduction

(23) JOP305--TFE Subsystem - TPFDD Evaluation - Mobility Facility Workloads

(24) JOP310--TFE Subsystem - TPFDD Evaluation - Lift Requirements

(25) JOP315--TFE Subsystem - Simulation Preparation - Construct/Update the TFE Control File

(26) JOP320--TFE Subsystem - Simulation Preparation - Channel Distance Generation

(27) JOP325--TFE Subsystem - Simulation Preparation - Vessel Selection and Port Constraints

(28) JOP330--TFE Subsystem - Simulation Preparation - Aircraft/Ship Analysis

(29) JOP335--TFE Subsystem - Air/Sea Simulations

(30) JOP340--TFE Subsystem - Post Simulation - Closure Match

(31) JOP345--TFE Subsystem - Post TFE Transportation Reports

(32) JOP350--TPFDD Refinement

(33) JOP400--Crisis Action System

The first four lessons of the new course are currently available. The old JOPS course has been provided for the edification of those students desiring to take it. The material is considered obsolete, and is of questionable value. The old JOPS course consists of nine lessons which should be executed consecutively:

(1) JOP510--The Basis of an Operation Plan

(2) JOP520--Plan Development - Converting an Existing Plan

(3) JOP530--Plan Development - Generating Forces

(4) JOP540--Plan Development - Tailoring Forces

(5) JOP550--Plan Development - Generating Nonunit Requirements

(6) JOP555--Plan Development - Transportation Evaluation

(7) JOP560--Plan Development - Entering Module Parameters, Part 1

(8) JOP561--Plan Development - Entering Module Parameters, Part 2

(9) JOP600--Crisis Action System (CAS) Overview

An additional lesson, JOP501, duplicates the material contained in Attachment 16 (exhibits for JOP510 through JOP600) of this manual. Accordingly, JOP501 should be executed only on a remote teletype terminal. The instructional lessons of the course have been especially designed for execution on the UIP and should not be attempted on the teletype terminal.

<u>4.2.4.7 Last Frame Labels.</u> The last frame in each lesson is labeled END.

<u>4.2.4.8 Lessons Available.</u> Following is a list of the available lessons, together with a synopsis of their content and estimates of completion time.

a. JOP005. This is the first lesson in the JOPS III User course. It sets the scene for deliberate planning and the rationalization for joint planning with ADP support, the scope of this Computer Directed Training (CDT) course, and the expected competence of the planner/user after completion.

TIME: Approximately 1.0 hour of on-line time is required to complete JOP005.

b. JOP010. This lesson provides a fundamental understanding of the joint planning process describing selected guidance, strategic, and technical documents. The CONPLAN, OPLAN, and OPORD are introduced.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP010.

c. JOP015. This lesson will familiarize the planner/user with the Joint Operation Planning System (JOPS). It outlines the five phases of the planning cycle with particular emphasis on those using Automatic Data Processing (ADP) support.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP015.

d. JOP020. The ADP support for operational planning is emphasized through an examination of the functional categories of the JOPS III subsystems. Basic computer languages, storage devices, and file structures are considered. The student learns about reference and OPLAN unique file content/structure.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP020.

Lessons JOP005 through JOP020 comprise the first module of the new JOPS CDTS course.

e. JOP501. In terms of training, JOP501 is not truly a lesson, because it contains no instruction. It merely enables the trainee who does not have a copy of attachment 16 to AFM 50-752 to obtain a teletype printout of the exhibits for lessons JOP510 through JOP600. (This is the only lesson in the course that should be executed on a teletype terminal.)

TIME: Approximately 3.0 hours are needed to copy the exhibits for all lessons.

f. JOP510. This is the first of the old JOPS III User Course. In this introductory lesson, the trainee is given an overview of the Joint Operational Planning system (JOPS). The lesscn begins with an introduction to the Joint Operation Planning system planning cycle, the Joint Organization for National Defense, and the Joint Strategic Planning System. The trainee is introduced to the six phases of the planning cycle (initiation, concept development, plan development, plan review, supporting plans, and execution planning). Before conducting the lesson, the trainee will be exposed to a simulation of JOPS III software, and will make the necessary responses to page a sample TDFDD file. The Deployment Reporting System (DEPREP) is also reviewed in this lesson.

TIME: Approximately 5.0 hours of on-line time are required to complete JOP510.

g. JOP520. The trainee will learn the preliminary steps in the conversion of an existing OPLAN to a new situation. Practical considerations are given to the process of selecting the best suited OPLAN for modification. The TPFDD

Interface Modules (D01A, D01B, and D03) are introduced. Given the TPFDD Interface Users Manual and a simulated situation that requires adapting a previously developed OPLAN, the trainee is required to select the modules to be executed and then provide the basic inputs for their successful operation.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP520.

h. JOP530. Instruction in this lesson focuses on the "generation of forces" during plan development. The lesson is designed to provide the trainee with the information necessary to effectively use the Force Requirements Generator (FRG) software system in the generation of identification data and force requirements for an OPLAN. Four subject areas are explored in detail--that is, (1) retrieval of the TPFDD file from magnetic tape (Module F69C), (2) entering OPLAN identification data (Module F10), (3) the selection/addition of Standard Force Requirements (Module F11), and (4) the selection/addition of Nonstandard Forces (Module F14). In a simulation of JOPS, the trainee is required to input the responses necessary to transfer a TPFDD file from tape to disk, change OPLAN identification data, select both standard and nonstandard forces, and produce the applicable reports.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP530.

i. JOP540. Like the previous lessons, JOP540--Plans Development - Tailoring Forces--relies heavily on simulation to teach the use of selected modules in OPLAN development. Two modules are simulated: (1) Module F20, Force Tailoring by Deletion, and (2) Module F21, Force Tailoring by Modification. A third Module, F22, Force Tailoring by Splitting Existing Units, is discussed in detail.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP540.

j. JOP550. The Movement Requirements Generator (MRG) is introduced as a tool to calculate the movement requirements that are nonunit-related. The trainee will practice the determination of nonunit-related cargo requirements and nonunit-related personnel requirements. This lesson also relies heavily on simulation to develop the desired skills and knowledges pertaining to use of the MRG. During this simulation, the trainee will use the MRG Users Manual and a Force List to provide the primary inputs to the MRG Modules that are needed to create a Planning Factors File (PFF), develop movement requirements for nonunit-related cargo and personnel, and update the TPFDD File with those movement requirements.

TIME: Approximately 3.0 hours of on-line time are required to complete JOP550.

k. JOP555. This lesson explores the three TPFDD/Interface (T/I) modules that assist transportation planners in making quick appraisals of workload at and through POEs and PODs. The trainee learns how Module D05 is used to generate and update facility utilization data; how Module D06 is used to produce reports applicable to POE/POD workload capabilities; and how D07 is used to produce reports applicable to specific time periods. The trainee will interact with a simulation of all three modules to produce the reports which will determine if POE and POD workloads have constraints that negate OPLAN feasibility.

TIME: Approximately 2.0 hours of on-line time are required to complete JOP555.

l. JOP560. The Transportation Feasibility Estimator (TFE) is introduced in this lesson. The trainee learns how to set up the Air and Sea Simulation Modules prior to their use in testing OPLAN feasibility. Topics discussed include constraints for the air and sea ports, routing channels, characteristics of ships and planes, and reduced capacity due to bomb damage and attrition. The following modules are discussed in detail: (1) Modules T10A and T10B, TPFDD/Scheduler Interface; (2) T11, Channel Summary; (3) T12, Route Planning; (4) T13, Port Retrieval; (5) T14, Retrieval; and (6) T15, Ship/Landing Craft Retrieval.

TIME: Approximately 4.0 hours of on-line time are required to complete JOP560.

m. JOP561. This is a continuation of lesson JOP560, which introduced the Transportation Feasibility Estimator (TFE) and simulated the entering of module parameters during OPLAN development. In this lesson, the trainee's ability to interact with the "sea" side of the TFE is tested. (LOP561 serves as the final project for lesson JOP560.)

TIME: Approximately 1.0 hour of on-line time is required to complete JOP561.

n. JOP600. This is the final lesson of the old JOPS III User Course. In this lesson, the trainee receives an in-depth overview of the Crisis Action System.

TIME: Approximately 1.0 hour of on-line time is required to complete JOP600.

4.2.5 Introduction to H6000 Series Computer Programming (H6000-CDT) Course.

4.2.5.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135B 001. It is titled "Introduction to H6000 Series Computer (H6000-CDT)," and it bears PDS Code T7K.

4.2.5.2 Purpose. This course is designed to provide trainees with an overview of hardware and software packages and Job Control Languages available on the H6000. It is intended for instructing persons who will be using one or more of the language processors: COBOL; FORTRAN; GMAP; JOVIAL; and SIMSCRIPT; Bulk Media Conversion (BMC) and Utility. The course is software-oriented, and is designed to give the student a basic understanding of the more important features and capabilities of H600 Series computers used at WWMCCS ADP sites.

4.2.5.3 Introduction to H6000 Series Computer Programming (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the Introduction to the H6000 course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS, Keesler AFB, MS 39534.

4.2.5.4 Prerequisites. Prior to starting this course, the trainee should satisfactorily complete one of the following three CDT lessons that teach how to operate a remote terminal and interact with CDTS: TUC010, Operation of the Remote Teletype; TUC011, Operation of the 786 VIP; or TUC012, Operation of the 7705 VIP. Persons who will be taking the course via teletype should take TUC010, persons who will be using VIP786 should take TUC011, and persons who will use VIP7705 should take TUC012.

4.2.5.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 17 of this manual. The computer will direct the trainee to the appropriate exhibits for each lesson.

4.2.5.6 Pattern of the Course. The Introduction to the H6000 Series Computer course consists of six lessons which should be executed consecutively:

(1) INT010--The Operating System
(2) INT020--Introduction to Batch Processing Software
(3) INT030--Job Control Language, Part 1
(4) INT040--Job Control Language, Part 2
(5) INT050--Bulk Media Conversion
(6) INT060--Utility

Each lesson interfaces with the next, so that the knowledges and skills acquired are cumulative. The first third of the course is information-oriented, while the two thirds is performance-oriented. Lesson difficulty increases as trainees proceed toward the final lesson. Although lessons can be taken independently, it is recommended that they be taken in numerical sequence.

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Thirty-four (34) hours are used for course length and diploma credit.

4.2.5.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.5.8 Lessons Available. Following is a list of the available lessons, together with an estimate of the time required for completion:

a. INT010. In this lesson, the trainee is introduced to the H6000 Operating System. Instruction focuses on the various phases that a computer program passes through (job flow), the composition of the General Comprehensive Operating Supervisor (GCOS), and methods of remote processing.

TIME: Approximately 2.5 hours of on-line time are required to complete INT010.

b. INT020. This lesson is designed to give the trainee an overview of the batch software available at a WWMCCS EDP site. It does not cover all of the available software, rather it concentrates on the software that would be available to a computer programmer using the central site computer. The lesson is divided into two sections. The first section deals with the language processors--COBOL, FORTRAN, GMAP, JOVIAL, and SIMSCRIPT. The second section deals with the system software--File System, Sort/Merge, Utility, Bulk Media Conversion, Library Editors, and data base processing.

TIME: Approximately 2.0 hours of on-line time are required to complete INT020.

c. INT030. This is the first part of a two-part lesson that pertains to Job Control Language (JCL). Instruction focuses on the purpose of JCL, job and activity definitions, and general format rules. The trainee will learn the basic principles of JCL, and (via simulation) build a job to assemble and execute a GMAP program. A basic job consisting of two activities--a COBOL activity and an execute activity--is explored. The functions and formats of the most commonly used JCL cards are discussed, and the trainee will practice coding these cards.

TIME: Approximately 3.5 hours of on-line time are required to complete INT030.

d. INT040. This is the second part of a two-part lesson on Job Control Language (JCL) for the H6000. The discussion of basic control cards is continued, and the trainee will be given an opportunity to analyze a job that includes four activities. The trainee will practice coding job control cards, and (via simulation) build a job that requires the use of several control cards. The final section of the lesson includes an analysis of the execution report. This lesson assigns INTRO PROJECT 1.

TIME: Approximately 4.0 hours of on-line and 4.5 hours off-line time are required to complete INT040.

e. INT050. After completing this lesson, the trainee will be able to code, run, and debug a Bulk Media Conversion (BMC) program. A basic BMC activity is explained, together with the necessary control cards. Typical BMC jobs are explained, including tape to printer, card to printer, card to tape,

tape to punch, and tape to card. An optional section of the lesson deals with the use of BMC to process nonstandard files to produce files in system standard format. This lesson assigns INTRO PROJECT 2.

TIME: Approximately 3.0 hours of on-line and 6.0 hours off-line time are required to complete INT050.

f. INT060. After completing this lesson, the trainee should be able to code, run, and debug a Utility program. Instruction includes an analysis of the required control cards and the optional control cards. The basic utility functions (COPY, MCOPY, COMP, SKIP, DUMP, DDUMP, ADUMP, AADUMP, and RWD/REW) are discussed. The trainee will practice coding a basic utility program. An optional section of the lesson deals with the HOLD and RPT functions, physical record processing, and numerous utility options. This lesson assigns INTRO PROJECT 3.

TIME: Approximately 3.0 hours of on-line and 5.5 hours off-line time are required to complete INT060.

4.2.6 WWMCCS System Software Analyst (H6000-CDT) Course. This course is currently under major revision. Information about course content will be forthcoming as new lessons become available.

4.2.7 Integrated Data Store (IDS) (H6000-CDT) Course.

4.2.7.1 Course Number, Title, and PDS Code. This course is number ECDTS5135B 000; it is titled "Integrated Data Store (IDS) (H6000-CDT)"; and it bears PDS Code T7J.

4.2.7.2 Purpose. This course is designed to provide experienced WWMCCS programmers and analysts with a working knowledge of the Integrated Data Store (IDS) system. Basically, the course teaches the IDS programming language; however, it combines instruction on the language with the "why's" and "how's" of the IDS system. After completing this course, the trainee should be able to code, prepare job control cards, compile, and execute an IDS program.

4.2.7.3 Integrated Data Store (IDS) (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the Integrated Data Store (IDS) course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.7.4 Prerequisites. Trainees entering this course must have a thorough working knowledge of COBOL as implemented on the H6000 Series computers. Those who enter this course without experience in the actual writing and running of COBOL programs will find that their background is probably inadequate. Prior to starting the course, the trainee should also complete one of the lessons that teach how to operate the remote terminal and interact with CDTS. Persons who will be taking the course via teletype should first complete lesson TUC010, those who will be using VIP786 should take TUC011, and those who will be using a series 7700 VIP should take TUC012.

4.2.7.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits. These exhibits are found in Attachment 19 of this manual. The computer will direct the trainee to the appropriate exhibits for each lesson.

4.2.7.6 Pattern of the Course. The Integrated Data Store (IDS) course consists of 20 lessons which should be executed consecutively:

(1) IDS010--Introduction to Integrated Data Store
(2) IDS011--Basic IDS Terms
(3) IDS012--General IDS File Structures
(4) IDS013--IDS Record Relationships
(5) IDS020--IDS Chains
(6) IDS021--IDS Physical Structure
(7) IDS030--COBOL IDS Program Differences
(8) IDS031--Environment and Data Division Changes
(9) IDS032--IDS Data Division Changes
(10) IDS040--IDS Parts of a COBOL Procedure Division
(11) IDS041--IDS Procedure Division (Part I)
(12) IDS042--IDS Procedure Division (Part II)
(13) IDS043--IDS Mid-course Project
(14) IDS050--Data Base Creation
(15) IDS060--IDS Data Base Management
(16) IDS070--IDS Definition Structure
(17) IDS080--Advanced IDS Programming Capabilities
(18) IDS090--IDS Data Base Management Software Routines
(19) IDS100--IDS Data Base Management Software Routines (Cont)
(20) IDS110--IDS Journalization

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Thirty (30) hours are used for course length and diploma credit.

4.2.7.7 Last Frame Labels. The last frame of each lesson is labeled END.

4.2.7.8 Lessons Available. Following is a list of the available lessons, together with an estimate of the time required to complete each one:

a. IDS0[illegible]. This lesson presents the basic concepts of IDS. Topics include definition of IDS, capabilities of IDS language, IDS Deck set-up, and compilation process.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS010.

b. IDS011. This lesson presents the basic terms of IDS. Topics include IDS file, page, types of pointers, chaining characteristics and advantages.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS011.

c. IDS012. This lesson presents general IDS file structure. Topics include structure diagrams (shorthand and longhand) and record classes.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS012.

d. IDS013. This lesson presents general IDS record relationships. Topics include hierarchy, network, details of more than one chain, master of more than one chain, multiple detail record types per chain, master/detail between multiple details, connector records, and parallel network. The trainee will interpret structure diagrams and distinguish between legal and illegal structures.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS013.

e. IDS020. This lesson, entitled "IDS Chains", discusses the interrelationships of IDS curreuts, chain-ordering, and chain tables. Instruction includes how detail records are ordered in a particular chain. The lesson also briefly covers how IDS keeps track of records for the IDS programmer.

TIME: Approximately 1.0 hour of on-line time is required to complete 'DS020.

f. IDS021. This lesson, entitled "IDS Physical Structure", focuses on the physical structures of an IDS file. It basically presents the five different aspects of IDS physical structure and how a Data Base is broken into its various parts.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS021.

g. IDS030. This lesscn, entitled "COBOL/IDS Program Differences", deals with some of the differences in a COBOL/IDS program. The differences are covered in generalized terms. Instruction includes changes in the Identification, Environment, and Data Divisions, with more emphasis applied in a later lesson. Finally, it includes the types of verbs that are used in the Procedure Division.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS030.

h. IDS031. This lesson focuses on the necessary changes in the Environment Division, and partially covers the Data Divisior. Instruction covers the "IDS-SPECIAL-NAMES" paragraph, the "SELECT IDS" statement, and IDS language in the Environment Division. The discussion of the Data Division deals with the Mnemonic (MD) Level INDICATOR entires and the 01 Record Description Entries.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS031.

i. IDS032. This lesson, entitled "IDS Data Division Changes", continues from IDS031 with more Data Division changes. Instruction includes the 02 and 98 level entries for the record descriptions in the Data Division. The lesson also covers the required and optional statements available. After completion of this lesson, the trainee should be able to code the IDS Environment and Data Divisions to conform to the specifications of IDS structure diagrams and the attendant narrative.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS032.

j. IDS040. The lesson entitled "IDS Parts of a COBOL Procedure Division", is the first of a three-lesson series that teaches the verbs of the IDS Procedure Division. Instruction includes a discussion of how to manipulate an IDS Data Base. Also included in the discussion is the "ENTER IDS" statement and the Data Base Maintenance and Recovery verbs.

TIME: Approximately 1.0 hour of on-line time and 0.5 hour of off-line time are required to complete IDS040.

k. IDS041. This lesson continues the discussion on the IDS Procedure Division and how it is used. Instruction includes the Data Base Update verbs, the Data Base Retrieval verbs, and the miscellaneous verbs.

TIME: Approximately 1 hour of on-line time is required to complete IDS041.

l. IDS042. This lesson is the last of the series that teaches the verbs of the IDS Procedure Division. Instruction focuses on how these verbs apply in a situation to process against or modify a Data Base after it has been loaded. It includes the types of transactions and how their use affects the IDS Data Base.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS042.

m. IDS043. This lesson, entitled "IDS Mid-course Project", focuses on coding the Procedure Division for the problem that is specified in the exhibits. Instruction includes a question/answer coding session, with a remedial review of areas as needed.

TIME: Approximately 1.0 hour of on-line and 3.0 hours off-line time are required to complete IDS043.

n. IDS050. This lesson focuses on the creation of an IDS Data Base. Instruction includes debugging aids, subroutines and subroutine statistics.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS050.

o. IDS060. This lesson concerns IDS Data Base Management. Instruction includes effective techniques to increase the efficiency of IDS programs, memory management, page buffers, and work areas.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS060.

p. IDS070. This lesson deals with the IDS definition structure. Instruction includes focus on the definitions and breakdown of each record type.

TIME: Approximately 1.0 hour of on-line time is required to complete IDS070.

q. IDS080. This lesson presents IDS advanced programming capabilities. Instruction includes multi-area considerations, dynamic area control, fixed area control and other options that are available.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS080.

r. IDS090. This lesson presents IDS Data Base Management software routines. Instruction includes utility programs that help a Data Base manager to monitor and keep control of his Data Base.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS090.

s. IDS100. This lesson concerns more IDS Data Base management software routines. Instruction includes more of the utility programs that allow a Data Base manager to monitor and to keep track of his Data Base.

TIME: Approximately 1.5 hours of on-line time are required to complete IDS100.

t. IDS110. This lesson deals with IDS journalization. Instruction includes basic IDS journalization/recovery capabilities, the abbreviated option of journalization and the IDS Data Base recovery system.

TIME: Approximately 1.0 hour of on-line and 3.5 hours off-line time are required to complete IDS110.

4.2.8 H6000 Computer Operator (H6000-CDT) Course.

4.2.8.1 Course Number, Title, and PDS Code. This course is numbered ECDTS51130 003; it is titled "H6000 Computer Operator (H6000-CDT)"; and it bears PDS Code S3C.

4.2.8.2 Purpose. This course is designed to train qualified computer operators from the joint military services in the skills and knowledges necessary to operate and manage the Honeywell 6000 Series Computer System. Instruction focuses on system configuration, system initialization, and system operation.

4.2.8.3 H6000 Computer Operator (H6000-CDT) Course Material OPR. Recommendations for improving the instruction materials in the H6000 Computer Operator course should be sent to 3300 TCHTW/TTGSZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.8.4 Prerequisites. Trainees who enter this course should have a minimum of 6 months' experience as a computer operator. Prior to starting the course, the trainee should also complete one of the lessons that teaches how to operate the remote terminal and interact with CDTS. Persons who will be taking the course via teletype should first complete lesson TUC010, those who will be using VIP786 should take TUC011, and those who will be using VIP7705 should take TUC012.

4.2.8.5 Usé of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 20 to this manual. The computer will direct the trainee to the appropriate exhibits for each lesson.

4.2.8.6 Pattern of the Course. The H6000 Computer Operator course is currently under major revision. The existing course consists of 13 on-line lessons (OPRXXX) and 5 Task Performance Training (TPTXXX) projects. The first three lessons of the new course are concurrently available with the old course, starting with lesson OPR510. The new course, when completed, will consist of four modules, which are comprised of 19 lessons and 5 off-line projects. Currently, certificates of completion will be issued for completion of lessons OPR010 thru OPR120. The lessons should be executed consecutively. The complete sequence of lessons and projects follows.

SECTION I--System Configuration

(1) OPR010--H6000 Computer System Hardware
(2) OPR020--SCU Configuration Panel Switches
(3) OPR030--IOM Configuration Panel Switches, Part 1
(4) OPR031--IOM Configuration Panel Switches, Part 2
(5) OPR040--DATANET 355 DIA Configuration Panel Switches
(6) OPR050--Processor Configuration Panel Switches
(7) TPT010--Main Frame Panel Switches

SECTION II--System Initialization

(8) OPR060
(9) TPT020--Hardware Configuration ($CONFIG) Section
(10) OPR070--Startup Program (Initialize, Edit, Files, Patch, and Load Sections)

SECTION III--System Operation

(11) OPR080--Peripheral Subsystems and Communications
(12) TPT030--Peripheral Operations
(13) *OPR090--Console Messages and Verbs*
(14) TPT040--System Operation - Console Messages and Verbs
(15) OPR100--Simulated System Operation TPT050--Cold Boot, Warm Boot, and Master Mode Dump
(16) OPR110--System Console Messages
(17) OPR120--Boot Simulator

Instruction in this version of the course consists of 37.0 hours of on-line training (lessons) via the Computer Directed Training System and 26.0 hours of "hands-on" Task Performance training in the computer room, plus 15.0 hours of off-line projects and reading assignments.

REVISED COURSE

MODULE I--Basic H6000 Operations

(1) OPR510--H6000 Mainframes
(2) OPR520--H6000 Software and Job Flow
(3) OPR530--H6000 Peripherals
(4) OPR540--Console Operations: Input Messages
(5) OPR550--Console Operations: Output Messages
(6) OPR560--H6000 Remote Processing Software
(7) OPR570--H6000 Remote Processing Hardware
(8) OPR580--Master Mode Dumps

MODULE II--H6000 Boots and Related Activities

(9) OPR590--H6000 Booting
(10) OPR600--H6000 Cold Booting
(11) OPR610--Cold Boot Simulation

MODULE III--System Configuration Techniques

(12) OPR620--Mainframe Hardcore Monitors
(13) OPR630--Mainframe Configuration Switches
(14) OPR640--H6000 Start-Up Deck: Configuration
(15) OPR650--H6000 Start-Up Deck: Edit-Load
(16) OPR660--Reconfiguration Considerations

MODULE IV--Total On-line Test System (TOLTS)

(17) OPR670--Introduction to the Total On-line Test System (TOLTS)
(18) OPR680--Peripheral On-line Test System (POLTS)
(19) OPR690--Main-frame On-line Test System (MOLTS)

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Sixty-two and a half (62.5) hours are used for course length and diploma credit.

4.2.8.7 Last Frame Labels. The last frame of each lesson is labeled END.

4.2.8.8 Lessons Available. Following is a complete list of available lessons, together with an estimate of the time required to finish each one:

a. OPR010. This lesson presents an overview of H6000 computer system hardware. Instruction includes an abbreviated analysis of the features, functions, and interrelationships of main frame components, including the memory module, the processor module, the input-output multiplexer, and the front-end network processor. Instruction also includes a discussion of various switches, pushbuttons, and indicators on the Operator's Control Panel.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR010.

b. OPR020. This lesson, titled System Controller Unit (SCU) Configuration Panel Switches, is the first of a series of lessons pertaining to the setting of switches. The purpose of lesson OPR020 is to develop knowledge that will enable the trainee to correctly set each switch of the SCU configuration panel for any specific system hardware or software configuration. The trainee will be familiarized with the physical location of each SCU configuration panel switch and group of switches and then use an exhibit to practice making switch settings for various configurations.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR020.

c. OPR030. This lesson focuses on the setting of Input-Output Multiplexer (IOM) Configuration Panel Switches. The trainee will first learn the physical location of each switch, then use an exhibit to practice setting all necessary switches for a stated H6000 configuration. After properly setting the IOM panel switches, the trainee will practice resetting the affected switches in response to changes in the system.

TIME: Approximately 6.5 hours of on-line time are required to complete OPR030.

d. OPR031. This lesson, a continuation of OPR030, tests the trainee's ability to set switches on the Input-Output Multiplexer (IOM) configuration panel.

TIME: Approximately 2.0 hours of on-line time are required to complete OPR031.

e. OPR040. This lesson, like the previous lessons, is geared toward developing the trainee's ability to correctly set switches. Lesson OPR040 teaches settings for switches on the DATANET 355 DIA configuration panel. Once again, the trainee will learn the physical location of the various switches; then use an exhibit to practice making correct switch settings for each DATANET 355, based on the system hardware/software configuration. The trainee will then be required to properly set the switches to accommodate changes in the configuration.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR040.

f. OPR050. This lesson, a continuation of the on-line instruction on switch settings, is titled Processor Configuration Panel Switches. The lesson theme is identical to that used in the previous lessons--that is, the trainee first learns the physical location of the switches, then practices making switch settings for specific software and hardware configurations. Finally, the trainee will practice resetting the affected switches in response to changes in the system configuration.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR050.

g. TPT010. This is the first Task Performance Training (TPT). This is not an on-line CDT lesson; instead, it is "hands-on" training time (on-the-job training) that requires the student to apply the knowledges gained in lessons OPR010 through OPR050. The student will learn how to apply power to hardware modules, and practice the applicable system configuration procedures. To satisfactorily complete this TPT, the student must perform the tasks listed in exhibit TPT010-1 in accordance with local standards and procedures.

TIME: TPT010 requires approximately 3 hours; however, exact time will be determined/specified by the local Training Monitor (TM).

h. OPR060. This is the first lesson in Section II, System Initialization. In this lesson, the trainee will become familiar with the System Startup Manual and gain a limited working knowledge of those portions that are frequently used by the computer operator. A part of the lesson is devoted to card formats, providing the trainee with the knowledge required to construct the hardware configuration section of the startup deck.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR060.

i. TPT020. This is the second Task Performance Training (TPT) project. In this "hands-on" lesson, the student (with assistance from the Training Monitor) will use a keypunch machine and applicable manuals to create a complete hardware configuration ($CONFIG) section of the startup program card deck for the local computer system. The student must satisfactorily perform the tasks listed in exhibit TPT020-1 in accordance with local standards and procedures.

TIME: TPT020 takes approximately 5 hours; however, exact time will be determined/specified by the local Training Monitor (TM).

j. OPR070. The purpose of this lesson is to instruct the trainee in the proper use and function of the remaining sections of the startup deck. Emphasis is placed on student familiarity with the System Startup Manual. The instruction is tailored toward developing those skills and knowledges that the trainee needs to construct the remaining system definition sections (initialize, edit, files, patch, and load) of the startup deck.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR070.

k. OPR080. This is the first lesson of the third and final section of the course (Section III, System Operation). This lesson presents an overview of H6000 Series peripheral subsystems and their interrelationships to the mainframe modules. After completing this lesson, the trainee should have a basic understanding of peripheral operations, communications, and IOM channel connections to specific peripheral devices.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR080.

1. TPT030. This is another Task Performance Training project. Again, this is not an on-line CDT lesson, rather it is "hands-on" training that is closely supervised by the Training Monitor (TM). In this TPT, the trainee will perform the various tasks listed in exhibit TPT030-1 in accordance with local standards and procedures. These tasks require the trainee to satisfactorily operate each peripheral subsystem, identify functions of the Operator's Control Panel, identify and correct peripheral equipment stoppages, perform operator maintenance, observe safety practices, power up/down each subsystem, and Bootload the DATANET 355.

TIME: TPT030 requires approximately 5 hours; however, exact time will be determined/specified by the local Training Monitor (TM).

m. OPR090. The primary purpose of this lesson is to develop the trainee's ability to use the Console Message Manual. The lesson, titled Console Messages and Verbs, requires the trainee to participate in simulated operational situations. The trainee will respond to console messages generated by the various situations. The lesson will teach the proper use of ten console verbs, and the trainee will become acquainted with the system replies that may be generated when each verb is used.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR090.

n. TPT040. This Task Performance Training (TPT) requires the student to operate the System Control Console a minimum of 4 hours. Under the guidance of the Training Monitor (TM), the trainee will: (1) correctly identify and respond to applicable system output messages, (2) identify system output messages of record, and (3) correctly input operator initiated console verbs.

TIME: TPT040 requires approximately 4 hours; however, exact time will be determined/specified by the local Training Monitor (TM).

o. OPR100. This lesson features a simulation of H6000 system operation. The trainee will use the startup deck and system description cards that were constructed in lesson OPR060 and lesson OPR070. Using the applicable manuals and a CDT simulation of an operational environment, the trainee will be required to initialize the system, operate the system, and provide responses and actions to various output messages. The purpose of this lesson is to provide the student with the skills and knowledges necessary to accomplish advanced H6000 computer operations. It will acquaint the trainee with some of the problems and conditions that will be encountered in a "live" operational environment.

TIME: Approximately 3.0 hours of on-line time are required to complete OPR100.

p. TPT050. This Task Performance Training (TPT) is titled "System Operation." In this project the trainee will call upon all of the knowledges and skills that have been accumulated throughout the H6000 Computer Operator course. Under the close supervision of the Training Monitor (TM), the trainee will be required to accomplish the various tasks listed in exhibit TPT050-1. These tasks require the trainee to: (1) cold boot the system, (2) warm boot the system, (3) execute a master mode dump, and (4) operate the system under "live" operational conditions.

TIME: TPT050 requires approximately 8 hours; however, the exact time will be determined/specified by the local Training Monitor (TM).

q. OPR110. This lesson is a "final" knowledge test on system operation. It consists of 50 questions that require the trainee to use the Honeywell System Console Messages reference manual.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR110.

r. OPR120. This lesson will simulate the Boot Procedure for the H6000.

TIME: Approximately 0.5 hour of on-line time is required to complete OPR120.

s. OPR510. This lesson, entitled "H6000 Mainframes", is the first lesson of the revised Operator course. The lesson covers the basic relationship of the four mainframe modules and how the overall system functions.

TIME: Approximately 1.0 hour of on-line time is required to complete OPR510.

t. OPR520. This lesson presents a general introduction of H6000 software concepts and the six phases of software job flow in the H6000 computer. Monitoring the video is also discussed.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR520.

u. OPR530. This lesson, entitled "H6000 Peripherals," acquaints the student with a cross section of peripherals used at standard WWMCCS sites.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR530.

v. OPR540. This lesson, entitled "Console Operations: Input Messages", covers the input messages used in console operations. The student will learn the various categories of input message (verbs), with concentration on a few of the most used verbs from each section.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR540.

w. OPR550. This lesson, entitled "Console Operation: Output Messages", teaches console output messages and interpretation of their meaning. The physical operation of the system console is also discussed.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR550.

x. OPR670. This lesson, entitled "Introduction to the Total On-Line Test System (TOLTS)", was developed to introduce the computer operator to the capabilities and advantages of this test and diagnostic package. The TOLTS subsystem and necessary processing command formats are identified.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR670.

y. OPR680. This lesson, entitled "Peripheral On-Line Test System (POLTS)", is devoted to the POLTS commands. The options of these commands are covered in depth and reinforced by on-line simulations.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR680.

z. OPR690. This lesson, entitled "Mainframe On-Line Test System (MOLTS)", focuses on the MOLTS commands. The options are covered in depth and reinforced by on-line simulation.

TIME: Approximately 1.5 hours of on-line time are required to complete OPR690.

NOTE: Information on additional lessons will be provided when they become available.

4.2.9 Computer Directed Training System (CDTS) Coursewriter (H6000-CDT) Course.

4.2.9.1 Course Number, Title, and PDS Code. A course number of ECDTS5135B 003 and a PDS Code of VZJ have been assigned to this course titled "Computer Directed Training System (CDTS) Coursewriter (H6000-CDT)."

4.2.9.2 Purpose. This course is designed to provide WWMCCS users with the ability to use the general purpose Computer Directed Training System for Site and Command unique training requirements. This course teaches the CDT programming language and the use of the compiler and loader for building CDT material. Upon the completion of this course, the trainee will be proficient as a CDT coursewriter.

4.2.9.3 Computer Directed Training System (CDTS) Coursewriter (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials in the CDTS Coursewriter course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.9.4 Prerequisites. The trainee should have a thorough working knowledge of terminal operations and the CARDIN and Text Editor time-sharing subsystems. This requirement may be satisfied by the successful completion of the Terminal User Course.

4.2.9.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 10 of this manual. The computer will direct the trainee to the appropriate exhibits for each lesson.

4.2.9.6 Pattern of the Course. The CDTS Coursewriter course consists of six on-line lessons which should be executed consecutively:

(1) CDT010--Frame Formats, Terminology, and Standards
(2) CDT020--Multiple-Choice (M) Frame Building
(3) CDT030--Question (Q) Frame Building
(4) CDT040--Decision (D) Frame Building
(5) CDT050--CARDIN Simulation
(6) CDT060--Compiling and Loading

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Thirteen (13) hours are used for course length and diploma credit.

4.2.9.7 Last Frame Labels. The last frame of each lesson is labeled END.

4.2.9.8 Lessons Available. Following is a complete list of available lessons, together with an estimate of the time required to finish each one:

a. CDC010. This lesson is designed to introduce you to CDTS coursewriting. You will be taught to identify CDT symbols, feedback messages, parameters, terms, and the standards used in CDT coursewriting.

TIME: Approximately 1.0 hour of on-line time is required to complete CDT010.

b. CDT020. This lesson is designed to give you the information that will enable you to code multiple-choice (M) frames.

TIME: Approximately 1.5 hours of on-line time are required to complete CDT020.

c. CDT030. This lesson is designed to give you the information necessary to code question (Q) frames.

TIME: Approximately 1.5 hours of on-line time are required to complete CDT030.

d. CDT040. This lesson instructs you on how to write decision (D) frames. It also explains how to identify connectives and use relational operators.

TIME: Approximately 1.0 hour of on-line time is required to complete CDT040.

e. CDT050. This lesson reviews the CARDIN subsystem and explains the BPRINT subsystem. This lesson assigns CDT PROJECT 1.

TIME: Approximately 1.0 hour of on-line and 3.0 hours off-line time is required to complete CDT050.

f. CDT060. This lesson instructs you in the use of the CDT Compiler and the Utility Load program. This lesson assigns CDT PROJECT 2.

TIME: Approximately 1.0 hour of on-line and 3.0 hours off-line time are required to complete CDT060.

4.2.10 PDP-11 Computer Operator (H6000-CDT) Course.

4.2.10.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135B 002; it is titled "PDP-11 Computer Operator (H6000-CDT) Course"; and it bears PDS Code VZK.

4.2.10.2 Purpose. This course familiarizes the student with minicomputer concepts and conventions used in the RSX-11D Operating System (Version 6). The trainee is taught how to bootstrap the system, use the MCR commands to control the system, assemble/compile programs, task build tasks, execute and abort tasks, execute the utility software packages, and perform a system generation.

4.2.10.3 PDP-11 Computer Operator (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the PDP-11 Computer Operator course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.10.4 Prerequisites. The trainee is expected to know basic computer operator techniques and be familiar with general data processing terms. Prior to starting the course, the trainee should also complete either lesson TUC010, TUC011, or TUC012, which describe how to interact with the CDT system.

4.2.10.5 Use of Exhibits. No exhibits are required; however, the PDP-11 (RSX-11D) technical manuals are referenced throughout the course to provide supplemental and optional material. Learning will be enhanced by using the manuals. See the description of each lesson in subsequent paragraphs regarding the specific manuals to be used in each lesson.

4.2.10.6 Pattern of the Course. The PDP-11 Computer Operator course consists of ten lessons which should be executed consecutively.

(1) PDP010--Introduction to Minicomputers and RSX-11D Terminology
(2) PDP020--PDP-11 Conventions and System Initialization
(3) PDP030--Monitor Console Routine (MCR) Commands
(4) PDP040--Assembling/Compiling Tasks
(5) PDP041--Task Building and Executing Tasks
(6) PDP050--The Peripheral Interchange Program (PIP) Utility
(7) PDP060--File Transfer, Dump, Librarian, Verify, and Zap Utilities
(8) PDP070--Line Text Editor (EDI) Utility
(9) PDP080--The Preserve Utility
(10) PDP090--System Generation (SYSGEN)

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Twenty-four (24) hours are used for course length and diploma credit.

4.2.10.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.10.8 Lessons Available. The following is a list of the available lessons, together with an estimate of the time required for completion and necessary manuals.

a. PDP010. Includes an Introduction to minicomputer concepts and descriptions of the hardware and software associated with the RSX-11D Operating System. Fundamentals of RSX-11D, such as partitions, checkpointing, device independence, and spooling are also discussed.

TIME: Approximately 2.0 hours of on-line time are required to complete PDP010. Reference Manual: RSX-11D User's Guide (optional)

b. PDP020. This lesson develops the basic formats and conventions used with the RSX-11D Operating System. Such things as device, file, and terminal control conventions are taught as well as the file specifier and command line syntax. Procedures are given to bootstrap the system.

TIME: Approximately 2.5 hours of on-line time are required to complete PDP020. Reference Manual: RSX-11D User's Guide (required)

c. PDP030. This lesson tells how and why the Monitor Console Routine (MCR) Commands are used. Both general and privileged users are discussed; the role of each type user, what options are available to them, and their applicable commands. The primary commands to be discussed are HELLO, PASSWORD, BYE, SYSTEM Query, MOUNT/DISMOUNT, OPERATE/QUEUE, TIME, INSTALL, RUN, REMOVE, ABORT, and commands to initialize volumes and directories.

TIME: Approximately 3.0 hours of on-line time are required to complete PDP030. Reference Manual: RSX-11D User's Guide (required)

d. PDP040. This lesson shows how to initiate the MACRO-11 assembler and assemble a source program and how to initiate the FORTRAN compiler and compile a source program.

TIME: Approximately 2.0 hours of on-line time are required to complete PDP040. Reference Manual: RSX-11D User's Guide (optional)

e. PDP041. This lesson shows how to initiate Task Builder and how to build and execute tasks.

TIME: Approximately 2.5 hours of on-line time are required to complete PDP041.

Reference Manual: RSX11-D Task Builder Reference Manual (optional)

f. PDP050. This lesson shows how and when to use the peripheral Interchange Program (PIP). Because PIP is so frequently used and highly beneficial to the operator, its commands and capabilities are fully discussed. Included are commands to copy, append, merge, list, delete, purge, remove, spool, and protect files.

TIME: Approximately 2.5 hours of on-line time are required to complete PDP050.

Reference Manual: RSX-11D Utility Programs Procedures Manual (required)

g. PDP060. Several of the utilities are discussed in the lesson, including File Transfer (FLX), Dump (DMP), Librarian (LBR), Verify (VFY), and AAP. FLX, LBR, and ZAP are not discussed in depth, only how and when to use them. DMP is discussed in greater detail, to include the options available and practice entering DMP commands. VFY also is discussed in depth, including how to verify volumes and locate lost files.

TIME: Approximately 2.0 hours of on-line time are required to complete PDP060.

Reference Manual: RSX-11D Utility Programs Procedures Manual (required)

h. PDP070. The Line Text Editor (EDI) is discussed in detail to include how it is used, its purpose, the control modes, how to create files, how to enter data, ways of modifying data, and methods of terminating Editor.

TIME: Approximately 2.5 hours of on-line time are required to complete PDP070.

Reference Manual: RSX-11D Utility Programs Procedures Manual (optional)

i. PDP080. The on-line and the stand-alone versions of the Preserve Utility are studied. The purpose of Preserve, the command line syntax, how each of the versions is used, and their benefits are the primary topics. The three format types (image, Files-11, and logical tape) plus other options are also discussed.

TIME: Approximately 1.5 hours of on-line time are required to complete PDP080.

Reference Manual: RSX-11D System Generation Reference Manual (optional)

j. PDP090. The purpose of System Generation (SYSGEN) and steps to be taken in preparing for a SYSGEN are discussed. The trainee progresses through equipment specifics, required files and directories, the generation and build command files, and selecting a target device. A simulated SYSGEN is accomplished to include executing Phases I and II and saving the generated system image.

TIME: Approximately 1.0 hour of on-line time is required to complete PDP090.

Reference Manual: RSX-11D System Generation Reference Manual (required)

4.2.11 Selective Inquiry System (SIS) (H6000-CDT) Course.

4.2.11.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135C 003; it is titled "Selective Inquiry System (SIS) (H6000-CDT)"; and it bears PDS Code ZQQ.

4.2.11.2 Purpose. This course is designed for use by computer programmers, systems analysts, and functional analysts working in a Honeywell Series 6000 computer environment. The course can be used to provide a working knowledge of the Selective Inquiry System to personnel who are totally unfamiliar with the retrieval language or as a review for experienced personnel.

4.2.11.3 Selective Inquiry System (SIS) (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the SIS course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.11.4 Prerequisites. Prior to starting this course, the trainee should complete either lesson TUC010, TUC011, or TUC012.

4.2.11.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 21 of this manual. The trainee will be directed by the computer to appropriate exhibits for each lesson.

4.2.11.6 Pattern of the Course. The Selective Inquiry System course consists of seven lessons which should be executed consecutively:

(1) SIS010--Job Control Language Overview
(2) SIS020--File Structures
(3) SIS030--Phase I Directives, Part 1
(4) SIS040--Phase I Directives, Part 2
(5) SIS050--Phase II Directives and Phase III Directives, Part 1
(6) SIS060--Phase III Directives, Part 2
(7) SIS070--Time-sharing Retrieval Subsystem

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Nine (9) hours are used for course length and diploma credit.

4.2.11.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.11.8 Lessons Available. The following is a list of the available lessons, together with an estimate of the time required for completion.

a. SIS010. This lesson is titled "H6000 Job Contiol Language," and it provides a general overview of the JCL statements which are normally required to execute the Selective Inquiry System.

TIME: Approximately 1.0 hour of on-line time is required to complete SIS010.

b. SIS020. The lesson is titled "File Structures," and it provides a general overview of the types of files that may be used as input to SIS.

TIME: Approximately 0.5 hour of on-line time is required to complete SIS020.

c. SIS030. This lesson is titled "Phase I Directives, Part 1," and it begins the discussion of the control cards that are required or optional in the execution of Phase I of the SIS program.

TIME: Approximately 1.0 hour of on-line time is required to complete SIS030.

d. SIS040. This lesson is titled "Phase I Directives, Part 2," and it completes the discussion of the directives that are requi-ed or optional in the execution of Phase I of the SIS program. This lesson assigns SIS OPTIONAL PROJECT 1.

TIME: Approximately 1.0 hour of on-line time is required to complete SIS040.

e. SIS050. This lesson is titled "Phase II Directives and Phase III Directives, Part 1," and it covers all directives required or optional to execute Phase II of the SIS program and a portion of the directives required or optional for Phase III. This lesson assigns SIS OPTIONAL PROJECT 2.

TIME: Approximately 1.5 hours of on-line time are required to complete SIS050.

f. SIS060. This lesson is titled "Phase III Directives, Part 2," and it concludes the discussion of the directives required or optional to execute Phase III of the SIS program.

TIME: Approximately 1.0 hour of on-line time is required to complete SIS060.

g. SIS070. This lesson is titled "Time-sharing Retrieval Subsystem," and it provides an overview of how to use the time-sharing version of SIS.

TIME: Approximately 1.0 hour of on-line time is required to complete SIS070.

4.2.12 H6000 Total On-Line Test System (TOLTS) (H6000-CDT) Course.

4.2.12.1 Course Number, Title, and PDS Code. This course is numbered ECDTS51130 002; it is titled "H6000 Total On-Line Test System (TOLTS) (H6000-CDT) Course; and it bears PDS Code VZH.

4.2.12.2 Purpose. This course has two objectives. First, it is designed to familiarize both site managers and computer operators with the capabilities and advantages of using the TOLTS testing and diagnostic software to locate computer hardware malfunctions. (No diploma is offered for this portion of the course.) Second, the course provides detailed instruction to experienced computer operators on executing the Peripheral On-Line Test Subsystem (POLTS) and the Main Frame On-Line Test Subsystem (MOLTS) to check for computer hardware malfunctions.

4.2.12.3 H6000 Total On-Line Test System (TOLTS) (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in this course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.12.4 Prerequisites. Prior to starting the course, trainees must complete either lesson TUC010, TUC011, OR TUC012. Site managers who only desire an overview of the TOLTS software package may enter the first lesson without further prerequisites. Computer operators who are taking the course for credit should have 6 months operating experience on the H6000 computer.

4.2.12.5 Use of Manuals. A trainee cannot successfully complete this course without the necessary manuals. Two manuals are used extensively in lessons TOL020 AND TOL030. The required Honeywell manuals are Total On-Line Test System (TOLTS) Reference Manual and Total On-Line Test System (TOLTS) Test Pages.

4.2.12.6 Pattern of the Course. The H6000 Total On-Line Test System (TOLTS) course consists of three lessons which should be executed consecutively.

(1) TOL010--Introduction to Total On-Line Test System (TOLTS)
(2) TOL020--Peripheral On-Line Test Subsystem (POLTS)
(3) TOL030--Main Frame On-line Test Subsystem (MOLTS)

The instruction in this course is self-paced, actual completion time ay vary. However, all lessons and lesson projects not indicated as optional are required. Eight (8) hours are used for course length and diploma credit.

4.2.12.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.12.8 Lessons Available. Following is a list of the available lessons, together with an estimate of the time required for completion:

a. TOL010. This lesson is an introduction to the course on the Total On-Line Test System (TOLTS). It was developed to benefit both the experienced operator and the site manager as to the capabilities and advantages of this test and diagnostic package. Within the lesson, the characteristics of TOLTS are described. You will also identify the TOLTS subsystems, the necessary processing command formats, and the use of remote terminal capabilities. This lesson will provide background to the site manager who will not be require to take the following lessons and to the operators who will learn the specifics of the subsystems in the next lesson.

TIME: Approximately 1.0 hour of on-line time is required to complete TOL010.

b. TOL020. This lesson is devoted to the Peripheral On-Line Test Systcm (POLTS). Within the lesson you will learn the commands needed to operate this subsystem, the test commands, and the options that are available to the operator. By use of simulation, this lesson will require you to enter appropriate commands to run PCLTS tests on the following pieces of hardware: disk, tape drives, printer, punch, card reader, and the system console.

TIME: Approximately 2.5 hours of on-line time are required to complete TOL020.

c. TOL030. This lesson focuses on the Main Frame On-Line Test System (MOLTS). This lesson will introduce you to the capabilities of the MOLTS subsystem and the procedures needed to run such tests. The command format, along with special options that can be used, will be discussed. Through the use of simulation, you will be required to enter the correct commands to run the MOLTS tests on the main frame components of the H6000 computer.

TIME: Approximately 2.5 hours of on-line and 2.0 off-line time are required to complete TOL030.

4.2.13 Management Data Query (MDQS) Basic User (H6000-CDT) Course.

4.2.13.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135C 000; it is titled "Management Data Query System (MDQS) Basic User (H6000-CDT)"; and it bears PDS Code X9A.

4.2.13.2 Purpose. This course is designed for a wide variety of trainees, ranging from secretarial clerks and managers who may not have an in-depth knowledge of data processing concepts to experienced computer programmers. Basically, the course teaches concepts, capabilities, and terminology of MDQS; the MDQS Basic Control Language (BCL); Query Procedure Language (QPL); Elementary Procedure Language (EPL); and the MDQS Report Package. After completing the course, the trainee will be able to code and execute basic MDQS procedures to retrieve, sort, and perform calculations on elements that have been extracted from a data base and place the information in report format.

4.2.13.3 Management Data Query System (MDQS) Basic User (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the MDQS Basic User course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.13.4 Prerequisites. Prior to entering this course, the trainee must possess a working knowledge of the H6000 time-sharing system to include the CARDIN and ACCESS subsystems. This requirement may be satisfied by completing CDT lessons TUC010 through TUC041 of the Terminal User Course.

4.2.13.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 22 of this manual. In addition to the exhibits, the Honeywell Data Management System User's Mini-Guide is required for reference while at the terminal and for reading assignments between lessons. During the course, the trainee will be directed by the computer to appropriate exhibits and reading assignments.

4.2.13.6 Pattern of the Course. The MDQS Basic User course consists of eight lessons which should be executed consecutively:

(1) MDQ010--Introduction to MDQS - Basic Concepts and Terminology
(2) MDQ020--MDQS Command Language
(3) MDQ030--Identifiers and Conditionals
(4) MDQ040--Query Procedure Language (QPL)
(5) MDQ050--Elementary Procedure Language (EPL)
(6) MDQ060--The MDQS Report Package
(7) MDQ070--Statistical Functions and Control Segments
(8) MDQ080--MDQS Basic Control Language (BCL)

The instruction in this course is self-paced,a ctual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Forty-two (42) hours are used for course length and diploma credit. training and 21.0 hours of off-line training.

4.2.13.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.13.8 Lessons Available. The following is a list of the available lessons, together with an estimate of the time required for completion:

a. MDQ010. This lesson presents an introduction to MDQS, and teaches basic data base terminology and concepts as well as basic MDQ terminology and concepts. Data base structures and H6000 file organization methods are discussed as well as MDQS control files and related features.

TIME: Approximately 3.0 hours of on-line time are required to complete MDQ010.

b. MDQ020. This lesson describes the purpose of MDQS Commands and how to use them. The commands discussed are RUN, CHECK, WWTUT, DJST, and REDIRECT. In addition to the commands, this lesson also teaches the Application Definition File Query (ADFQ) subsystem. The trainee learns how to use the ADFQ to determine the composition of a data base as defined by the object Application Definition File (ADF). This lesson assigns MDQ PROJECT 1.

TIME: Approximately 2.5 hours of on-line and 3.5 hours off-line time are required to complete MDQ020.

c. MDQ030. This lesson begins to develop common portions of different statements that will be used in later lessons. Specifically, the lesson teaches simple item and variable identifiers, report element descriptors (Column, Title, and Justified Clauses), and conditionals. Through simulation, the trainee will create clauses that will be part of QPL, EPL, and BCL statements.

TIME: Approximately 2.0 hours of on-line time are required to complete MDQ030.

d. MDQ040. This lesson teaches the trainee how to write an MDQS procedure (program) using Query Procedure Language (QPL) which will retrieve, sort, and print information from a data base. The statements discussed are QUERY, SORT, PRINT, LET, REMARK, and END. The trainee will build one simulated procedure on-line and then will be assigned five problems to be accomplished off-line. The five off-line problems will process actual information from a sample data base on the site computer. (The sample data base is furnished as part of the Computer Directed Training System.) This lesson assigns MDQS PROJECT 2.

TIME: Approximately 3.0 hours of on-line and 3.5 hours of off-line time are required to complete MDQ040.

e. MDQ050. This lesson teaches the trainee how to utilize Elementary Procedure Language (EPL) to create a more comprehensive MDQS procedure than in the previous lesson. The EPL statements discussed include INVOKE, RETRIEVE, SORT, PRINT, WRAP-UP, and END. Additionally, the trainee learns the concept of "event segments" and how to use them to control the flow of execution. Several EPL procedures are created on-line and two procedures are assigned to be accomplished off-line. This lesson assigns MDQS PROJECT 3.

TIME: Approximately 2.5 hours of on-line and 3.5 hours of off-line time are required to complete MDQ050.

f. MDQ060. This lesson presents material on the MDQS report package. The REPORT statement with most of its clauses (Column Spacing, Page Length, etc.) is covered as well as the LINE and SPACE statements. At the conclusion of the lesson, the trainee is assigned a project containing a report package. This lesson assigns MDQS PROJECT 4.

TIME: Approximately 3.0 hours of on-line and 3.5 hours of off-line time are required to complete MDQ060.

g. MDQ070. This lesson presents material on the control headings and control footings of the report package, the MDQS statistical functions and the control segment. In conjunction with the control segment, the CONTROL and ENABLE statements are discussed as well as the ENABLING clause of the LET statement. At the conclusion of the lesson, a trainee project is assigned which contains a report package, a statistical function, a control segment, and a wrap-up segment. This lesson assigns MDQS PROJECT 5.

TIME: Approximately 2.5 hours of on-line and 3.5 hours of off-line time are required to complete MDQ070.

h. MDQ080. In this lesson, the trainee learns how to use the Basic Control Language (BCL) subsystem to generate MDQS procedures. The CHECK, LIST, RESAVE, RUN, and SAVE commands are discussed. By simulating the BCL subsystem, CDT allows the trainee to experience entering the BCL subsystem, executing the BROWSE function, and submitting responses to the BCL dialogue. The trainee will be assigned three problems that are to be solved using the BCL subsystem. This lesson assigns MDQS PROJECT 6.

TIME: Approximately 2.5 hours of on-line and 3.5 hours of off-line time are required to complete MDQ080.

4.2.14 Indexed Sequential Processor (ISP) (H6000-CDT) Course.

4.2.14.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135C 002; it is titled "Indexed Sequential Processor (ISP) (H6000-CDT)"; and it bears PDS Code YCH.

4.2.14.2 Purpose. This course is designed to provide WWMCCS programmers and analysts with a working knowledge of Indexed Sequential Processor system. Basically, the course teaches the ISP programming language; however it combines instruction on the language with the "why's" and "how's" of the ISP system. After completing this course, the trainee should be able to encode, compile, and execute the ISP to create, maintain, and retrieve data.

4.2.14.3 Indexed Sequential Processor (ISP) (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the Indexed Sequential Processor (ISP) course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.14.4 Prerequisites. Trainees entering this course must have a working knowledge of COBOL as implemented on the H6000 series computers. Those who enter this course without experience in the actual writing and running of COBOL programs will find their background is probably inadequate. Prior to starting this course, the trainee should also complete one of the lessons that teach how to operate the remote terminal and interact with CDTS. Persons who will be taking the course via teletype should first complete lesson TUC010, those who are using VIP786 should take TUC011, and those using VIP7705 should take TUC012.

4.2.14.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 23 of this manual. The various ISP lessons will direct the trainee to the appropriate exhibit when it is needed.

4.2.14.6 Pattern of the Course. The Indexed Sequential Processor (ISP) course consists of five lessons which should be executed consecutively:

(1) ISP010--ISP Concepts, Capabilities, and Formats
(2) ISP020--ISP Data Base Load
(3) ISP030--ISP Data Base Retrieval
(4) ISP040--ISP Data Base Modification
(5) ISP050--ISP Utilities

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Thirty-one and a half (31.5) hours are used for course length and diploma credit.

4.2.14.7 Last Frame Labels. The last frame of each lesson in this course is labeled END.

4.2.14.8 Lessons Available. Following is a list of the available lessons and the time required to complete each:

a. ISP010. This lesson presents the basic concepts, capabilities, and formats of ISP. The trainee will learn the two access methods utilized by

ISP; they are the LINKED-LIST and the INDEXED ORGANIZATION. This lesson further teaches the six different formats within ISP which are as follows: (1) DATA FILE, (2) DATA PAGE, (3) DATA RECORD, (4) INDEX FILE, (5) INDEX PAGE, and (6) INDEX RECORD.

TIME: Approximately 1.5 hours of on-line and 1.0 hour off-line time are required to complete ISP010.

b. ISP020. The purpose of this lesson is to teach the four ISP subroutines concerned with loading an ISP data base; they are:

(1) NOPEN--used to open a new ISP data base.

(2) FILINT--used to write a new ISP data record or index record to the ISP data base.

(3) ICLOSE--used to close a specific ISP data base.

(4) WRAP-UP--used to close all ISP data bases.

TIME: Approximately 2.0 hours of on-line and 5.0 hours off-line time are required to complete ISP020.

c. ISP030. This lesson is concerned with retrieval of data from an ISP data base. It fully explains the two retrieval methods, sequential and random. The ISP subroutines taught in this lesson which perform data retrieval are:

(1) IOPEN--used to open an existing ISP data base.

(2) GETSEQ--used to sequentially retrieve ISP data.

(3) GETRAN--used to randomly retrieve ISP data.

TIME: Approximately 1.5 hours of on-line and 6.5 hours of off-line time are required to complete ISP030.

d. ISP040. This lesson teaches the ISP subroutines necessary to modify an existing ISP data base. These subroutines are:

(1) ADDREC--used to add new ISP records to an existing ISP data base.

(2) DELREC--used to delete existing records from an ISP data base.

(3) CHGREC--used to change fields within existing ISP records in an ISP data base.

TIME: Approximately 1.5 hours of on-line and 5.0 hours off-line time are required to complete ISP040.

e. ISP050. This lesson teaches three additional ISP subroutines and two ISP utilities. The additional subroutines are:

(1) XREWND--used to reposition the current record pointer within the ISP data base.

(2) XSTAT--used to obtain the ISP data base statistics.

(3) XFLUSH--used to clear ISP record buffers by writing the buffer contents to an ISP data base.

Utilities are used by the Indexed Sequential Processor to either reorganize an existing ISP data base or to reconstruct a destroyed ISP data base. The ISP utility commands to perform these functions are:

(1) XUTIL--used to reorganize an existing ISP data base by removing logically deleted records and replacing overflow records into their proper position. (Causes the data base to be physically and logically organized sequentially.)

(2) XJRNAL--used to reconstruct a destroyed ISP data base from a JOURNAL tape.

TIME: Approximately 2.5 hours of on-line and 5.0 hours off-line time are required to complete ISP050.

4.2.15 General Macro Assembly Programming (GMAP) (H6000-CDT) Course.

4.2.15.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5135A 001; it is titled "General Macro Assembly Programming (GMAP) (H6000-CDT)"; and it bears a PDS Code VWU.

4.2.15.2 Purpose. This course is designed to provide programmers and system analysts with a working knowledge of the basic GMAP instruction set and how to use it. It also includes some of the most common system macro instructions and subroutines. After completing this course, the trainee should be able to code, assemble, and execute a simple GMAP program or subroutine and interpret its output listings.

4.2.15.3 General Macro Assembly Programming (GMAP) (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the General Macro Assembly Programming (GMAP) course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.15.4 Prerequisites. Trainees entering this course should have a working knowledge of some programming language as run on the H6000 series computers. Prior to starting this course, the trainee should have completed one of the lessons that teaches how the remote terminals interact with CDTS. Persons who will be taking the course via teletype should first complete lesson TUC010, those who are using VIP786 should take TUC011, and those using VIP7705 should take TUC012.

4.2.15.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 11 of this manual. The various GMAP lessons will direct the trainee to the appropriate exhibit when it is needed.

<u>4.2.15.6 Pattern of the Course.</u> The General Macro Assembly Programming (GMAP) course consists of 22 lessons which should be executed consecutively:

(1) GMP010--Features of H6000 Hardware/Software
(2) GMP020--User/System File Structure
(3) GMP030--Job Control Language and Program Deck Setup
(4) GMP040--Machin‿ Word and Register Structures
(5) GMP050--Instruction and Data Word Formats
(6) GMP060--Storage Allocating and Data Generating Pseudo-Operations
(7) GMP070--Data Movement Instructions (Loads and Stores)
(8) GMP080--GMAP "Shift" and "Rotate" Instruction Sets
(9) GMP090--Integer Addition
(10) GMP095--Integer Subtraction and Negate Instructions
(11) GMP100--Execution Reports, Assembly Listings, and Dumps
(12) GMP110--GMAP Programming Project I (Parts 1 and 2)
(13) GMP120--Introduction to the General Loader and the Base Address Register
(14) GMP130--Compare Instructions
(15) GMP140--Transfer of Control Instructions
(16) GMP150--Address Modification
(17) GMP160--Boolean Operations
(18) GMP170--GMAP Programming Project II (Program Looping)
(19) GMP180--Elementary Macro Operations
(20) GMP190--Basic Subroutine Linkage
(21) GMP200--Input-Output Operations Using File and Record Control Subroutines
(22) GMP210--GMAP Programming Project III (Input-Output Programming)

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Seventy (70) hours are used for course length and diploma credit.

<u>4.2.15.7 Last Frame Labels.</u> The last frame of each lesson in this course is labeled END.

<u>4.2.15.8 Lessons Available.</u> Following is a list of the available lessons and the time required to complete each one:

a. GMP010. This lesson will give you a basic understanding of the major hardware and software components of the H6000 computer.

TIME: Approximately 2.0 hours of on-line time are required to complete GMP010.

b. GMP020. This lesson will teach you how to differentiate between user and system files. It will also explain the function of all the files associated with a GMAP program.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP020.

c. GMP030. This lesson goes into a detailed explanation of the H6000 Job Control Language (JCL). It also has an option to allow the trainee to only cover those JCL cards which are normally used by a GMAP programmer.

TIME: Approximately 9.0 hours of on-line time (3.0 hours for the GMAP JCL) are required to complete GMP030.

d. GMP040. This lesson covers the arithmetic processes used by the computer. It also goes into detail about the H6000 computer word structure and the structure of the processor registers.

TIME: Approximately 2.0 hours of on-line time are required to complete GMP040.

e. GMP050. This lesson discusses the GMAP coding format and the data and instruction word formats. I. also explains the two phases of the GMAP assembler.

TIME: Approximately 1.0 hour of on-line time is required to complete GMP050.

f. GMP060. This lesson covers the GMAP pseudo-operations (pseudo-ops) which create numeric constants, alphanumeric character strings, and allocate data storage areas.

TIME: Approximately 2.0 hours of on-line time are required to complete GMP060.

g. GMP070. This lesson explains the primary data movement instructions. Loads copy data from memory to a register, and stores copy data from a register to memory.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP070.

h. GMP080. This lesson discusses the shift and rotate instructions. These instructions allow programmers to manipulate data within a register.

TIME: Approximately 2.0 hours of on-line time are required to complete GMP080.

i. GMP090. This lesson teaches how to do addition, using integer numbers.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP090.

j. GMP095. This lesson concludes your study of the integer arithmetic instructions with the subtract and negate instructions.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP095.

k. GMP100. This lesson explains the various reports produced by a GMAP program being assembled and executed. It goes into a detailed explanation of the execution report and the assembly listing. It also explains how to find things in the dump by using the memory map.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP100.

l. GMP110. This lesson assigns a two-part project. The trainee will be required to use the basic pseudo-ops, data movement, and integer arithmetic instructions. This lesson assigns GMAP PROJECT 1.

TIME: Approximately 7.50 hour of on-line time is required to complete GMP110.

m. GMP120. This lesson discusses the basic process required for the general loader to prepare a program for execution, control cards affecting the general loader, and the purpose and use of the base address register.

TIME: Approximately 2.25 hours of on-line time are required to complete GMP120.

n. GMP130. This lesson covers the GMAP compare instructions. It discusses the difference between algebraic and logical comparisons.

TIME: Approximately 1.25 hours of on-line time are reouired to complete GMP130.

o. GMP140. This lesson presents ten transfer instructions. These instructions cause transfer of control within the program based upon the settings in the indicator register.

TIME: Approximately 2.0 hours of on-line time are required to complete GMP140.

p. GMP150. This lesson teaches how to use index registers to modify addresses. It also teaches the looping process.

TIME: Approximately 2.5 hours of on-line time are required to complete GMP150.

q. GMP160. This lesson covers the various Boolean operations. They include the AND, OR, Exclusive OR, and "Boolean compare" instructions.

TIME: Approximately 3.0 hours of on-line time are required to complete GMP160.

r. GMP170. This lesson modifies the project from GMP110. The trainee will be required to use compare and transfer instructions and address modification. This lesson assigns GMAP PROJECT 2.

TIME: Approximately 7.5 hour of on-line time is required to complete GMP170.

s. GMP180. This lesson presents the GMAP macro function. Both the macro prototypes and the macro operation statements will be discussed.

TIME: Approximately 1.25 hours of on-line time are required to complete GMP180.

t. GMP190. This lesson explains the open, closed internal, and closed external types of subroutines. It also shows the use of the CALL pseudo-op and the instructions it generates.

TIME: Approximately 2.25 hours of on-line time are required to complete GMP190.

u. GMP200. This lesson demonstrates the use of the four most common file and record control subroutines (GET, PRINT, OPEN, and CLOSE).

TIME: Approximately 2.5 hours of on-line time are required to complete GMP200.

v. GMP210. This lesson gives you a project to do, requiring you to use the file and record subroutines. This lesson assigns GMAP PROJECT 3.

TIME: Approximately 7.5 hour of on-line time is required to complete GMP210.

4.2.16 H700 Test and Diagnostics (H6000-CDT) Course.

4.2.16.1 Course Number, Title, and PDS Code. This course is numbered ECDTS51130 001; it is titled "H700 Test and Diagnostics (H6000-CDT)"; and it bears PDS Code YSK.

4.2.16.2 H700 Test and Diagnostics (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the H700 Test and Diagnostics course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.16.3 Prerequisites. Trainees entering this course should be familiar with the standard program loading procedures and alsc be familiar with the H700 control panel. Prior to starting this course, the trainee should have completed one of the lessons that teaches how the remote terminals interact with CDTS. Persons who will be taking the course via teletype should first complete lesson TUC010, those who are using VIP786 should take TUC011, and those using VIP7705 should take TUC012.

4.2.16.4 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 12 of this manual. The Test and Diagnostics lessons will direct the trainee to the appropriate exhibit when it is needed.

4.2.16.5 Pattern of Course. The H700 Test and Diagnostics course consists of one lesson--TAD700--H700 Test and Diagnostics Seminar. Instruction consists of 5.0 hours of on-line terminal time.

4.2.16.6 Last Frame Labels. The last frame of the lesson in this course is labeled END.

4.2.16.7 Lessons Available. Following is the available lesson and the time required to complete it: TAD700. This lesson is designed to familiarize both EDP managers and computer operators with the capabilities and advantages of H700 Test and Diagnostics.

TIME: Approximately 5.0 hours of on-line time are required to complete TAD700.

4.2.17 Introduction to Data Processing (H6000-CDT) Course.

4.2.17.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5131B 002; it is titled "Introduction to Data Processing (H6000-CDT)"; and it bears PDS Code VWT.

4.2.17.2 Purpose. This course is designed as an orientation course for personnel coming into the data processing field with little or no knowledge of computer systems or data processing.

4.2.17.3 Introduction to Data Processing (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the Introduction to Data Processing Course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.17.4 Prerequisites. Trainees entering this course should have taken CDT lessons TUC010, TUC011, or TUC012, or have equivalent experience.

4.2.17.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 24 of this manual. The Introduction to Data Processing lessons will direct the trainee to the appropriate exhibit when it is needed.

4.2.17.6 Pattern of the Course. The Introduction to Data Processing course consists of five lessons which should be executed consecutively:

(1) IDP010--Introduction to Basic Concepts of Data Processing
(2) IDP020--Introduction to Computer Hardware
(3) IDP030--Introduction to Data Description, Programming Languages, and Number Systems
(4) IDP040--Introduction to Problem Solving
(5) IDP050--Overview of Honeywell Information System H6000 Series Hardware

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Eleven and a half (11.5) hours are used for course length and diploma credit.

4.2.17.7 Last Frame Labels. The last frame of each lesson in this course is labeled END.

4.2.17.8 Lessons Available. Following is a list of the available lessons and the time required to complete each:

a. IDP010. This lesson provides an introduction to the basic concepts of data processing, including the development of the term "data processing," the steps in the data processing cycle, and the various methods of data processing.

TIME: Approximately 2.0 hours of on-line time are required to complete IDP010.

b. IDP020. This lesson provides an introduction to computer hardware, including the evolution of computer development, classifications of computers, and the basic elements of digital computers.

TIME: Approximately 2.0 hours of on-line time are required to complete IDP020.

c. IDP030. This lesson provides an introduction to data descriptors, characteristics of programming languages, and general information on numbering systems.

TIME: Approximately 2.0 hours of on-line time are required to complete IDP030.

d. IDP040. This lesson provides an introduction to the methods of problem solving, types of algorithms, and top down structured programming design and techniques.

TIME: Approximately 3.0 hours of on-line time are required to complete IDP040.

e. IDP050. This lesson provides a general overview of the WWMCCS Electronic Data Processing (EDP) hardware. The features, characteristics and capabilities of major H6000 Series hardware components and devices are explored.

TIME: Approximately 2.5 hours of on-line time are required to complete IDP050.

4.2.18 WWMCCS Intercomputer Network (WIN) User (H6000-CDT) Course.

4.2.18.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5116 003; it is titled "WWMCCS Intercomputer Network (WIN) (H6000-CDT)"; and it bears PDS Code 11J.

4.2.18.2 Purpose. This course is designed to teach the user how to use WIN software in Standard WWMCCS Applications.

4.2.18.3 WWMCCS Intercomputer Network (WIN) Course Material OPR. Recommendations for improving the instructional materials for the WIN course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.18.4 Prerequisites. Trainees entering the course should have taken CDT lessons TUC010, TUC011, or TUC012, or have similar H6000 TSS experience.

4.2.18.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits. (The WIN Terminal User's Pocket Guide can be substituted for several lessons.) The WIN exhibits are found in Attachment 25 of this manual. The WIN course will direct the trainee to the appropriate exhibit when needed.

4.2.18.6 Pattern of the Course. The WIN course consists of 15 lessons which should be executed consecutively:

SECTION I--Teleconferencing (TLCF)

(1) WIN010--Introduction to WIN
(2) WIN020--Joining a Conference
(3) WIN030--The Listen Mode
(4) WIN040--The Talk Mode
(5) WIN050--The Command Mode
(6) WIN060--Chairman Responsibilities
(7) WIN070--TELNET
(8) WIN080--Teleconferencing Simulation

SECTION II--File Transfer Service (FTS)

(9) WIN090--FTS Introduction
(10) WIN100--FTS Mail and Readmail
(11) WIN110--FTS File Transfer (Disk)
(12) WIN120--FTS Service Commands
(13) WIN130--FTS Simulation
(14) WIN140--FTS Netjob (Optional lesson)
(15) WIN150--FTS Command Stream (Optional lesson)

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Seventeen (17) hours are used for course length and diploma credit.

4.2.18.7 Last Frame Labels. The last frame of each lesson in this course is labeled END.

4.2.18.8 Lessons Available. Following is a list of the available lessons and the time required to complete each:

a. WIN010. This lesson serves as an introduction to WIN. It describes how teleconferencing (TLCF), TELNET and File Transfer Service (FTS) are used.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN010.

b. WIN020. This lesson teaches the student how to log on to Teleconferencing and join a conference. A log on simulation is included.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN020.

c. WIN030. This lesson explains how to use the Listen Mode of Teleconferencing. It also has a section on formatting conventions.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN030.

d. WIN040. This lesson teaches the concepts and commands used in the Talk Mode of Teleconferencing.

TIME: Approximately 1.5 hour of on-line time is required to complete WIN040.

e. WIN050. This lesson teaches concepts of the Command Mode as well as participant commands.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN050.

f. WIN060. This lesson teaches the procedures for initiating and chairing a conference. The lesson shows how to use "chairman only" commands and it also has an optional chairman initiation simulation.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN060.

g. WIN070. This lesson teaches the concepts and commands required for using TELNET.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN070.

h. WIN080. This lesson simulates logging on to TELNET and joining a Teleconference at a remote host. The simulation is based on the knowledge taught in the previous seven lessons.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN080.

i. WIN090. This lesson introduces the student to the capabilities of the File Transfer Service (FTS), modes of operation, and types of commands used in FTS.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN090.

j. WIN100. This lesson teaches the student how to send mail messages to other WIN users and how to read and/or manipulate the contents of his/her mailbox.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN100.

k. WIN110. This lesson teaches the procedures for using FTS to transfer disk-to-disk files. The student learns capabilities of SEND, REPLACE, and APPEND for transferring files.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN110.

l. WIN120. This lesson introduces the FTS service commands which are auxiliary directives used to control the FTS session and are not executable.

TIME: Approximately 1.5 hours of on-line time are required to complete WIN120.

m. WIN130. This lesson simulates logging on to FTS and performing various tasks as taught in the previous four lessons.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN130.

n. WIN140. This optional lesson teaches the concepts and commands required for using FTS Netjob.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN140.

o. WIN150. This optional lesson teaches the procedures for using FTS Command Stream.

TIME: Approximately 1.0 hour of on-line time is required to complete WIN150.

4.2.19 Graphics Information Presentation System (GIPSY) (H6000-CDT) Course.

4.2.19.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5116 002; it is titled "Graphics Information Presentation System (GIPSY) (H6000-CDT)"; and it bears PDS Code 11H.

4.2.19.2 Purpose. The course is designed to teach the user about the software needed to use GIPSY commands.

4.2.19.3 Graphics Information Presentation System (GIPSY) (H6000-CDT) Course Material OPR. Recommendations for improving the instructional materials for the Graphics Information Presentation System (GIPSY) course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.19.4 Prerequisites. Trainees entering the course should have taken CDT lessons TUC010, TUC011, or TUC012, or have similar H6000 TSS experience.

4.2.19.5 Use of Exhibits. A trainee cannot successfully complete this course without the exhibits found in Attachment 26 of this manual. The GIPSY course will direct the trainee to the appropriate exhibit when needed.

4.2.19.6 Pattern of the Course. The Graphics Information Presentation System Course consists of 13 lessons which should be executed consecutively.

(1) GIP010--Introduction
(2) GIP020--Instruction Formats
(3) GIP030--Basic Commands
(4) GIP040--Data Retrieval
(5) GIP050--Building Tabular Reports
(6) GIP051--WWMCCS Standard Graphics Terminal (WSGT) Operations
(7) GIP060--Displaying and Modifying Reports
(8) GIP070--Displaying Graphs and Curves
(9) GIP080--Building and Displaying Geographic Reports
(10) GIP090--Additional Geographic Report Display Options
(11) GIP100--Producing Symbol Plots from User Data Files
(12) GIP110--Producing Track Plots from User Data Files and Listing the Data Used
(13) GIP120--Interactive Display Building and Modification

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Twenty-eight (28) hours are used for course length and diploma credit.

4.2.19.7 Last Frame Labels. The last frame of each lesson in this course is labeled END.

4.2.19.8 Lessons Available. Following is a list of the available lessons and the time required to complete each.

a. GIP010. This lesson gives an introduction to computer graphics. It also describes the GIPSY graphics system and gives an outline for the entire course.

TIME: Approximately 0.5 hour of on-line time is required to complete GIP010.

b. GIP020. This lesson describes the symbols used to define the GIPSY commands and the log-on procedures needed to get into the GIPSY system. The student is given a chance to practice entering the GIPSY system through a log-on simulation.

TIME: Approximately 1.0 hour of on-line time is required to complete GIP020.

c. GIP030. This lesson teaches the basic GIPSY commands TITLE, CLASS, and SET. It also teaches how to insert comments within the GIPSY commands for documentation. Finally, the student is introduced to special commands called INTERRUPT commands.

TIME: Approximately 1.5 hours of on-line time are required to complete GIP030.

d. GIP040. This lesson teaches the commands used to create the data files that are needed to produce reports, graphs, and maps. The following commands are covered: BUILD FDT, FDT, FILE, RETRIEVE, SAVE, and RESAVE. This lesson assigns GIPSY PROJECT 1.

TIME: Approximately 2 hours of on-line and 1.0 hour off-line time are required to complete GIP040.

e. GIP050. This lesson teaches the commands used to produce tabular reports. The following commands are taught: BUILD TABULAR REPORT, CALCULATE DATA, LOGIC TABLE, and MATH TABLE. This lesson assigns GIPSY PROJECT 2.

TIME: Approximately 1.5 hours of on-line and 2.0 hours off-line time are required to complete GIP050.

f. GIP051. This lesson teaches the booting procedures for WWMCCS Standard Graphics Terminal (WSGT). It also covers the procedures for linking the WSGT with H6000 computer.

TIME: Approximately 1.0 hour of on-line time is required to complete GIP051.

g. GIP060. This lesson teaches the student how to display a report that has already been created and how to make modifications to it. This lesson assigns GIPSY PROJECT 3.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP060.

h. GIP070. This lesson teaches the commands needed to display graphs and plots on the terminal. This lesson assigns GIPSY PROJECT 4.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP070.

i. GIP080. This lesson teaches the basic commands needed to build and display geographic reports. This lesson assigns GIPSY PROJECT 5.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP080.

j. GIP090. This lesson teaches WINDOW and LOCATION tables, the GRID and ZOOM commands, and the geographic options of the SET command. This lesson assigns GIPSY PROJECT 6.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP090.

k. GIP100. This lesson teaches how to produce maps that have symbols and circles on them by using information from the user's data file. This lesson assigns GIPSY PROJECT 7.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP100.

l. GIP110. This lesson teaches how to use information from a user's data file to produce maps that have tracks and how to list the data used for symbols or tracks. This lesson assigns GIPSY PROJECT 8.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP110.

m. GIP120. This lesson teaches how to modify a map that has been displayed by adding symbols, tracks, circles, the coordinate location of points, and the azimuth time and distance between points. It also teaches how to remove information from a map. This lesson assigns GIPSY PROJECT 9.

TIME: Approximately 1.5 hours of on-line and 1.0 hours off-line time are required to complete GIP120.

4.2.20 Basic Language Programming (BSC) (H6000-CDT) Course.

4.2.20.1 Course Number, Title, and PDS Code. This course is numbered ECDTS5131B 006; it is titled "Basic Language Programming (BSC) (H6000-CDT)"; and it bears PDS code 4JS.

4.2.20.2 Purpose. This course is designed for use by computer programmers. The course can be used to provide a working knowledge of BASIC to trainees who are unfamiliar with the language.

4.2.20.3 BASIC Language Programming (H6000-CDT) Course Material OPR. Recommendations for improvement of the course material in the BASIC Language Programming course should be sent to 3300 TCHTW/TTGXZ, Attn: WWMCCS CDT, Keesler AFB, MS 39534.

4.2.20.4 Prerequisites. In terms of training, the prerequisite for this course is the completion of Terminal USER course, Lessons TUC010, TUC011, or TUC012 or have equivalent experience.

4.2.20.5 Use of Exhibits. One cannot successfully complete this course without the use of exhibits found in Attachment 27 of this manual. The BASIC Language Programming lesson will direct the person to the appropriate exhibits for each lesson.

4.2.20.6 Pattern of the Course. The BASIC Language Programming course consists of 6 lessons which should be executed consecutively:

(1) BSC010 -- Introduction to BASIC
(2) BSC020 -- Data Input and Branching
(3) BSC030 -- BASIC Loops and Arrays
(4) BSC040 -- MAT Statements and Subroutines
(5) BSC050 -- Math, print and other Functions
(6) BSC060 -- Strings and String Functions

The instruction in this course is self-paced, actual completion time may vary. However, all lessons and lesson projects not indicated as optional are required. Fifteen (15) hours are used for course length and diploma credit.

4.2.20.7 Last Frame Labels. The last frame in each lesson is labeled END.

4.2.20.8 Lessons Available. Following is a list of the available lessons, together with an estimate of the time required for completion.

a. BSC010. This lesson in BASIC (Beginners All Purpose Symbolic Instruction Code) discusses the Hardware and Software components of a computer along with the similarities between BASIC and the English language. It explains PRINT, STOP and END statements and describes how to form expressions in BASIC.

TIME: Approximately 2.0 hours of on-line time are required to complete BSC010.

b. BSC020. This lesson teaches the student how to interact with the BASIC compiler by using the Input, Read, and Data statements. It introduces Branching concepts by describing the IF and GOTO statements. This lesson assigns BASIC Projects 1 and 2.

TIME: Approximately 2.0 hours of on-line and 1.5 hours off-line time are required to complete BSC020.

c. BSC030. This lesson describes the FOR and NEXT statements which will cause a program to execute loops. It also describes Arrays which are used to expand the storage capacity of variables.

TIME: Approximately 2.0 hours of on-line time are required to complete BSC030.

d. BSC040. This lesson covers Matrix operations (Array operators) and subroutines.

TIME: Approximately 2.0 hours of on-line time are required to complete BSC040.

e. BSC050. This lesson covers Library and User-defined Functions. It describes which are are available and how to create new ones.

TIME: Approximately 2.0 hours of on-line time are required to complete BSC050.

f. BSC060. This lesson reviews what a string is. It also describes various string functions available in BASIC, what they are used for, and how they can make programming easier. This lesson assigns BASIC PROJECTS 3 or 4 and 5.

TIME: Approximately 2.0 hours of on-line and 1.5 hours off-line time are required to complete BSC060.

BY ORDER OF THE SECRETARY OF THE AIR FORCE

OFFICIAL

CHARLES A. GABRIEL, General, USAF
Chief of Staff

JAMES H. DELANEY, Colonel, USAF
Director of Administration

SUMMARY OF CHANGES

This manual has been completely revised to include revisions on existing course material, and the deletion of Section 5. New lessons for the Terminal User, Introduction to H6000 Series computer Programming, Integrated Data Store, COBOL Programming and Introduction to Data Processing have been added. A new six lesson course on Basic Language Programming has also been added and the WWMCCS System Software Analyst Course was deleted.

COMPUTER DIRECTED TRAINING SYSTEM (CDTS) PROCEDURES REFERENCE

TABLE OF CONTENTS

A1.1 PURPOSE OF REFERENCE. The following guide is provided so that users of this manual, specifically Training Systems Managers (TSMs) and Training Monitors (TMs), will have an easy to read and follow reference to Computer Directed Training System procedures.

A1.2 BRIEF EXPLANATION OF ATTACHMENTS.

a. Attachment 1 - Computer directed Training System Procedures Reference. This attachment contains a guide to attachments 1 through 9.

b. Attachment 2 - Training System Manager Handbook. This attachment outlines and describes the responsibilities of the Training System Manager (TSM) and the Training Monitor (TM).

c. Attachment 3 - Procedures for Restoring CDTSCDTS. This attachment gives specific procedures for restoring the CDTS release tapes.

d. Attachment 4 - CDTS Commands, Executive Messages and Lesson Execution. This attachment gives information for executing lessons on the Computer Directed Training System.

e. Attachment 5 - CDTS Student Records Summary Program and Building of Trace Tapes. This attachment describes the collection of student records and the requirement for submission of trace tapes.

f. Attachment 6 - Development Center Trace Tape Processing. This attachment describes the procedures followed by the CDTS Development Center at Keesler AFB in processing trace tapes.

g. Attachment 7 - Printing "Hard" Copies of Edited CDTS Lessons. This attachment gives specific procedures for printing edited copies of selected CDTS lessons.

h. Attachment 8 - CDTS Lesson Development. This attachment gives a description of how to write, compile, debug and load lessons for use on the Computer Directed Training System.

i. Attachment 9 - Development Center Processing Inventory. This attachment gives a breakdown of the two object libraries on the CDTS release tapes.

TRAINING SYSTEM MANAGER (TSM) HANDBOOK

Contents

FIGURES

A2.1 GENERAL DESCRIPTION.

A2.1.1 Purpose of the Training Systems Manager Handbook. This handbook provides information pertaining to the additional duties of the Training Systems Manager (TSM) and the Training Monitor (TM) for operation of CDTS at on-site locations. This handbook serves only as a recommended guideline for the TSM/TM and is not limited to these particular responsibilities.

A2.1.2 Responsibilities.

a. Using Site. A TSM should be appointed at each site that uses CDTS. This responsibility is intended to be an additional duty.

b. Appointment of the TSM. The appointing authority should give careful consideration to selecting a Training System Manager. The individual selected will have the overall responsibility for CDTS training at the site. Therefore, the TSM must be knowledgeable in the various tasks required. Figure A2-1 illustrates the position in the flowchart. These additional duties are discussed in this handbook.

A2.2 TRAINING SYSTEMS MANAGER (TSM) RESPONSIBILITIES.

A2.2.1 On-Site CDTS Training. As the newly appointed TSM, it is recognized that H6000 computer sites have personnel who require additional training. These training needs may be in the functional area, such as computer operator, system analyst, or computer programmer. It is your responsiblity, after identifying these individuals, through their supervisors, to see that they receive the additional training necessary.

a. Material Support. To accomplish the on-site training needs of your personnel, there are other areas you must become familiar with. You must know the number of terminals (TTYs or VIPs) that are available to your site and how to obtain materials, such as AF manuals, Honeywell manuals, handouts, diplomas, etc. used to support the training. These materials must be stored and controlled by you.

b. Personnel Support. You must establish contact with all qualified or appropriate personnel within each functional area. It is of extreme importance that you maintain a good working relationship with them. Some benefits that may be derived from a good working relationship are: insurance that all available software and courseware materials are loaded into the computer; notification whenever releases pertaining to CDTS are received; and insurance that remotes and computer time are made available for training. These are but a few of the many benefits you may obtain from support personnel.

A2.2.2 Appointing Training Monitor (TM). When you determine that assistance is needed to help monitor training (large sites) other than normal supervisor assistance, you may select a Training Monitor (TM). You, as the TSM, will have to decide the criteria to use to make the selection. Here are some suggestions. With the supervisor's assistance, select the most qualified personnel within a functional area. Appoint one (1) per course (operator, analysis, etc.) as needed. The number needed at any one site will depend on you and your site training needs. Replacement of the TM, for any reason, is at your discretion.

A2.2.3 Consolidate Training Requirements. With the knowledge of the needs of your training site and appointment of the training monitors, you can consolidate your training requirements. Here are some suggestions.

a. Releases. Check with appropriate personnel for release of new lessons, courses, or any changes that might affect training.

(1) Insure that personnel involved with CDTS are familiar with its objectives.

(2) Review the cover sheet that acccmpanies the release to determine the extent of any changes.

(3) Release Tapes. The CDTS release consists of two tapes.

(a) A FILSYS SAVE of the UMC CDTSCDTS, contains the object form of the CDTS lessons. Refer to attachment 3 of this manual for infomation on the program to restore CDTSCDTS.

(b) A tape saved from a removable disk pack, contains the subcatalog CDTSCDTS/HCOPY which provides the hard copy capability. Refer to attachment 7 of this manual for information on the program to print hard copies of edited CDT lessons.

NOTE: In order to restore CDTSCDTS/HCOPY to a permanent pack instead of a removable pack, the RESET DEVICE or RESET RFILES option must be used. Refer to the Honeywell File Management Supervisor manual for further information.

b. Possible types of changes to look for.

(1) Changes to software.

(a) Log-on procedures.

(b) Device compatibility.

(c) Reporting format (VCSRSJ).

(2) Changes to courseware.

(a) Added courses or lessons.

(b) Changes to lessons.

(c) Deletion f courses or lessons.

(d) Changes in course exhibits (Reference attachments A10 through A27).

c. Assess the impact on the TM and/or students.

(1) Insure students complete their current lesson prior to loading the new release, as positional pointers to old lessons will be lost.

(2) What effect will the release have on the current training program (cancel MTTs, TDYs, etc.)?

(3) Will additional TMs be needed, relieved, or assigned?

d. Actions you may take as a result of releases.

(1) Assign new TMs as required.

(2) Notify TMs of new releases or changes.

(3) Coordinate update actions with the TM.

(4) Notify appropriate personnel (computer operators, etc.) to update system according to agreement between the TMs and you.

(5) Verify that the system has been updated as instructed.

(6) Notify TMs when update is completed.

e. Schedule Training.

(1) Establish a tentative schedule indicating periods when CDTS training can be accomplished.

(2) Insure that required lessons or courses are loaded by checking with the appropriate personnel.

(3) Coordinate with the appropriate offices to insure that remotes and on-line time have been made available.

(a) Determine the number of remotes available.

(b) Determine when CDTS can be available.

(c) Coordinate the available remotes with the available CDTS on-line time.

(d) Develop schedules and procedures for allocations of available CDTS training periods to TMs. As a general rule, a training period should be at least 45 minutes long.

(4) Resolve participation and scheduling problems between TMs.

(a) Insure that all student activities are coordinated between each TM and you. This includes the (1) available courseware and software for these activities, (2) number of trainees from each functional area that require training, (3) number starting, failing or completing courses, and (4) deletion of those trainees who fail to complete lessons in a prescribed length of time.

(5) Required material.

(a) See that required materials (manuals, handouts, diplomas, etc.) are available to the TM.

(b) Air Force manuals may be ordered through your local Publications Distribution Office (PDO).

(c) Handouts may be reproduced at your site.

(d) Diplomas may be requested as stated in paragraph 3e(8) below of this manual.

(e) Control and storage of most material should be at your discretion.

(f) Reproduction of course exhibits from AFM 50-752 is authorized.

(6) Provide TM support for tenant organizations if required.

(7) Certificate of completion (diploma). Upon application by the TM.

(a) Insure that students who successfully complete a CDTS course have their records certified by the TM.

(b) Fill out duplicate certificates of completion and give to the TM. (See figure A2-2 for course number, PDS code, and course length.)

(c) Insure that the TM advises the student that the duplicate copy must be furnished to the servicing personnel office for posting to the trainee's record.

(8) Ordering certificates of training (AF Form 1256). Army, Navy, or any other organizations wishing to issue this form to graduates of CDTS courses can obtain copies by sending a letter to:

AF Publications Distribution Center
Baltimore, MD 21220

The requestor should include the return address, together with a statement pertaining to the number of copies needed and an explanation whether the requirement is "recurring" or "one time." Another suggestion: If the TSM has no forms available, type a "Certificate of Completion," in duplicate. An example is shown in figure A2-3.

A2.2.4 Student Records Summary (VCSRSJ) Outputs. This program reads accumulated student records data output by the CDTS Executive and produces a report of student activity on CDTS. Detailed information about program VCSRSJ as well as the requirement, processing procedures and mailing address for trace tapes can be found in attachment 5 of this manual.

A2.2.5 Distribution of Reports. You, as the TSM, will have the responsibility of assisting in determining the need for and preparation of appropriate Incident Reports (IR) or Difficulty Reports (DIREP). Information pertaining to the distribution of these reports are found in paragraph 3.5b, and AFM 171-752, paragraph 2.7. Recommendations for improvement of course material for CDTS courses should be sent to:

3300 TCHTW/TTGXZ
Attn: WWMCCS CDT
Keesler AFB MS 39534

A2.3 TRAINING MONITOR (TM) RESPONSIBILITIES. The TSM at each WWMCCS site will appoint technically qualified individuals for the additional duty of training monitor (TM). These appointments may be by CDTS course, functional area, work center, or any other division considered appropriate. This additional duty includes, but is not limited to, the following responsibilities.

A2.3.1 Coordinating New CDTS Lessons or Changes. Check with the TSM for release of new lessons or changes that could affect CDTS training.

a. Assess changes to the training system.

(1) What will be the effect on students in the course?

(2) Would the student be allowed to continue the current lessons?

(3) Should course graduates reaccomplish this training?

b. Advise the student.

(1) To complete the lesson.

(2) Discontinue the lesson.

(3) Of the established time allowed to complete the lesson.

c. Coordinate any action with the TSM.

(1) Number of students allowed to complete the current lesson.

(2) Time the students will complete the lesson prior to update.

(3) Coordinate the date/time for update.

(4) Assign , udents to the appropriate lessons after notification of good update from the TSM.

A2.3.2 Scheduling CDTS Training.

a. Design a Training Schedule. Assist the TSM in designing a tentative training schedule for the particular needs of your area.

(1) Assess your requirements.

(a) The number of personnel requiring training.

(b) The number of remotes available for training.

(c) Type of training required by the personnel.

(d) Time available for CDTS training.

(e) Obtain the required material (handouts, manuals, etc.) from the TSM.

b. Monitor student enrollments, failures, and deletions.

(1) Coordinate with the TSM.

(2) Prepare student progress record (figure A2-4).

(3) Issue required training materials to the student. (See figure A2-6 for a list of required and optional training materials.)

c. Brief the student on procedures for using CDTS.

(1) Log-on procedures.

(2) On-line time limits.

(3) Off-line CDTS work projects.

(4) Technical problems.

(5) Results of failures of CDTS lessons.

(6) Purpose of duplicate course completion certificates.

d. Monitor CDTS training sessions.

(1) Resolves technical problems on-line.

(2) Evaluates student progress (pass, failure, lesson completion).

(3) Evaluates off-line CDTS work projects. (See figure A2-5 for a list of these projects.)

(4) Assists trainees having difficulties.

(5) Advises and requests course completion certificate from the TSM prior to completion of the courses by the trainee.

(6) Certifies on the student progress record or elsewhere when each lesson requirement is satisfied.

e. Update Student Records.

(1) Certifies on the student progress record the courses that are completed satisfactorily.

(2) Turns the student progress record over to the TSM.

(3) Receives from the TSM and awards the course completion certificate (diploma) to the trainee.

f. Print "hard" copies of edited CDTS lessons.

(1) Provides students with printed copies of individual CDTS lesson text material for study purposes.

(2) The procedures for printing individual CDTS lessons can be found in attachment 7 of this manual.

A2.3.3 Privacy Act Statement. Personal information from the individual is solicited by the CDTS system. As required by the Privacy Act of 1974, we advise:

a. Authority: 10 USC 8012 and Executive Order 9397, 22 November 1943.

b. Principle Purpose: To maintain automated records of student activity on CDTS.

c. Routine Uses: To inform the training system manager (TSM) of student progress in CDTS. The students may verify their lesson completions either by going to the TSM or by having the TSM write the CDTS Section, Keesler AFB, MS.

d. Disclosure is voluntary: Students may take CDTS lessons by entering any alphanumeric code of up to nine characters instead of SSAN. However, students will not receive credit for this training unless they enter either their social security account number or a unique control number acceptable to the local TSM.

KEY
AFDSDC - Air Force Data Systems Design Center
CDT - Computer Directed Training
C&S - Courseware and Software
DPI - Data Processing Installation
PDO - Publications Distribution Office
TM - Training Monitor
TSM - Training Systems Manager
TT - Trace Tape
VCSRSJ - Student Record Summary Program

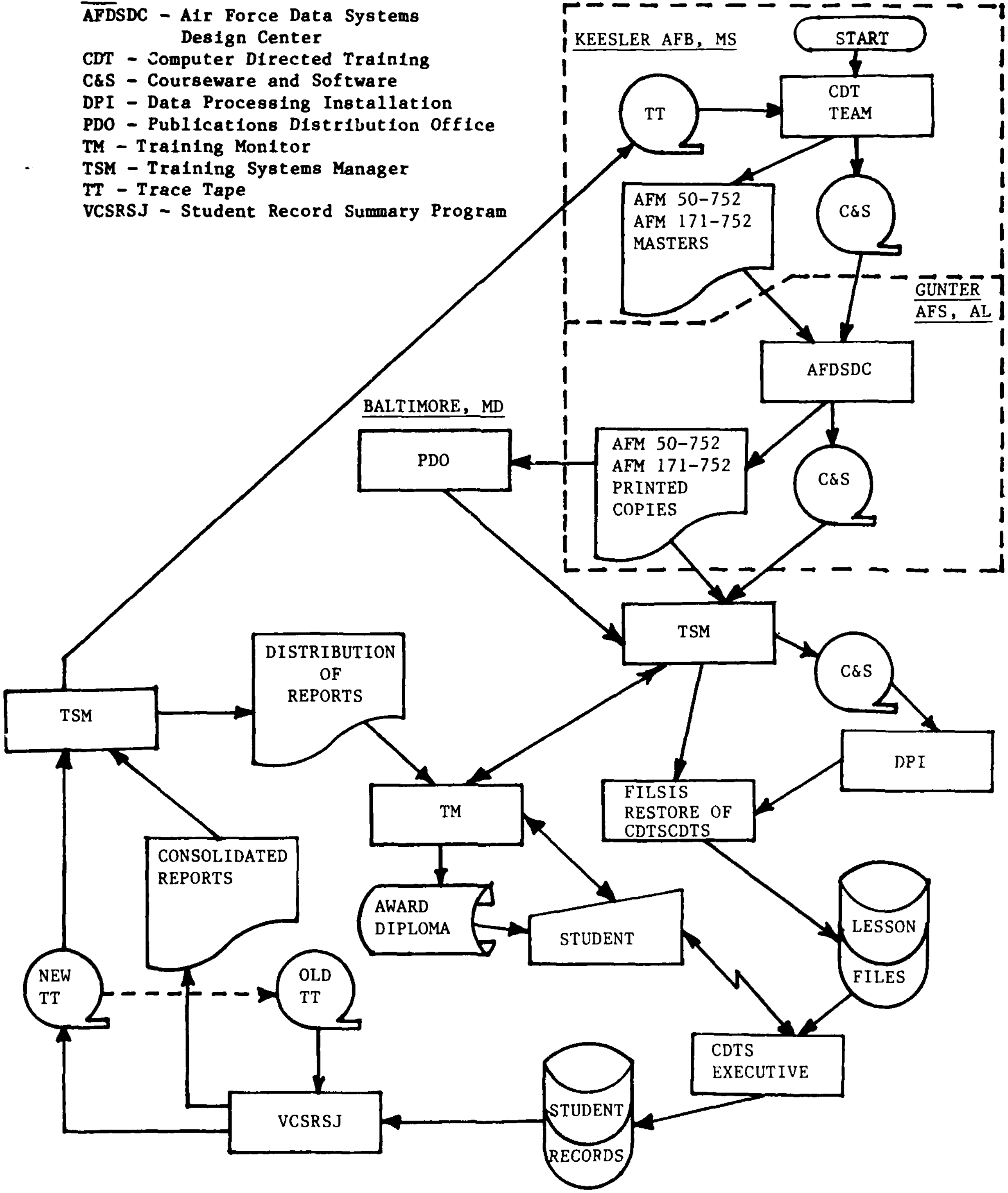


Figure A2-1. Training Systems Manager and Training Monitors Responsibility Flowchart

Course Title	Course Number	PDS Code	Total Hours
COBOL Programming (H6000-CDT)	ECDTS5131B 001	SZ8	40
Computer Directed Training System (CDTS) Coursewriter (H6000-CDT)	ECDTS5135B 003	VZJ	13
FORTRAN Programming (H6000-CDT)	ECDTS5131B 000	SZ9	40
General Macro Assembly Programming (GMAP) (H6000-CDT)	ECDTS5135A 001	VWU	70
Graphics Information Presentation System (GIPSY)	ECDTS5116 002	11H	28
Integrated Data Store (IDS) (H6000-CDT)	ECDTS5135B 000	T7J	30
Introduction to Data Processing (H6000-CDT)	ECDTS5131B 002	VWT	11.5
Introduction to H6000 Series Computer Programming (H6000-CDT)	ECDTS5135B 001	T7K	34
Indexed Sequential Processor (ISP) (H6000-CDT)	ECDTS5135C 002	YCH	31.5
Joint Operational Planning System (JOPS) III User (H6000-CDT)	ECDTS5116 001	T4U	0
H6000 Computer Operator (H6000-CDT)	ECDTS51130 003	S3C	62.5
Management Data Query System (MDQS) Basic User (H6000-CDT)	ECDTS5135C 000	X9A	42
PDP-11 Computer Operator (H6000-CDT)	ECDTS5135B 002	VZK	24
Selective Inquiry System (SIS) (H6000-CDT)	ECDTS5135C 003	ZQQ	9
K700 Test and Diagnostics (H6000-CDT)	ECDTS51130 001	YSK	5
H6000 Total On-Line Test System (TOLTS) (H6000-CDT)	ECDTS51130 002	VZH	8
Terminal User (H6000-CDT)	ECDTS5116 000	S7M	22
WWMCCS Intercomputer Network (WIN) User	ECDTS5116 008	11J	17
BASIC Language Programming (H6000-CDT)	ECDTS5131B 006	4JS	15

Figure A2-2. Course Control Information List

REPLY: (local)

TO: (Individual concerned or Servicing Personnel Office)

SUBJECT: Completion of CDT Course

Grade/Name ____

SSAN ____

DUTY SECTION ____

has successfully completed Computer Directed Training Course # __________

titled ______________________________ #hrs____ PDS______

at ______________(location)______________

Request the servicing Personnel Office enter a record of training in the member's personnel records.

____
TRAINING SYSTEMS MANAGER

Figure A2-3. Certificate of Completion

Student's Name ____

SSAN ______________________ Course Title ______________________

Lesson No. ______ **Start Date** ______ **Completion Date** ______ **Comments** ______ **TM's Initials**

__________	__________	______________	________	____________
__________	__________	______________	________	____________
__________	__________	______________	________	____________
__________	__________	______________	________	____________

" " " " " " " " " " " " " " " ETC. " " " " " " " " " " " " " " " " "

__________	__________	______________	________	____________

CERTIFICATE OF TRAINING (DIPLOMA)

Rec'd by ____(Student's Signature)___ Date ________________________

Figure A2-4. Student Progress Record

Course Title	Lessons Which Have Projects
COBOL Programming (CBL) (H6000-CDT)	CBL010 CBL060 CBL070 CBL120 CBL140
Computer Directed Training System (CDT) Coursewriter (H6000-CDT)	CDT050 CDT060
FORTRAN Programming (FTN) (H6000-CDT)	FTN010 FTN020 FTN030 FTN040 FTN041 FTN042 FTN050 FTN060 FTN080 FTN090
General Macro Assembly Programming (GMAP) (H6000-CDT)	GMP110 GMP170 GMP210
Integrated Data Store (IDS) (H6000-CDT)	IDS040 IDS043 IDS110
Introduction to H6000 Series Computer (INT) (H6000-CDT)	INT040 INT050 INT060
Indexed Sequential Processor (ISP) (H6000-CDT)	ISP020 ISP030 ISP040 ISP050
Selective Inquiry System (SIS) (H6000-CDT)	SIS040 SIS050
Terminal User (TUC) (H6000-CDT)	TUC050 TUC060
Management Data Query System (MDQ) Basic User (H6000-CDT)	MDQ020 MDQ040 MDQ050 MDQ060 MDQ070 MDQ080
Basic Programming Language (BSC) (H6000-CDT)	BSC020 BSC060

Figure A2-5. Course Projects List

Course Title	Title
COBOL Programming (H6000-CDT)	Standard COBOL-68 Reference Manual (DE17)
	Standard COBOL-68 Users Guide (DD18)
Computer Directed Training System (CDTS) Coursewriter (H6000-CDT)	None
FORTRAN Programming (H6000-CDT)	FORTRAN Manual (DD02)
General Macro Assembly Programming (GMAP) (H6000-CDT)	Control Cards Reference Manual (DD31)
	Macro Assembler Program (GMAP) (DD08)
Graphics Information Presentation System (GIPSY) (H6000-CDT)	Graphic Information Presentation System (GIPSY) Users Manual (CSM UM 259-79)
Integrated Data Store (IDS) (H6000-CDT)	IDS Programmers Guide (DC52)
	IDS Users Guide (DC53)
Introduction to Data Processing	None
Introduction to H6000 Series Computer (H6000-CDT)	GCOS Bulk Media Conversion (BMC) Manual (DD11)
	GCOS Utility Manual (DD12)
	Control Cards Reference Manual (DD31)
Indexed Sequential Processor (ISP) Manual (H6C00-CDT)	Indexed Sequential Processor (ISP) (DD38)
Joint Operational Planning System (JOPS) III USER (H6000-CDT)	JOPS presently being rewritten.
H6000 Computer Operator (H6000-CDT) GCOS	GCOS GRTS/355/6600 Startup Procedures Manual (DD05)
	System Startup Manual (DD33)
	Equipment Operators Manual (DA33)
	GCOS System Console Message Manual (DD13)

Figure A2-6. Course Materials List

Course Title	Title
Management Data Query System (MDQ) Basic User (H6000-CDT)	Management Data Query System/IV Users Guide (DD92)
PDP-11 Computer Operator (H6000-CDT) (These publications may be purchased from: Digital Equipment Corp 146 Main Street Maynard, Mass 01754)	PDP-11 Processor Handbook RSX-11D Users Guide RSX-11D System Generation Reference Manual RSX-11D Utility Programs Procedures Manual
Selective Inquiry System (SIS) (H6000-CDT)	AFM 171-714 Selective Inquiry System (SIS) Volume I Computer Operations Manual Volume II Users Manual Control Cards Reference Manual (DD31)
H700 Test and Diagnostics (H6000-CDT)	None
H6000 Total On-Line Test System (TOLTS) (H6000-CDT)	TOLTS Total Online Test System Reference Manual (DD39) TOLTS Total Online Test Pages Manual (DD49)
Terminal User (H6000-CDT)	Time-Sharing Text Editor (DD18) Time-Sharing System General Information Manual (DD22)
WWMCCS Intercomputer Network (WIN) User (H6000-CDT) (These publications are issued by: Director, CCTC (C400) 11440 Isaac Newton Square North Reston, VA 22090	WIN Terminal User's Pocket Guide WIN User's Manual, Vol I

NOTE: Most CDT courses use the exhibits contained in this manual. The publications listed here are used for supplemental instruction and, depending on the course, may or may not be required.

Figure A2-6. Course Materials List (Cont)

FEATURES FOR RESTORING CDTSCDTS

A3.1 PROGRAM APPLICATION.

a. The program in figure A3-1 is used to restore CDTSCDTS from the Computer Assisted Instruction (CAI) release tapes sent by the Air Force Data Systems Design Center (AFDSDC). The Standard Military Release Consists of two magnetic tapes: one tape is a FILSYS save of the System Master Catalog, CDTSCDTS, which is a copy of all the Computer Directed Training System (CDTS) lessons in executable (object) form. The second tape is a FILSYS save of the catalog CDTSCDTS/HCOPY. This is a copy of all the CDTS lessons in edited form which can be printed off for student use.

b. Prior to executing the FILSYS program to restore CDTSCDTS/HCOPY, a catalog named CDTSCDTS/HCOPY must first be created which will point to the appropriate removable disk pack (see figure A3-2). To restore the release tape containing the edited lessons to a removable disk pack refer to the program in figure A3-3. In order to restore the tape to a permanent disk pack instead, the Reset/Rfiles/ or Reset/Device/ FILSYS option must be used.

c. The lessons on CDTSCDTS/HCOPY are edited so that the CDTS lessons appear without CDTS logic or test questions and therefore reads like a book. For procedures for printing these edited lessons refer to Attachment 7 of this manual.

A3.2 OUTPUT REPORTS. After executing the programs to restore the CDTS Release tapes, the output reports should be examined to insure all files on the release tapes were able to be restored. If unable to properly restore a CDTS Release, contact the Air Force Data System Design Center (AFDSDC). For additional information on FILSYS directives refer to the Honeywell manual <u>File Management Supervisor</u>.

```
1      8      16

$      FILSYS
 USERID CDTSCDTS$(site unique password)
 RESTORE CDTSCDTS
$      TAPE9  PR,,,(tape #),,,,DEN8
$      ENDJOB
```

Figure A3-1. Program to Restore CDTSCDTS

```
1      8       16

$      FILSYS
 USERID CDTSCDTS$(site unique password)
 CC CDTSCDTS/HCOPY,DEVICE/yyyPK,Dxxxx/
$      ENDJOB
```

Where the yyy represent the device type of the disk unit, i.e., 450, 181, etc.

Where the xxxx represent the removable pack name, i.e., D5008.

Figure A3-2. Program to Create Pointer

```
1      8       16

$      FILSYS
 USERID CDTSCDTS$(site unique password)
 RESTORE CDTSCDTS/HCOPY
$      TAPE9   PR,,,(tape #),,,,DEN8
$      ENDJOB
```

Figure A3-3. Program to Restore CDTSCDTS/HCOPY

CDTS COMMANDS, EXECUTIVE MESSAGES AND LESSON EXECUTION

A4.1 INTRODUCTION. The following command formats may be used by the student to direct the operation of the CDTS Executive. All commands may be entered in response to ENTER COMMAND. All commands with the exception of ?LIST may be entered in lieu of a response to a question from a lesson in execution. The ?CLOSE and ?SYST commands may be entered in response to a request for a CAT/FILE STRING.

A4.2 FORMAT. In the formats below, brackets, [and], denote optional material. Braces, {and}, indicate that a choice of one of the items in the braces must be made.

A4.3 COMMANDS.

a. The ?GET command causes the lesson file to be searched for the indicated lesson. When the lesson is located, student status information in the CDTS Executive is placed in a "ready" state for that lesson. The NEW option indicates that the student does not have the lesson recorded as "in progress" in his student file. The OLD (or default) option indicates that there is an "in progress" record for the lesson on the student file. When the lesson is "ready," a message to that effect is printed as a reminder before each ENTER COMMAND.

```
                 ⎧ NEW   ⎫
?GET lesson-name ⎨       ⎬
                 ⎩ [OLD] ⎭
```

b. The ?GOTO command directs the CDTS Executive to place a "ready" lesson into execution. The asterisk (*) option indicates that the lesson is either to be started at the beginning--if the NEW option was used with ?GET--or restarted at the question at which a previous training session ended--if the OLD option was used with ?GET. (See the discussion of ?DONE below.) If ?GOTO * is entered as a response to a lesson question, the question will be repeated. The frame-label and frame-number options permit entry to the lesson at arbitrary points; the student should not use these options unless directed to do so by the courseware or the TSM.

```
      ⎧ frame-label  ⎫
?GOTO ⎨ frame-number ⎬
      ⎩ *            ⎭
```

c. The ?DONE command is meaningful only in response to a lesson question. Execution of the lesson is stopped and the Executive returns to the "lesson ready" state. When the lesson is restarted by a ?GOTO * the question will be repeated. The commands ?GET, ?GOTO frame, ?CLOSE, and ?NEW, entered in response to a lesson question have the same effect as if a ?DONE were entered first.

```
?DONE
```

d. The ?CLOSE command directs the CDTS Executive to terminate and returns the user to the time-sharing system-select (*) level.

?CLOSE

e. The L option of the ?LIST command causes a list of all lessons available on the lesson file to be printed. The S option causes a list of all "in progress" lessons on the student file to be printed. If neither option is given, both lists are printed. This command may not be entered in response to a lesson question. If this error occurs, the input is ignored, and the lesson question remains pending. The ?LIST command cancels any preceding ?GET command.

?LIST [$\begin{Bmatrix} \text{L} \\ \text{S} \end{Bmatrix}$]

f. The ?NEW command allows the user to identify a new lesson file to the Executive. A catalog/file-string will be requested as the next input line.

?NEW

g. The ?SYST command permits the user to enter another time-sharing system without terminating CDTS. A system name will be requested as the next line of input. Valid system names may include ABACUS, BASIC, FORTRAN, EDITOR, CARDIN, etc. A command in one of these systems which would ordinarily return to the time-sharing * level will return to CDTS. Additional "push-down" calls (for example EDITOR from CARDIN) may go only one level down. (RUNOFF could not be called from EDITOR out of CARDIN). There are very few cases, however, in which violation of this restriction is even possible. If the ?SYST command is entered in response to a lesson question, the question will still be pending on ~eturn to CDTS. A ?GOTO * may be used to repeat the question, if needed. Student errors in the entered system may terminate both that system and CDTS. Recovery can be made as if a ?CLOSE had been given to CDTS. Students who are not familiar with H-6000 time-sharing should avoid the ?SYST command unless directed to use it.

?SYST

h. The ?PASS command permits the student to take an unofficial look at the list of lessons successfully completed. (Note: The only lessons listed will be those passed while using the current student file.) This command may be used any time the ENTER COMMAND message is present. (If the student has passed more than 300 lessons, the oldest passes will be deleted as new passes are added.)

?PASS

A4.4 CDTS EXECUTIVE MESSAGES. For a list of messages, conditions, and corrective actions see figure A4-1.

A4.5 EXECUTING A LESSON. For a step by step procedure on accessing and executing CDTS lessons, see figure A4-2.

MESSAGE	CONDITION	ACTION
CDTS ENDED	CDTS has terminated.	Information only
CDTS RETURNS	A return has been made from the pushdown execution of another time-sharing system.	Information only
CDTS STARTED	CDTS is in execution.	Information only
ENTER COMMAND	CDTS is waiting for command.	Enter a command
EXECUTIVE ERROR (OP) or EXECUTIVE ERROR (ADDR)	An error in lesson file content has been encountered. The lesson file is invalid; or the file given is not, in fact, a lesson file.	Check to be sure you have a correct lesson file. If not the wrong file, save all available output. Notify the TSM.
FILE ACCESS ERROR STATUS 77777777777	An error has occurred while attempting to access a file.	CDTS will re-issue the CAT/FILE STRING request. If the error can be determined by inspection, re-enter the corrected data. The specific error may be determined by entering the HELP subsystem with message 50, Group 4, and a status equal to the third and fourth digit in the CDTS message.
FRAME 999.9 XXXXXX ITEM ILLEGAL VALUE	The lesson contains an internal error.	The lesson will continue, but evaluation of student actions may be incorrect. Record the information in the message. Save all available output. Notify the TSM.
FRAME LABEL/NUMBER NOT FOUND	The student has requested entry to the lesson at a non-existent frame.	Use an existent label number.
ILLEGAL FRAME LABEL/NUMBER	An illegal frame label/number has been used in a ?GOTO command.	Use a legal label/number.

Figure A4-1. CDTS Executive Messages

MESSAGE	CONDITION	ACTION
ILLEGAL ID	The student SSAN input does not match the SSAN on the student file referenced.	CDTS will be terminated.
ILLEGAL IN EXECUTION	The ?LIST command may not be entered in response to a lesson question.	Use another command or respond to the original question.
ILLEGAL NAME	The student has entered an illegal lesson name for a ?GET command.	Use a legal lesson name.
I/O ERROR STATUS 777777777777	An error has occurred while reading or writing a disk file.	Record the 12-digit error status. Save all available output. Notify the TSM. The error status is the first word of a MME GEINOS status return as documented in the GCOS Manual.
LESSON ALREADY STARTED-RESTART?	The student has requested entry to a lesson already started.	Answer YES or NO. YES will cause the in-progress status of the lesson to be erased. NO causes the lesson to be treated as if a ?GET...OLD had been entered.
LESSON ENDED	The current lesson is concluded.	Information only
LESSON FILE - CAT/FILE STRING	CDTS requests identification for a lesson file.	Enter a file descriptor (cat/file string). See the note.
LESSON NOT FOUND	The lesson file does not contain the requested lesson.	If the lesson name has been misspelled, re-enter. If the wrong lesson file was given to CDTS, use the ?NEW command. Otherwise notify the TSM.
NO CURRENT LESSON	No lesson is ready for a ?GOTO	Use the ?GET command.

Figure A4-1. CDTS Executive Messages (Cont)

MESSAGE	CONDITION	ACTION
NO DATA FOR RECORDS	No activity occurred during the current session to warrant records generation.	Information only
NO ROOM IN STUDENT FILE	All space in the student file is occupied by records for incomplete lessons.	Complete one of the lessons.
OLD LESSON NOT IN STUDENT FILE	The student has requested resumption of a lesson which he did not start.	Use ?GET lesson-name NEW.
OLD LESSON UPDATE SINCE LAST USE	The version of the lesson on the lesson file is not compatible with the information retained on the student file.	The student must re-take the lesson. The student must start the lesson with a ?GET xxxxxx NEW.
XXXXXX READY	Lesson XXXXXX is available for execution.	Information only
RECORDS SPAWNED 9999T	Job 9999T was spawned to record the student's activity during the current session.	Record the date, time, and SNUMB (9999T) in the event that records become lost due to system malfunction.
STUDENT FILE - CAT/FILE STRING	CDTS requests identification for the student's student file.	Enter a file descriptor (cat/file string). See the note.
SYSTEM NAME?	A ?SYST has been received.	Enter a time-sharing system name.
UNKNOWN COMMAND	The student has entered an input beginning with "?" which is not a legal command.	Enter a legal command or correct the error input.

NOTE: CATALOG/FILE STRINGS. Catalog/File strings extending over more than one line of input may be used, provided that the last nonblank character on a line to be continued is "/" or "$".

Figure A4-1. CDTS Executive Messages (Cont)

STEP 1. After completing your TS1 log-on procedure, you will be prompted with:

```
    *
```

Reply with CDTS. This will place the CDTS time-sharing executive program into execution.

STEP 2. You will then see

```
     CDTS STARTED-

     STUDENT FILE-
     CAT/FILE STRING
```

Enter the catalog/file-string for your student file.

STEP 3. Then you will see

```
     IAW THE PRIVACY ACT OF 1974, YOU ARE ADVISED THAT
     DISCLOSURE OF THIS INFORMATION IS VOLUNTARY
     SEE AFM 50-752, ATTACHMENT 8, FOR MORE INFORMATION
     STUDENT SSAN (9)
```

Enter your social security number. If you have used your student file before, go to step 4; otherwise, additional identification will be requested. In all cases, the number in parentheses is the maximum number of characters allowed for the input.

```
     GRADE OR TITLE (6)

     LAST NAME (30)

     FIRST NAME...MI (30)

     ORGANIZATION (24)
```

For this last item, enter the organization to which your activity reports are to be sent. Contact your TSM for the format of this entry.

STEP 4. The next output is

```
     LESSON FILE-

     CAT/FILE STRING
```

Enter the catalog/file-string of the lesson file containing the lesson you wish to take. The catalog/file-string will be supplied by your TM.

STEP 5. Then you will see

```
    ENTER COMAMND
```

Figure A4-2. Executing a Lesson

Reply with

```
    ?GET XXXXXX NEW
```

where XXXXXX denotes the name of the lesson you wish to take. If you have already started this lesson in a previous session, enter

```
    ?GET XXXXXX OLD
```

The OLD is a default and may be omitted.

STEP 6. This will be followed by

```
     XXXXXX READY

     ENTER COMMAND

     *
```

Reply with

```
     ?GOTO *
```

This will cause the lesson to be started at the beginning, or the place where you previously stopped it.

STEP 7. The lesson will execute, presenting instructional material and asking questions. The lesson can then be stopped in either of two ways:

i. At the end of the lesson it is stopped automatically when you see

```
LESSON ENDED
ENTER COMMAND
```

If you wish to take another lesson, go to Step 8; otherwise, go to Step 9.

j. In response to any lesson question, you may enter the CDTS command

```
?DONE
```

which will stop the lesson until you wish to resume it. If you wish to take another lesson, go to Step 8; otherwise, go to Step 9.

STEP 8. If you wish to start another lesson that you know is on this lesson file, go to Step 5; otherwise, enter the CDTS command

```
    ?LIST
```

to find what lessons are incomplete on your student file and what lessons are available on the lesson file, then go to Step 5. If the desired lesson is on another lesson file, enter the CDTS command

```
    ?NEW
```

which returns you to Step 4.

Figure A4-2. Executing a Lesson (Cont)

STEP 9. To terminate CDTS, enter the CDTS command

```
    ?CLOSE
```

You will then see

```
     RECORDS SPAWNED  XXXXT

     CDTS ENDED

     *
```

You may then enter some other time-sharing subsystem or terminate time-sharing with the command BYE.

Figure A4-2. Executing a Lesson (Cont)

CDTS STUDENT RECORDS SUMMARY PROGRAM AND BUILDING OF TRACE TAPES.

A5.1 PROGRAM APPLICATION AND BACKGROUND.

a. The CDTS Executive picks up all CDTS student responses to questions within lessons, and records the number of students and hours of terminal time per lesson. All this information is placed on the student record accumulator file (FVCXASJ3U). The CDTS Student Records Summary program reads the accumulator file, builds a Student Activity Summary report and outputs a trace tape.

b. The Student Activity Summary is a report provided for the local Training System Manager (TSM) to use in evaluating local CDTS utilization.

c. A flowchart of the CDTS Student Records Summary program is shown in figure A5-1. This figure shows an old trace tape to be merged with new student record data from the accumulator file (FVCXASJ3U) to create a new trace tape.

d. The CDTS Student Records Summary program (see figure A5-2) allows the trace tape output to be "grandfathered" by being an input on a subsequent run of this program. This "old master/new master" cycle allows trace tape data to be accumulated and merged into a single tape for quarterly transmission to the CDTS Development Center at Keesler AFB, MS.

A5.2 ACCUMULATOR FILE. Each time a student ends a CDTS session, the executive spawns a batch job containing records (in the form of card images) of the progress through the lessons taken. The spawned job accumulates records directly to a permanent file which must be reinitialized to save file space and prevent duplicate inputs to the Summary program. A card containing "RESET" in columns 1-5 may be inserted in the summary program after the $LIMITS cards. If the card is present, a single blank card will be written to the accumulator file satisfying the reinitialization requirement.

A5.3 TRACE FILE. Files FO and FN (see figure A5-2) are respectively the "old master" and "new master" for update accumulation of a trace of student responses to lessons. File FO is an <u>optional</u> file which must be omitted from the program when no "old master" exists. An FN file transmitted to Keesler AFB for quarterly reporting must <u>not</u> be re-input as FO for the next processing cycle.

A5.4 SAMPLE OUTPUT. Each time the Student Records Summary program is run, two distinct types of output are produced. One output is a summary report and the other is a trace tape.

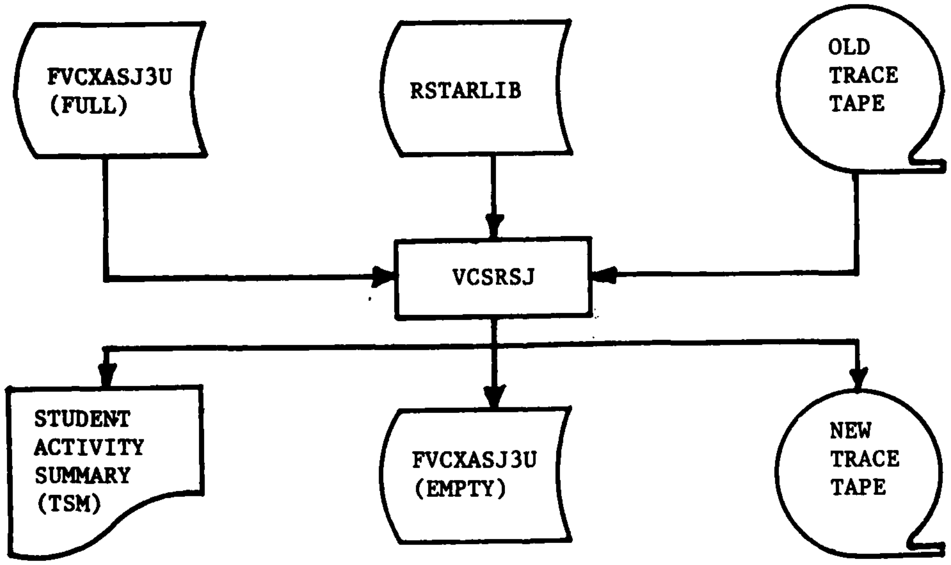
a. Lesson Trace Tape. This information is a highly detailed record of student responses to lesson material. Previous trace data and new trace data are taken from file codes FO and F3 respectively, merged, and written to file code FN. This information is intended for use by the CDTS Development Center Coursewriters at Keesler AFB, MS. The various reports obtained from this trace tape are briefly outlined in Attachment 6 of this manual.

b. Student Activity Summary Report. This information is a listing of student starts, passes, fails, and ends of lessons. It is not cumulative; it reports only those events which have occurred since the last time the summary was executed. This report is intended to serve the Training System Manager (TSM) as source information for updating of off-line records of student enrollment and progress through CAI lessons. Figure A5-3, Explanation of Stduent Activity Summary, shows the format of the printed report and provides guidelines for interpretation.

A5.5 RELEASE OF CURRENT LESSON TRACE FILE. Once each quarter (Feb, May, Aug, Nov) and each time a new release is loaded, each TSM will ensure that the current lesson trace file is copied onto magnetic tape. This tape must be labeled with tape density and external reel number. Mail the tape to:

3300 TCHTW/TTGXZ
Attn: WWMCCS CDT
Keesler AFB, MS 39534

The tape will be returned to the sender as soon as it has been processed. Most of the information contained on the trace tape is kept on a master file by the Development Center at Keesler. Summaries and portions of this information are available to the TSM or student by request to the 3300 TCHTW/TTGXZ.



Note: FVCXASJ3U is the Student Record Accumulator File and RSTARLIB is where VCSRSJ resides.

Figure A5-1.Flowchart of Student Records Summary Program

```
1      8      16

$      LIBRARY LB
$      USE    VCSRSJ
$      ENTRY  VCSRSJ
$      EXECUTE
$      PRMFL  LB,R,S,CDTSCDTS/RSTARLIB
$      LIMITS 99,24K,,5000
$      DATA   I*
RESET
$      PRMFL  F3,R/W,S,(Cat/File string of accumulator file)
$      TAPE   FO,,,(Reel # for old trace tape)
$      TAPE   FN,,,(Reel # for new trace tape)
$      FILE   WØ,,30       (see note)
$      FILE   W1,,10       (see note)
$      FILE   WR,,30R      (see note)
$      SYSOUT PT
```

File codes associated with the program are as follows.

F3 -- A deck, or permanent disk file containing accumulated student record data. See paragraph A5.2 for additional information.

FO -- The old student response trace file. See paragraph A5.3 for additional information.

FN -- The updated student response trace file.

WØ -- A work file (see note).

W1 -- A work file (see note).

WR -- The assigned file code for a sort collation file.

PT -- The file on which the Student Activity Summary report will be written.

NOTE: File sizes shown are "typical," but will vary from site to site. In general, file WØ and WR must be at least the size of the F3 file. A size of 10 LINKS (as shown) for file W1 should be sufficient for all applications.

Figure A5-2. Student Activity Summary Program

```
                              SAMPLE PAGE (Note 1)
****         STUDENT STATUS REPORTS FOR (organization, Note 2)          ****

GRADE  LAST NAME   FIRST NAME  MI      SSAN      LESSON STATUS DATE
 (student supplied information,                  ------ ------ ------
            Note 3)                  (Note 4)    TUC010 START  820326
                                     (Note 5)    TUC010 PASS   820327
                                                 TUC010 END    820327
                                     (Note 6)    TUC020 START  820328  (2 TIMES)
                                                 TUC020 PASS   820328
                                                 TUC020 FAIL   820328
                                                 TUC020 END    820328  (2 TIMES)
```

Note 1. Dimensions and spacing are approximate, to conform to the limitations of manual format.

Note 2. A new page will be started for each distinct organization name. This information is taken from student input, and the TSM or TM should take steps to insure uniformity of responses.

Note 3. This information is taken from student input. Once established it remains permanently recorded on the student file.

Note 4. These three lines show a "typical" report of student progress through a lesson. Note that the dates, in YYMMDD format indicate that two days were required for completion of the lesson. The "START" and "END" status indicators are self-explanatory.

Note 5. A "PASS" status indicates successful completion of the lesson, subject to TM verification of successful completion of assigned projects or exercises. If such an assignment is given it will be noted in the course description of this manual. A "FAIL" status indicates that the student did not successfully complete the lesson.

Note 6. These four lines indicate that the student took the same lesson twice in the same day. As a matter of instructional effectiveness, a student who fails a lesson should be discouraged from retaking it the same day. Status records are sorted in START, PASS, FAIL, END order within days, and the order of listing on this report does not reflect a time-sequence. The appearance of a PASS status should be taken as evidence of successful completion, subject to Note 5 above.

Note 7. Due to system malfunctions, etc., some student status records may be lost. The student is advised of the SNUMB of the job that was spawned to accumulate these records, and should be encouraged to make a note of it. In the event that a job shows abnormal termination on the H6000 system log, some data, including PASS records, may have been lost. Discrepancies of this type will be resolved at the discretion of the TM.

Figure A5-3. Explanation of Student Activity Summary

DEVELOPMENT CENTER TRACE TAPE PROCESSING

A6.1 PROCESSING OF TRACE TAPES. The following is a breakdown of the procedure applied to these tapes by the CDTS Development Center.

a. When a trace tape is received from the field, it is processed by first building a base identifier file so that the data from the tape can be credited to the proper categories in the data bases. Each site that sends trace tapes to Keesler AFB has an entry in the CAI data base.

b. Each trace tape contains two general types of information. One category of data is summary totals showing number of students and hours of training, broken down by course. The other category of data is lesson validation data which shows questions miss rates for each lesson. The development center extracts each record type and loads the data along with the base identifier file to the separate data bases.

c. In addition to the two general types of data, each trace tape contains student critique responses, including unusual responses and an invalid course request list. When the development center processes a trace tape to extract summary totals statistics, the following reports are built:

(1) Utilization Report (2 copies)

(2) Unusual Critique Response Report (if any)

(3) Invalid Course Request Report (if any)

One copy of the Utilization Report along with an Invalid Course Request Report (if there is one) is returned to the sending site along with their trace tape. The other copy of the Utilization Report and the unusual Critique response Report are maintained at the CDTS Development Center at Keesler AFB. The Utilization Report provides the local TSM with a summary by SSAN of all lessons processed during this reporting period. A breakdown by course showing total student hours spent in each lesson of each course, total number of students and total number of hours involved in CAI training during this reporting period is included in this report. The Invalid Course Request Report is provided to the local TSM for information, it shows courses requested by students in response to the "Lesson File Cat/File String?" question in CDTS.

d. CDTS Development Center utilizes the unusual critique responses to improve course material. If CDTS Development Center has questions on any response, contact may be made through the local TSM to the student for further clarification.

e. CDTS Development Center builds a master data base file from all sites for summary totals of CDTS utilization. Each year a report is made to the WWMCCS Training Panel showing totals by service (Air Force, Army, and Navy) of CDTS utilization. In addition to grand totals of numbers of students and total hours of training for all courses per service, each lesson is also broken down to show utilization. From this data, WWMCCS Training Panel decides course priorities and tasking.

f. The lesson validation category of information on each trace tape is used to produce a report used by coursewriters at the Development Center to validate questions and instructions in each lesson.

PRINTING "HARD" COPIES OF EDITED CDTS LESSONS

A7.1 The main purpose of printing "hard" copies of edited lessons is to provide students with copies of text material for study purposes.

A7.2 The procedure for printing individual CDTS lessons is as follows:

a. Using TSl on card input, set up a Bulk Media Conversion (BMC) activity as shown in figure A7-1.

b. After setting up the BMC activity, enter a RUN command. For CARD FORMAT, DISPOSITION?, enter STRIP. For TAB CHARACTERS AND SETTINGS?, enter NORM.

c. Continue running this program changing the lesson catalog/file name (refer to TUC/TUC010 in figure A7-1) until you have the required [illegible]sons.

d. Refer to pages 2-6 through 2-13 of this manual for a comp[illegible]e list of courses and lessons which are available for "hard" copy printing.

A7.3 Before printing any "hard" copies of CDTS lessons be sure tl[illegible]cedures concerning restoring the catalog CDTSCDTS/HCOPY in Attachment 3 [illegible]en followed.

```
1      8       16

$      IDENT   (Local Requirement)
$      USERID  (Not required for TSl)
$      CONVER
$      LIMITS  ,,,3000
$      PRMFL   IN,R,S,CDTSCDTS/HCOPY/TUC/TUC010 (or required lesson
$      SYSOUT  OT                                catalog/lesson name)
$      ENDJOB
```

NOTE: A $MULTI card can be used to print multiple copies.

Figure A7-1. Bulk Media Conversion (BMC) Activity

CDTS LESSON DEVELOPMENT

TABLE OF CONTENTS

A8.1 SYSTEM OPERATION.

A8.1.1 LESSON PREPARATION. A CDTS lesson is a combination of subject matter text and a computer program which directs the presentation of the lesson. Both of these types of information must be "coded" according to the rules of the CDTS language. The coded lesson must then be prepared as "machine readable" input to the computer.

a. Experience has shown that the best method of creating an effective lesson is to code a complete draft in manuscript before beginning computer input. Once the manuscript has been written and reviewed, the lesson may be prepared for input, either on standard 80-column punched cards, or in the form of a time-sharing text file.

b. During initial data entry, lesson refinement, and lesson maintenance, use of a time-sharing text file permits changes and corrections to be made very easily with the text-handling features of H6000 time-sharing. Standard utility programs are available for conversion from card to text file format if data entry was done via punched cards.

A8.1.2 COMPILING AND LOADING. A complete lesson in "source" form (as coded) must be input to the CDTS compiler as the first step in eventual presentation to the student. The CDTS compiler is a computer program which performs two essential functions:

a. First, the compiler examines the source lesson for coding errors and flags them in a printed listing of the source lesson. Since a computer cannot make "reasonable assumptions" about what was intended by erroneous code, all such errors must be found and corrected.

b. Second, and more importantly, the compiler produces an "object" lesson file as output. The object lesson file contains the information content of the source lesson in a form more suitable for use by a computer in lesson presentation. Subject matter information (text) is carried from source to object without change. Control information (the "program" aspect of the lesson) is, however, greatly transformed.

c. The object lesson file output by the compiler must then be input to the CDTS general utility program. As its name implies, the general utility performs several functions, but in this context the most important is the "load" function. (See paragraph A8.6 for more information on the load program.)

d. Lesson loading consists of little more than compacting the data on the object lesson, and writing it to a random-access magnetic disk file. This disk file is a "lesson file" in the final form used by the CDTS executive to present the lesson. A given lesson file may contain several distinct lessons.

A8.1.3 EXECUTION. In data-processing terminology, the CDTS executive is a "simulator"; that is, a program which causes the computer system to simulate the operation of another device. In this case the device being simulated is a "teaching machine" with a limited, but highly specialized repertoire of functions. Each item of control information on a lesson file corresponds to a "teaching machine" function. It is the task of the executive to fetch this information from the lesson file. analyze it, and cause the computer to take appropriate action. In many cases, a control function will have associated subject-matter information, such as a line of text to be displayed.

a. The CDTS executive runs under the control of the H6000 time-sharing system. In this mode of operation, individual students on-line are connected to their own copy of the executive. Multiple copies of the executive can concurrently be read from different locations on a single lesson file. Hence, each student is independent of other students, and each may proceed with their own choice of lessons at their own pace.

b. Although an unlimited number of students can share a lesson file, a unique "student file" is required for each student. At regular intervals the executive writes the student's location in the lesson and other unique status information to the student file. If the student wishes to terminate the time-sharing session before the lesson is complete (or if the computer system fails), the student may return to the lesson at the point corresponding to the last status record for that individual.

c. A student file is also a random-access disk file. Each block of the file can contain a status record for an active (incomplete) lesson. The number of active lessons a student may have is limited only by the allocated size of the student file. This capability could be used, for example, to allow a student to review another lesson before continuing the current lesson. Once a lesson is complete, the block assigned to it is marked available for re-use.

A8.1.4 SIZE LIMITATIONS. A frame may be any practical size. A lesson may contain, at most 9999 frames. The generated object lesson (which generally runs between 30% and 50% of the source lesson size, at 72 characters per card) may not exceed 262144 characters.

A8.2 COMPUTER DIRECTED TRAINING SYSTEM CODING. The following section contains complete coding conventions and performance specifications for the CDTS language. (See also file CDTSCDTS/CDT/SKELETON.)

A8.2.1 REFERENCE FORMAT. CDTS language syntax combines column-aligned, record-oriented conventions and free-form, sentence-like structure. The following reference format has been specifically designed (with adaptations from the COBOL reference format) for documenting CDTS syntax requirements.

a. Language elements shown as upper case letters, digits, and special characters must be coded as shown.

b. Language elements represented by lower-case terms are described or defined by the term. Supplemental notes will be provided.

c. A number above a language element indicates a mandatory card-column alignment for its first character.

d. Square brackets, [and], indicate that the language elements contained there are optional.

e. Braces, {and}, indicate that a choice of one of the elements vertically tabulated therein must be made.

f. The ellipsis, . . ., following an element in square brackets, indicates that the element may be repeated indefinitely.

g. Blanks are required as language element separators in some contexts and forbidden in others. The special symbol B̸ (slashed upper-case B) will be used to indicate that at least one blank is required at that point in a format. The special symbol b̸ (slashed lower-case b) will be used to indicate that blanks are optional at that point in a format. Blanks following a language element may be assumed only if it is the last element on a card, or the next element is explicitly column-aligned. A column-aligned optional element will be blank-filled if the option is not exercised. Actual spaces appearing in a format are for clarity only. They do not imply that blanks may be coded.

A8.2.2 Card Format. All CDTS lessons must be prepared on 80-column BCD cards, or some other medium having the same "image" on input to the computer. In the discussions of CDTS coding conventions that follow, all input records will be referred to as "cards" on the understanding that other media may be used.

a. There are three legal card types in a CDTS source lesson.

(1) The lesson card, distinguished by an "L" in column 1. This card type is discussed in further detail as a separate topic.

(2) Comment cards, distinguished by an "*" in column 1. Uses and restrictions on comment cards are discussed separately.

(3) Group cards, distinguished by a digit (1 to 4) in column 1. There are several types of group cards which are discussed separately. In all cases, card column 1 is reserved for card type information only. As a convenience, however, only the first of several cards with the same first-column digit (a "group") must contain the digit. Remaining cards in the group may have a blank in column 1. In the reference formats shown in this manual, the digit will always be shown for clarity.

b. Card columns 2-72 are available for coding as appropriate to the card type.

c. Card columns 73-80 must contain an ascending sequence field. This requirement may be bypassed if the SEQ option is exercised on the lesson card. The sequence field is not shown in reference formats.

A8.2.3 COMMENT CARD. A comment card permits coursewriter remarks to be inserted in a source lesson.

Format:	1	2-72
	*	Comments

NOTES:

1. Comment cards may not precede the lesson card in the source lesson. Otherwise, there is no restriction on their appearance in the lesson.

2. Comments appear in the listing of a lesson produced by the CDTS compiler, but have no effect on the lesson as it is presented to the student. Their purpose is to provide documentation of the lesson internal to the text of the lesson itself.

3. The following uses are suggested for comments:

a. To record lesson version date and maintenance history.

b. To provide a frame-number cross-reference table for labeled frames.

c. To record all items used, and their significance.

d. To divide the lesson into topical and functional segments.

e. To provide detailed explanations of lesson flow-of-control logic.

A8.2.4 TEXT HANDLING.

a. The special term - text - denotes textual material. Text may consist of any string of characters, and may contain leading and embedded blanks.

b. Each text element which is to be displayed will be presented as a single line, left-justified (with leading blanks retained) on the student's remote terminal.

c. In the absence of form controls, each display line is followed by a single line feed. The form control character \ (backslash) at the end of a text string causes an additional line feed to be generated—one additional line feed for each backslash. A backslash followed by any character other than a space or another backslash is treated as a text character.

d. A text string may consist of form controls only. In that case, the line-feed count is added to the count for the last line displayed. Note that a form controls only string of N backslashes causes N+1 line feeds to be generated.

e. The maximum number of backslashes recognized on a single text string is four.

f. Four backslashes are recognized as a request to "page" a CRT-type terminal. A page cue (caret asterisk or blinking asterisk) is generated. When the student indicates that a page has been finished (by a null transmit), the screen is cleared and a new page is presented.

A8.2.5 LESSON. A lesson is the largest unit of instruction recognized by the CDTS compiler and executive programs.

```
lesson-card
    .
    .
    .
one or more frames
    .
    .
    .
```

NOTES:

1. A lesson card followed by one or more frames comprises a CDTS lesson.

2. A student may take a lesson in several time-sharing sessions. The CDTS executive saves the "location" in the lesson and various indicators of the student's progress from each session to the next. When a student returns to a lesson, presentation resumes at the beginning of the frame the student was in when the lesson was terminated. Indicators are reset as though the frame had never been entered.

3. Unless an explicit transfer of control is encountered, the normal flow of a lesson is from frame to frame in the order coded.

4. No special card or coding is needed to mark the end of a lesson.

5. A lesson is terminated and cleared from the student's student file only when the normal flow of the lesson passes through the last frame.

A8.2.6 QUESTION FRAME. A question frame permits students to give a "constructed" response (in their own words) to a lesson question.

Skeleton (multi-card):

```
     Group 1 card

     (group 1 of the frame)
         .
         .
         .
     one or more group QM2 cards
     (group 2 of the frame)
         .
         .
         .
     one or more group Q3 cards
     (group 3 of the frame)
         .
         .
         .
     one or more group QM4 cards
     (group 4 of the frame)
         .
         .
         .
```

NOTES:

1. Two or more cards of the types shown above, and in the order shown above comprise a question frame.

2. Presentation of the frame corresponds to its division into groups as follows:

Group 1 -- identifies the frame and its type. No operations are visible to the student.

Group 2 -- contains text to be displayed to the student. On completion of the text display, an input cue consisting of a single asterisk, left-justified on a separate line, is generated.

Group 3-- contains a list of "anticipated" student responses. No text is displayed.

Group 4 -- contains sections of "action commands" which will be performed in reply to a given student response. Every question frame in Group 4 must contain a provision for an unanticipated response ("-" tag). Failure to meet this requirement will be treated as an error by the compiler.

3. Some groups are optional in a question frame:

Group 1 -- must be present. No frame may consist of group 1 only.

Group 2 -- is optional. If it is omitted, no text is presented prior to the input cue.

Group 3 -- is optional. If it is omitted, no input is requested from the student.

Group 4 -- is required if a group 3 is present; otherwise it is optional. If no group 3 is coded, group 4 may contain only "unconditional" action commands.

A8.2.7 MULTIPLE-CHOICE FRAME. A multiple-choice frame permits the student to choose from a list of answers to a question.

```
Skeleton: (multi-card):
     group 1 card
     (group 1 of the frame)
          .
          .
          .
     one or more group QM2 cards
     (group 2 of the frame)
          .
          .
          .
     one or more group M3 cards
     (group 3 of the frame)
          .
          .
          .
     one or more group QM4 cards
     (group 4 of the frame)
          .
          .
          .
```

NOTES:

1. Two or more cards of the types shown above, and in the order shown above comprise a multiple-choice frame.

2. Presentation of the frame corresponds to its division into groups as follows:

Group 1 -- identifies the frame and its type. No operations are visible to the student.

Group 2 -- contains text to be displayed to the student.

Group 3 -- contains the text of the alternative answers to be displayed to the student. Answers to multiple choice questions may extend over more than one line. A continuation line must have blanks in card columns 2 and 3. The Group 3 differs from the raw text of Group 2 in that the list of letters which identify the answers must be coded in a specific format. On completion of the text display, an input cue consisting of a single asterisk, left-justified on a separate line, is generated.

Group 4 -- contains sections of "action commands" which will be performed in reply to a particular student choice.

3. Some groups are optional in a multiple-choice frame:

Group 1 -- must be present. No frame may consist of group 1 only.

Group 2 -- is optional. If it is omitted, textual display begins with the contents of group 3.

Group 3 -- is optional. If it is omitted, no input is requested from the student.

Group 4 -- is required if a group 3 is present; otherwise it is optional. If no group 3 is coded, group 4 may contain only "unconditional" action commands.

A8.2.8 DECISION FRAME. A decision frame permits tests on the status of indicators of student progress, and conditional actions thereon.

```
Skeleton (multi-card):
     group 1 card
     (group 1 of the frame)
         .
         .
         .
     one or more group D2 cards
     (group 2 of the frame)
         .
         .
         .
```

NOTES:

1. Two or more cards of the types shown above, and in the order shown above comprise a decision frame.

2. Execution of the frame corresponds to its division into groups as follows:

Group 1 -- identifies the frame and its type. No operations are visible to the student.

Group 2 -- contains condition tests and conditional or unconditional "action commands" indicating operations to be performed.

3. Both groups of a decision frame are required.

4. A decision frame may contain multiple conditional expressions, not to exceed 20 simple conditions. A conditional expression is begun by an IF, separated by an ELSE and terminated with an END or a new frame.

A8.2.9 Lesson Card. A lesson card identifies the lesson to the compiler, and contains some options which apply to the entire lesson.

Format:

```
1 2-7                  10        15
L lesson-name       [{FLAG}]    [SEQ]
                    [{NREC}]
```

NOTES:

1. A lesson card must be the first card of each lesson.

2. A lesson name may be one to six characters in length. The first character must be alphabetic. The remaining characters may be any combination of alphabetics and numerics.

3. The FLAG option causes all frames in the lesson to be flagged.

4. The NREC option instructs the compiler to suppress generation of data required for CDTS student records.

5. The SEQ option is required if columns 73-80 of the source lesson cards do not contain valid alphanumeric sequence numbers. If sequence numbers are present, the SEQ option suppresses them in the source listing.

A8.2.10 GROUP 1 CARD. A group 1 card identifies a frame and its type.

Format:

```
1  2        8               16    17    19         26

1  [FRAME]  frame-number   ( Q )  [1]   [LABEL =]  [frame-label]
                           { M }
                           ( D )
```

NOTES:

1. A group 1 card must be the first card of a frame.

2. The FRAME and LABEL = options have no effect on the compilation of the lesson. They are provided solely to improve the readability of the lesson listing.

3. A frame number may be any value in the range 0.1 to 999.9, in increments of 0.1. Each frame must have a unique number. Frame numbers must be assigned in ascending order through the lesson. Successive frame numbers need not be consecutive. The first digit, regardless of the magnitude of a frame number, must be left-justified to column 8 on a group 1 card. Leading zeros may be used to align the decimal point in column 11.

4. Q, M, and D in column 16 denote a question, multiple-choice, or decision frame, respectively.

5. A 1 in column 17 denotes a flagged frame. Each time that a flagged frame is encountered in presenting a lesson, the following information is recorded in a "flagged frame table" maintained for each student for each lesson:

a. the identity (number) of the frame;

b. whether there is a correct answer to the frame, and if so -- whether the student's response to the frame was correct or incorrect.

c. the group 3 tag letter identifying the student's response.

6. The flagged frame table is maintained in chronological order. It has a maximum capacity of 250 entries. Once the maximum capacity is reached, the oldest entry will be "pushed out" to make room for the newest entry.

7. A frame label may be one to six characters in length. The first character must be alphabetic. The remaining characters may be any combination of alphabetics and numerics. If a frame label is assigned to a frame, it must be unique.

A8.2.11 GROUP QM2 CARD. A group QM2 card contains text to be presented in a question or multiple-choice frame.

Format:

```
1 2
- -
2 text
```

NOTES:

1. One or more cards of the format above comprise group 2 of a question or multiple-choice frame. Each card represents a line of text to be presented to the student.

2. The text may contain form controls.

A8.2.12 GROUP Q3 CARD. A group Q3 card contains an anticipated response to student input to a question frame, or a switch set/reset instruction which controls the response-matching procedure.

Format 1:

1 2	3	4
3 tag	[+]	text

Format 2:

1 2

$$3\ \text{digit}\ \not{b}\quad [\text{SET}\ \not{b}]\quad \left\{\begin{array}{l}\text{PHONETIC}\\ \text{KEYWORD}\\ \text{ORDER}\end{array}\right\}\quad \not{b}\quad \left\{\begin{array}{l}\text{ON}\\ \text{OFF}\end{array}\right\}$$

NOTES:

1. One or more cards of either format above comprise group 3 of a question frame.

2. Response-matching for a question frame may be thought of as proceding from card-to-card in the group. Two actions result; depending on card format:

Format 1 -- The student response is matched to the anticipated response on the card. If the match is successful, control is transferred to the section of group 4 in the frame identified by the tag letter on the card. If the match is not successful, the next card in group 3 is examined.

Format 2 -- A switch is set/reset which controls the detailed logic of the response-matching procedure for following Format 1 cards.

3. Response-matching is performed on the "words" of the responses. A "word" in this context is a string of consecutive, non-blank characters (including digits, punctuation, and special characters). Blanks in a response are treated as word separators, but the number of blanks is not significant. In the absence of switch settings which permit more lenient matching criteria, a student response will match an anticipated response only if the "words" of both responses are identical in order and content.

Notes on Format 1:

4. Every question frame group 3 must contain at least one Format 1 card.

5. The tag must be a single alphabetic. The same tag may be applied to more than one anticipated response (so that the same group 4 actions may be taken on any one of them).

6. A + in column 3 indicates that the answer on the card is correct. When at least one group 3 card is so marked in the group, then those cards not marked are considered to contain incorrect answers; otherwise, no answer is either correct or incorrect.

7. Columns 4-72 contain the text of an anticipated response. Form controls (as form controls) may not be used.

Notes on Format 2:

8. Format 2 cards are optional in a question frame.

9. The digit may be any digit. Zero is preferred.

10. On entry to a question frame, all switches revert to their "normal" settings. At the start of a lesson, the "normal" setting of all switches is OFF. A Format 2 card containing the SET option takes effect in the current frame, and simultaneously changes the "normal" setting of the indicated switch.

11. The PHONETIC switch allows for misspelled words in a student response. If the switch is ON, then -- prior to any other matching procedure -- each word of each response is phonetically transformed. If two words are spelled to have about the same English pronounciation before transformation, they will be identical after transformation, permitting a character-by-character match. Phoneticization does not "destroy" the original student response. Hence, the switch may be set ON and OFF in the group, as desired.

12. The KEYWORD and ORDER switches may be in effect for a particular anticipated response (with or without prior phoneticization) in any combination. A successful match will be found:

WITH SWITCHES	ONLY IF THE STUDENT RESPONSE
KEYWORD OFF ORDER OFF	is identical, word-for-word, with the anticipated response
KEYWORD ON ORDER OFF	contains at least the words of the anticipated response in the same order
KEYWORD OFF ORDER ON	contains exactly the same words as the anticipated response in any order
KEYWORD ON ORDER ON	contains at least the words of the anticipated response in any order

13. When more than one switch option is used, the digit for each option must be different.

A8.2.13 GROIP M3 CARD. A group M3 card contains the text of an alternative answer to a multiple-choice frame.

Format 1:

1	2	3	4
3	tag	[+]	text

Format 2:

1	4
3	text

NOTES:

1. One or more cards of either format shown above comprise group 3 of a multiple-choice frame. The textual content of the group is displayed, with identifying tag letters, as a continuation of the text of group 2. An input cue is given at the end of the group 3 text display.

2. The tag must be a single alphabetic. The student indicates the choice of an alternative answer by responding with a tag letter. Control is then transferred to the section of group 4 of the frame identified by the tag letter.

3. If the student response is not exactly one tag letter from the list displayed, a message of the form

CHOOSE ONE LETTER FROM ABOVE

is displayed and the input cue is repeated.

4. A + in column 3 indicates that the alternative answer is correct. When at least one Format 1 card is so marked, those answers not marked are incorrect; otherwise, all answers are neither correct nor incorrect.

5. A complete display line consists of the contents of column 2, a blank (replacing column 3), and the contents of columns 4-72. By convention, each alternative answer is begun on a Format 1 card. Format 2 cards may be used to continue alternative answer text over more than one card. Form controls may be used.

A8.2.14 GROUP QM4 CARD. A group QM4 card contains actions to be performed in reply to a student response.

Format 1:

```
1  2

4  tag-string     [b̸ action-command] . . .
```

Format 2:

```
1

4  b̸ action-command     [b̸ action-command] . . .
```

NOTES:

1. One or more cards of either of the formats above comprises group 4 of a question or multiple-choice frame. The group is subdivided into sections by tag-strings as shown in Format 1.

2. A tag string consists of one or more consecutive tags. The tags must be either

a. group 3 tag letters; or

b. the special tag "-" (minus) denoting an "unanticipated response" to a question frame (one which failed to match any group 3 anticipated response).

3. In a question frame, the student response will correspond to a tag letter. In a multiple-choice frame, the student response will be a tag letter. In reply to the student response, the group 4 section which begins with a tag-string containing the tag is entered. (However, see the "RETRY" action command.) All action commands in the section (up to the next tag-string or the end of the group) are performed. If no action command encountered was an explicit transfer of control, an exit is then taken to the next frame in the lesson.

4. Format 2 may be used to continue a section of action commands over more than one card.

5. Format 2 may also be used to code action commands which precede any tag-string. Such "unconditional" action commands will be performed immediately after the student response is received and before performance of any action commands following a tag-string.

6. Every group 3 tag letter, and the minus tag in question frames, must appear in some tag-string. Format 1, with no following action command, may be used to satisfy this requirement when no action on a particular tag is desired in the frame.

A8.2.15 ACTION COMMANDS.

A8.2.15.1 BRANCH. The branch action command causes a transfer of control to a designated frame.

Format:

$$\text{B:} \quad \left\{\begin{matrix}\text{frame-number}\\ \text{frame-label}\end{matrix}\right\}$$

NOTES:

1. Control is transferred to the frame whose number or label follows the B:.

2. The B: action command takes effect immediately as it is performed. An action command which follows it will never be performed unless there is an intervening tag-string, ELSE, END, or end-of-frame.

A8.2.15.2 FEEDBACK. The feedback action command causes a line of text to be displayed to the student.

Format:

F: [text]

NOTES:

1. With text, the F: action command causes the text to be displayed to the student. The text may contain (or be) form controls.

2. Valid action commands may follow the text on the same card. Hence, the text is considered to be terminated by any sequence of characters which has the form of an action command. Specifically, any construction of the form:

$$\not{b} \quad \left\{\begin{matrix}\text{F}\\ \text{B}\\ \text{R}\\ \text{C}\\ \%\\ \text{S}\end{matrix}\right\}:$$

will be treated as marking the end of the text element.

3. The optional form of the F: action command without text may be used only in the following circumstances:

a. in a question or multiple-choice frame with at least one correct response marked in group 3; and

b. "conditionally" (following a tag-string) in the frame. In this case the executive makes a "random" draw from a table of affirmative or negative replies and displays it. The affirmative or negative nature of the reply is determined by the group 3 marking of the first tag in the tag-string that precedes the F:.

A8.2.15.3 RETRY. The retry action command permits the student to enter another response to a question or multiple-choice frame.

Format:

```
R:  [text]
```

NOTES:

1. The R: action command may only be used in question and multiple-choice frames.

2. Performance of the R: action command consists of two distinct operations: First, a line of text is displayed, then a transfer of control within the frame is done.

3. If text is supplied with the action command, that text is displayed. Form controls may be used.

4. Valid action commands may follow the text on the same card. Hence, the text is considered to be terminated by any sequence of characters which has the form of an action command. Specifically, any construction of the form

$$\cancel{b} \left\{ \begin{matrix} F \\ B \\ R \\ C \\ Z \\ S \end{matrix} \right\} \quad :$$

will be treated as marking the end of the text element.

5. If no text is supplied, a message of the form

WRONG, TRY AGAIN

is displayed.

6. Following the display of text, control is transferred back to the point in frame processing at which the student's input cue is generated.

7. In a question frame, all switches retain the settings they had at the time an exit-on-match was taken from group 3. Hence, the initial settings for evaluation of the new student response may differ from the previous initial settings.

8. Each time that a question or multiple-choice frame is entered, the executive sets a "retry counter" to one. Each time that an R: action command is performed, in the frame, the value of the retry counter is incremented by one. Student failure to reply with a valid tag letter in a multiple-choice frame does not increment the retry counter.

9. A given tag (letter or minus) may appear up to eight times in a question or multiple-choice frame group 4. Tags which are grouped into tag-strings need not be grouped identically for each appearance.

10. When the tag identifying a student response has been determined, the current value of the retry counter is used to "index" all appearances of the tag in group 4. If the retry counter is one, the group 4 section at the first appearance of the tag is entered. If the retry counter is two, the section at the second appearance of the tag is entered; and so on. If the value of the retry counter exceeds the number of appearances of the tag, the last appearance of the tag is entered. Note that this convention creates the possibility of an "unending loop" on re-tries.

11. There is only one retry counter, not one for each tag.

12. "Unconditional" action commands in a group 4 will be performed each time that a retry is given.

13. The R: action command takes effect immediately as it is performed. An action command which follows it will never be unless there is an intervening tag-string. In the particular case of multiple lines of text to be displayed before the retry is given, only the last line may be coded with an R:.

A8.2.15.4 CORRECT ANSWER. The correct answer action command may be used to display the correct answer to a question.

Format:

C:

NOTES:

1. The C: action command may only be used in a question or multiple-choice frame which has at least one answer marked correct in group 3.

2. The answer or tag displayed is the first one marked correct in group 3.

3. In a question frame, a two-line message of the form

THE CORRECT ANSWER IS

text

is displayed, where the text is taken from columns 4-72 of the anticipated response card.

4. In a multiple-choice frame, a message of the form

THE CORRECT ANSWER IS x

is displayed, where "x" is the tag letter of the alternative answer.

A8.2.15.5 ITEM. The item action command permits the value of an item to be set, incremented, decremented, or displayed.

Format 1:

%: item-name = item name [$\left\{\begin{matrix}+\\-\end{matrix}\right\}$ constant]

Format 2:

%: item-name = constant

Format 3:

%: item-name = p

NOTES:

1. An item is a numerical accumulator. The CDTS executive maintains 64 items for each student within each lesson. The values of all items (with the exception of reserved items) are saved from session to session during the progress of a lesson.

2. At the beginning of a lesson, all items (except reserved items) are set to zero. The range of permissible item values is all integers from 0 to 4095 inclusive.

3. If an item action command results in a nonpermissible value, a message of the form

```
     FRAME frame-number     [frame-label]

     item-name ILLEGAL VALUE
```

is displayed and the item retains its original value.

4. An item-name has the form

```
     ITEM digit    [digit]
```

with no embedded blanks (as shown). The one- or two-digit number is an "item number." Item numbers may range from 0 to 63 inclusive. If an item number is less than 10, the one- and two-digit forms (e.g., ITEM1 and ITEM01) may be used interchangeably in any context.

5. In Formats 1 and 2, the constant may be any decimal integer in the range 0 to 2047.

6. Format 1 causes the item named to the left of the = to be set to

a. the value of the item named to the right of the = or

b. the sum or difference of the value of the item named to the right of the = and the constant.

7. Format 2 causes the item named to be set to a constant.

8. Format 3 causes the value of the item named to be displayed. The display format (from display column 1) is two blanks and four digits (with leading zeros replaced by blanks).

9. The following items are reserved and set for the purposes indicated:

ITEM60 is set to 0 or 1 each time that a ?GET command is entered. A value of 0 indicates that the student is currently using a "teletype" or other continuous, hard-copy device. A value of 1 indicates that the student is currently using a CRT-type device.

ITEM63 is initially set to zero. It is used for interface with the CDTS student records system.

10. Reserved items are not protected from setting or testing by the coder. ITEM60, in particular, may be incremented and tested to determine if the student has ended a session between two points in a lesson.

A8.2.15.6 SUBSYSTEM. The subsystem action command permits a standard H6000 time-sharing subsystem to be entered from CDTS with a return to CDTS.

Format:

```
S: name
```

NOTES:

1. The name is the name of a standard H6000 time-sharing subsystem (CARD, EDIT, ABC, etc.).

2. On performance of the S: action command, the CDTS executive is saved in its current status and the named subsystem is entered. The first thing the student will see is the standard starting sequence for the subsystem (such as *).

3. The student may continue with operations in the subsystem for as long as he wishes. A time-sharing command which "normally" terminates the subsystem (usually DONE) will return to CDTS. A message of the form

```
CDTS RETURNS
```

is then displayed.

4. Control returns to CDTS as if an ordinary action command (such as F:) had been performed at that point in the lesson.

5. Additional subsystem calls from the subsystem entered may go only one level down (i.e., EDIT could be called from CARD, but RUNOFF could not be entered in turn from EDIT).

6. A student action which "abnormally" terminates the subsystem will terminate CDTS. Recovery can be made as if a ?CLOSE had been entered.

7. This action command and the associated ?SYST on-line command use DRL CALLSS. Refer to the Honeywell TSS Programmer's Reference Manual for additional information.

A8.2.16 GROUP D2 CARD. A group D2 card contains conditional and unconditional action commands, and tests on items and flagged frames which evaluate the controlling conditions.

Format 1:

```
1

2 ƀ IF ƀ condition [ƀ {AND} ƀ condition] . . . [ƀ action-command] . . .
                      {OR }
```

Format 2:

```
1

2 ƀ {AND} ƀ condition [ƀ {AND} ƀ condition] . . . [ƀ action command] . . .
    {OR }                 {OR }
```

Format 3:

```
1

2 ƀ ELSE  [ƀ action-command] . . .
```

Format 4:

```
1

2 ƀ END
```

Format 5:

```
1

2 ƀ action-command   [ƀ action-command] . . .
```

NOTES:

1. One or more cards in the formats above comprise group 2 of a decision frame. The group may consist of:

a. "conditional statements" containing action commands which will be performed only if given tests are satisfied; or

b. "unconditional statements" consisting of a series of action commands which will always be performed.

2. A conditional statement has the general construction:

```
IF condition [{AND} condition] . . . action-command [action command] . . .
              {OR }

[ELSE action-command  [action-command] . . . ]

END
```

(Note that mandatory blanks are not shown.) Such a statement may be continued over several cards using Format 1 to 5 as needed. The numbered formats limit the points at which a statement may be "broken" for continuation. Note, for example, that ELSE may not follow action commands on the same card.

3. One or more conditions conjoined by AND and OR form a "conditional expression". In the evaluation of conditional expressions (true or false), AND takes precedence over OR. The following example

condition-1 AND condition-2 OR condition-3

is true only if <u>both</u> conditions 1 and 2 are true, <u>or</u> condition-3 is true.

4. In a conditional statement, the action commands following the conditional expression are performed only if the conditional expression is true. If an ELSE is coded, the action commands following it are performed only if the conditional expression is false.

5. An unconditional statement consists of a sequence of action commands. Format 5 is used as needed.

6. An END card (Format 4) is required only to separate a conditional statement from a following unconditional statement (otherwise there is no recognizable break in the sequence of action commands). Its use in all other contexts is optional.

A8.2.17 CONDITION.

A8.2.17.1 ITEM. An item condition permits tests on the value of an item.

Format:

```
item-name ƀ     ⎧ GR ⎫  ƀ constant
                ⎪ GQ ⎪
                ⎨ EQ ⎬
                ⎪ LQ ⎪
                ⎩ LS ⎭
```

NOTES:

1. See the discussion of the item action command for item names.

2. The relationals in this condition are defined as follows:

GR	is greater than
GQ	is greater than or equal to
EQ	is equal to
LQ	is less than or equal to
LS	is less than

3. The constant may be any decimal integer in the range 0 to 4095.

4. The condition is true or false according to the ordinary sense of the arithmetic comparison in the format.

A8.2.17.2 FLAGGED FRAME. Flagged frame conditions test previous student responses to flagged frames.

Format 1:

```
⎧ [ALL] ⎫  ƀ  ⎧ RIGHT ⎫  ƀ frame-number   [- frame-number]
⎪ EVER  ⎪     ⎨ WRONG ⎬
⎨ NONE  ⎬     ⎩ SEEN  ⎭
⎩ LAST  ⎭
```

Format 2:

$$\left\{\begin{matrix}\text{GR}\\ \text{GQ}\\ \text{[EQ]}\\ \text{LQ}\\ \text{LS}\end{matrix}\right\}\quad \not{b}\ \text{constant}\ \not{b}\quad \left\{\begin{matrix}\text{RIGHT}\\ \text{WRONG}\\ \text{SEEN}\end{matrix}\right\}\quad \not{b}\ \text{frame-number}\quad \text{[- frame-number]}$$

Format 3:

frame-number, tag-list [b̸ frame-number, tag-list] . . .

NOTES:

1. All frames explicitly referenced in these formats must be flagged.

2. A given flagged frame may be entered several times as a result of branching within a lesson. Since the flagged frame table is a chronological list, there may be several entries in the table for the same flagged frame. This situation may result in exceptions to the general evaluation of some conditions. These exceptions will be noted below.

3. Retries within a flagged frame do not generate multiple entries in the flagged frame table. The single entry posted will reflect the last student response to the frame. The flagged frame table contains space for 250 entries.

4. A decision frame may be flagged. If so, all conditions within the frame pertain only to the portion of the flagged frame table chronologically forward from (and excluding) the most recent, previous appearance of that decision frame. If there is no previous appearance, or the frame is not flagged, the entire flagged frame table is examined. This convention applies to all conditional evaluations described below. It may be used to exclude unwanted entries from evaluation (see note 2, above).

Notes on Formats 1 and 2:

1. Formats 1 and 2 operate on a "range" of flagged frames indicated by the

frame-number [- frame-number]

construction in the formats. A range will consist of a single frame, or every flagged frame in the source lesson between (and including) the two frames indicated. If a multi-frame range is coded, the frame to the left of the dash must precede the frame to the right of the dash in the source lesson.

2. Not every frame in a range must be flagged. It is the responsibility of the coder to determine which frames should be flagged to satisfy the intent of a condition. In the remainder of these notes, references to "frames within a range" will be assumed to pertain only to the flagged frames within the range.

Notes on Format 1:

1. If the first word (ALL, EVER, etc.) is omitted from the format, ALL is assumed.

2. The general evaluations of conditions in this format are discussed below. Exceptions are noted.

a. ALL RIGHT

This condition is true only if the student has entered a correct response to every frame in the range.

Exceptions: Multiple entries for the same frame are examined separately. Hence, the condition may be false even though the student's most recent responses were correct. If no entry is found for a frame in the range, it is treated as if a correct response has been entered. (This condition is evaluated by search for a negating case.)

b. ALL WRONG

This condition is true only if the student has entered an incorrect response for every frame in the range.

Exceptions: Multiple entires for the same frame are examined separately. Hence, the condition may be false even though the student's most recent response was incorrect. If no entry is found for a frame in the range, it is treated as if an incorrect response had been entered. (This condition is evaluated by search for a negating case.)

c. ALL SEEN

This condition is true only if every frame in the range has been entered.

d. EVER RIGHT

Tnis condition is true only if the student has entered at least one correct response to every frame in the range.

Exceptions: EVER RIGHT differs from ALL RIGHT in that multiple entries for the same frame will not invalidate a correct response. It need not, however, be the most recent entry. An entry for every frame in the range must be found; else, the condition is false.

e. EVER WRONG

This condition is true only if the student has entered an incorrect response to any frame in the range.

Exceptions: Multiple entries for the same frame are evaluated separately. Hence, the condition may be true even though the student's most recent responses were correct. If no frame in the range has an entry, the condition is false. If some frames are found, evaluation depends on those frames. (This condition is evaluated as the logical negation of ALL RIGHT.)

f. EVER SEEN

This condition is true only if every frame in the range has been entered.

NOTES: EVER SEEN and ALL SEEN are the same condition.

g. NONE RIGHT

This condition is true only if the student has entered an incorrect response to every frame in the range.

Exceptions: NONE RIGHT and ALL WRONG are the same condition. Exceptions for ALL WRONG apply.

h. NONE WRONG

This condition is true only if the student has entered a correct response to every frame in the range.

Exceptions: NONE WRONG and ALL RIGHT are the same condition. Exceptions for ALL RIGHT apply.

i. NONE SEEN

This condition is true only if no frame in the range has been entered.

Notes: NONE SEEN is not the (strict) logical negation of EVER SEEN, since neither is true when some but not all frames have been seen.

j. LAST RIGHT

This condition is true only if the student's most recent response to every frame in the range was correct.

Notes: An entry must be found for every frame in the range, else the condition is false.

k. LAST WRONG

This condition is true only if the student's most recent response to every frame in the range was incorrect.

Note: If no entry is found for a frame in the range, the condition is true. (LAST WRONG is the logical negation of LAST RIGHT.)

l. LAST SEEN (single-frame range)

This condition is true only if the flagged frame most recently entered (excluding the current decision frame, if flagged) is the designated frame.

m. LAST SEEN (multiple-frame page)

This condition may be used to test for previous branching as follows:

The area of the flagged frame table bounded by the most recent entries for the "lower" and "upper" frames in the range is identified.

The condition is true only if all frames in the area defined fall within the range.

NOTES: If no entries are found for the bounds frames, the condition is false. The "upper" frame of the range need not be LAST SEEN in the single-frame sense for the condition to be true.

Notes on Format 2:

1. The relationals (first word of the format) are defined as follows:

GR	is greater than
GQ	is greater than or equal to
EQ	is equal to
LQ	is less than or equal to
LS	is less than

If the relational is omitted, EQ is assumed.

2. The constant may be any decimal integer in the range 0 to 4095.

3. This condition is evaluated true or false according to the ordinary sense of an arithmetic comparison of the form

counter relational constant

The value of the counter depends on an analysis of the flagged frame table and the RIGHT/WRONG/SEEN option, as follows:

a. RIGHT

The value of the counter is the number of frames in the range to which the student gave a correct response.

b. WRONG

The value of the counter is the number of frames in the range to which the student gave an incorrect response.

c. SEEN

The value of the counter is the number of frames in the range which were entered.

4. Note that multiple entries for the same frame are counted separately.

Notes on Format 3:

1. A tag-list is a string of consecutive characters which may be any of the following:

A group 3 tag letter

A + denoting any correct response

A - denoting any incorrect response

2. This "pattern" condition is true only if

a. The most recent entries for each frame identified by number in the pattern are found in the order coded; and

b. The student response to each of these frames matches at least one tag in the corresponding tag-list.

3. As indicated in the formats of group D2 cards, a pattern condition may not be continued over more than one card. The effect of continuation may be achieved by using the logical connective AND, and repeating the last element of the previous pattern (thereby ensuring the correct sequence of most recent entries.)

A8.3 CODING EXAMPLE. The following short coding example illustrates the form of actual CDTS coding, as well as several features of the language. A possible remote terminal dialog resulting from presentation of this material is also shown.

```
LELEC01      SEQ
  .
  .
  .
1FRAME 101.0  Q1
2\\
2IF THE VOLTAGE DROP ACROSS A 4 OHM RESISTOR IS 20 VOLTS, WHAT IS THE
2CURRENT FLOWING THROUGH IT?
30 KEYWORD ON
3A+5
3A+FIVE
4A F:CORRECT. B:102
4-
1FRAME 101.2  D
2IF GR 1 SEEN 101
2F:THE CORRECT ANSWER IS 5 AMPS, SINCE I=E/R AND 5=20/4.
2F:LET'S TRY ANOTHER PROBLEM.  B:102
2ELSE F:NO.  LET'S REVIEW A LITTLE.
1FRAME 101.4  M
2\\
2WHICH OF THE FOLLOWING FORMS OF OHM'S LAW WOULD YOU USE TO SOLVE
2THIS PROBLEM?(
3A VOLTAGE = CURRENT TIMES RESISTANCE          (E=I*R)(
3B+CURRENT = VOLTAGE DIVIDED BY RESISTANCE     (I=E/R)(
3C RESISTANCE = VOLTAGE DIVIDED BY CURRENT     (R=E/I)(
4AC F:NO. YOU WOULD USE THE FORM IN WHICH THE UNKNOWN VALUE IS BY
4R:ITSELF ON THE LEFT-HAND SIDE. TRY AGAIN.
4AC F:STILL WRONG. I THINK YOU NEED MORE REVIEW.  %:ITEM1=2  B:OHMS
4B  F:RIGHT. NOW LET'S TRY THAT PROBLEM AGAIN. B:101
1FRAME 102.0   Q
  .
  .
  .
```

The previous coding example would generate the following student presentation:

(page)

IF THE VOLTAGE DROP ACROSS A 4 OHM RESISTOR IS 20 VOLTS, WHAT IS THE CURRENT FLOWING THROUGH IT? * 80 (student response) NO. LET'S REVIEW A LITTLE.

(page)

WHICH OF THE FOLLOWING FORMS OF OHMS LAW WOULD YOU USE TO SOLVE THIS PROBLEM?

A	VOLTAGE = CURRENT TIMES RESISTANCE	(E=I*R)
B	CURRENT = VOLTAGE DIVIDED BY RESISTANCE	(I=E/R)
C	RESISTANCE = VOLTAGE DIVIDED BY CURRENT	(R=E/I)

* A (student response) NO. YOU WOULD USE THE FORM IN WHICH THE UNKNOWN VALUE IS BY ITSELF ON THE LEFT-HAND SIDE. TRY AGAIN. * B (student response) RIGHT. NOW LET'S TRY THAT PROBLEM AGAIN.

(page)

IF THE VOLTAGE DROP ACROSS A 4 OHM RESISTOR IS 20 VOLTS, WHAT IS THE CURRENT FLOWING THROUGH IT? * 5 AMPS (student response) CORRECT.

. . .

If the student's response to the last question had been

* 10 AMPS (he would then see ...) THE CORRECT ANSWER IS 5 AMPS, SINCE I=E/R AND 5=20/4. LET'S TRY ANOTHER PROBLEM.

. . .

A8.4 ERROR MESSAGES. The CDTS compiler generates four types of error messages:

a. Type 1 - Syntax errors

Type 1 (syntax) errors are immediate errors in coding format. The error message will appear in the source listing following the line in which the error is found. A message has the following format:

```
*text                        99 -> 99
```

The message text describes the error and is self-explanatory. The two numbers following the text indicate the initial and final positions (card columns) of the card scan at the time the error was detected.

When an error is detected, the compiler attempts to resume normal scanning as soon as it finds something that appears to be a language element. In some cases, this will cause error messages to "cascade" from a single, actual error. Careful inspection of the erroneous code will permit the nature of the actual error to be determined.

b. Type 2 - Context errors

Type 2 (context) errors generally pertain to inconsistencies between parts of a frame which may be legal in themselves. The error message will appear in the source listing following the line in which the inconsistency became apparent. A message has the format:

```
**text                    99 -> 99
```

The message text describes the error and is self-explanatory. The two numbers following the text are defined as for type 1 errors, but may not be meaningful here.

The special message

```
**NOT PROCESSED
```

indicates that, as a result of some error, the compiler has not attempted to scan a card. (For example, if the frame type is unknown.) If a card is flagged as "not processed" it may contain undetected errors. Such cards should be double-checked when other errors are being corrected.

A summary listing of type 1 and 2 errors is printed at the end of the source listing. Each line has the form

```
ERR  original error message       AT 9999
```

The number "9999" refers to the line number at which the error was flagged.

c. Type 3 - Cross-reference errors

Type 3 (cross-reference) errors pertain to errors in references between parts of the lesson. Error messages appear following the summary listing of type 1 and 2 errors. They have two forms:

```
***UNDEFINED          mnemonic     AT 9999

***MULTIDEFINED       mnemonic     AT 9999
```

The significance of these message is summarized in Table A8-1.

TABLE A8-1

CDTS COMPILER ERROR MESSAGES

UNDEFINED	LABLxxxxxx	AT 9999	A reference to a nonexistent frame label xxxxxx appears in source line 9999.
	FNUM NNND		A reference to a nonexistent frame number nnn.d appears in source line 9999.
	FFLG nnnd		A reference which requires frame nnn.d to be flagged appears in source line 9999. Frame nnn.d is either not flagged or nonexistent.
	ACTS nnndt		Tag letter t in frame nnn.d appears in group 3, but not in group 4.
	ACTS nnnd-		Question frame nnn.d does not contain a minus tag (provision for an unanticipated response) in group 4.
	ANSW nnndt		"Random" feedback (F: without text) cannot be given for tag t in frame nnn.d because no + appears in group 3.
MULTIDEFINED	*LABL xxxxxx*		*Frame label xxxxxx appears* on more than one frame. Each appearance line 9999) will print a separate message.
	ANSW nnndt		Tag t in frame nnn.d is marked as both correct and incorrect.

NOTE: Other error messages in the general type 3 format may appear. These messages are secondary consequences of errors which have been flagged elsewhere. Direct corrective action is required only for a type 3 error which appears in the table above.

d. Type 4 - Compiler Error

A possible compiler malfunction may result in an error message of the form:

```
****LOST LOCATION FOR  additional text
```

If this message appears, save all available listings, input data, etc., and notify compiler maintenance personnel immediately.

A8.5 CDTS COMPILER.

a. Program Application. The CDT Compiler is applicable only to those sites developing site-unique CDT courses. The CDT Compiler analyzes a source lesson input by the coursewriter and produces an object file as output. The information on the object file is placed on a lesson file by the CDTS General Utility Loader. The lesson file is used by the CDTS Executive to present the lesson.

b. Input/Output. The file codes associated with the execution of the program are listed below. All files are locally maintained.

FILE/CODE	DESCRIPTION OF FILE
CS	A deck, or BCD card-image file, containing the source lesson to be compiled.
PC	A source listing with diagnostic messages
W1	A compiler work file
W2	A compiler work file
FO	A generated object file
WR	The assigned file code for a sort collation file

c. Job Setup. The following control cards comprise an activity setup for the program. The activity limits and temporary file sizes shown will be sufficient for almost all applications. This represents half of the CDTS General Utility Loader Program which is contained on your release under CDTSCDTS/JCL/Load. For a complete listing of this program refer to paragraph g.

```
1      8       16

$      LIBRARY LB
$      USE     VCCMSJ
$      ENTRY   VCCMSJ
$      EXECUTE
$      PRMFL   LB,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  10,30K,6K,10000
$      DATA    CS
$      SELECTA (Cat/File-String of an ASCII Source Lesson File)
$      SYSOUT  PC
$      FILE    W1,,20
$      FILE    W2,,20
$      FILE    WR,,20R
$      FILE    FO,X1S,20     (Note 2)
```

NOTE 1. This deck setup assumes that the lesson was built as an ASCII file under time-sharing. A PRMFL card may be substituted for the DATA and SELECTA cards if the file is BCD. If the coursewriter is using cards, the lesson source deck would replace the SELECTA card.

NOTE 2. This file is saved to be used by the CDTS General Utility Loader in a later activity.

A8.6 CDTS GENERAL UTILITY LOADER.

a. Program Application. The CDTS General Utility Loader is applicable only to those sites developing site-unique CDT courses, and for selective lesson loading.

b. Input/Output. The file codes associated with execution of the program are listed below. Specified file-ID's depend on the application of the program.

FILE CODE	DESCRIPTION OF FILE
F1	A tape input file. When the application is selective lesson loading, the courseware tape will be assigned to this file code.
FA	A disk input lesson file.
F2	A tape output file in the same format as tape TA. With the ADD option, this tape becomes the new-master.
FB	A disk output lesson file.
FX	A tape old-master file containing courseware to be updated.
FO	An "object" file generated by the CDTS Compiler.

c. Skeleton Job Setup. The following control cards (JCL) and program control cards comprise a job setup for this program. In the job setup below, all file cards are shown for reference purposes. A given program function requires only the file cards for the files it uses. Sample deck setup is in paragraph g.

```
1      8       16

$      SNUMB
$      IDENT
$      USERID
$      LIBRARY LB
$      USE     VCUTSJ
$      ENTRY   VCUTSJ
$      EXECUTE
$      PRMFL   LB,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  (run time),18k,,1000
$      TAPE    F1,...
       (or)
$      PRMFL   FA,R,R,...
$      TAPE    F2,...
       (or)
$      PRMFL   FB,R/W,R,...
$      TAPE    FX,...
$      FILE    FO,X1R
       ...     (Program Control Cards)
$      ENDJOB
```

d. Program Control Cards. Program control cards identify the utility functions to be performed and the lessons on which they operate. A complete deck consists of a media card followed by function cards. (PCN SE776A001)

(1) MEDIA CARD. The first card of each program control card deck must specify the media to be used. It has the format:

CARD-COL	FIELD
1-4	Input medium
8-11	Output medium
16-19	Option

(a) The input medium must be TAPE, DISK, or blank (some functions do not require an input file). If the input medium is TAPE, file F1 must be defined. If the input medium is DISK, file FA must be defined.

(b) The output medium must be TAPE or DISK. If the output medium is TAPE, file F2 must be defined. If the output medium is DISK, file FB must be defined.

(c) The option field may contain INIT or blank. If the INIT option is used, the output file is to be initialized as an "empty" lesson file. It must be used if file FB contains no prior lesson data. Any prior lesson data on file FB will be destroyed.

(2) FUNCTION CARDS. The remainder of the program control cards consists of function cards. The effects of function cards are discussed in paragraph e. The format of function cards is:

CARD-COL	FIELD
1-4	Function
8-13	Lesson name

(a) The function field may contain COPY, DELE, ADD or LOAD. As a convenience, a blank function field causes the card to be processed as if it contained the last non-blank function field encountered. The first function field may not be blank. The lesson name field must contain a lesson name, left-justified, with blank fill.

(b) With the exception of the ADD function, each function card is processed in the order read, independently of the other function cards. Subject to restrictions on media, the COPY, DELE, and LOAD functions may be repeated and combined in any desired way. The ADD function may not be used with any other function. Normally, the DELE and COPY functions may be sequenced anywhere relative to the rest of the deck, since this function references only file FB.

e. Utility Functions. The program will perform the following functions:

COPY

The COPY function causes the associated lesson to be copied from file F1 or FA to file F2 or FB. Files used: F1 or FA, F2 or FB.

DELE

The DELE function causes the associated lesson to be deleted from file FB. The file space formerly occupied by the lesson is available for re-use. Files used: FB.

ADD

The ADD function causes all the lessons on file FX to be copied to file F2. During the course of this process, the lessons named in the ADD function cards are extracted from file F1 or FA and merged into the file being built on F2. If a lesson name referenced by an ADD card is already present on tape FX, the version of FX is ignored. Files used: FX, F1 or FA, F2.

LOAD

The LOAD function causes an object lesson file on file FO to be converted to executable format and written to file FB or to be converted to the equivalent tape format and written to file F2. File FO is an output file from the CDTS Compiler which will ordinarily be transmitted to the General Utility from a previous compile activity in the same job unit. Only one LOAD function may be performed in an execution of the program. The lesson name of file FO must match the lesson name on the program control card. Files used: FO, F2 or FB.

f. Output Products. When the output medium is DISK, the General Utility produces a Lesson File Inventory report showing the contents and file space utilization of the output file. Preceding this report, a listing of Program Control Cards is produced for reference. If any errors occur, and error message will be printed immediately below the card image in the processing on which the error occurred. Messages have the format:

**text

Table A8-2 shows all error messages and action to be taken.

TABLE A8-2

UTILITY ERROR MESSAGES

Message	Condition	Action
BAD FORMAT ON F1 or BAD FORMAT ON FX	End of file was reached on the indicated tape while reading under control of a lesson size block count.	Notify the originator of the tape.
DUPLICATE LESSON NAME	The output disk file already contains a lesson with the given name.	Use DELE to remove the old version. Rerun.
ILLEGAL FUNCTION WITH ADD	An attempt has been made to combine another function with ADD.	Run as separate jobs.
ILLEGAL MEDIA CARD	The media card contains incorrect or incorrectly formatted information.	Correct and rerun.
LESSON NOT FOUND	The lesson cannot be found on file TA or DA. This error is also given for an attempt to find a lesson on tape out of collating sequence order.	Correct and rerun.

TABLE A8-2 (Continued)

UTILITY ERROR MESSAGES

Message	Condition	Action
NO ACTIVITY	No program control cards were included with the job.	Correct and rerun.
NO DATA ON OBJECT FILE	The object file is empty. This is probably due to compile errors.	Check compile activity.
NO INPUT DEFINED	No input was identified on the media card.	Correct and rerun.
NO INPUT (UPDATE) DEFINED	No input/update file (TA or DA) was defined for an ADD.	Correct and rerun.
NO SPACE IN INDEX	The output disk file already contains 106 lessons.	Use another lesson file.
NOT ADDED - OUT OF ORDER	The user has attempted to add a lesson out of collating sequence order.	Correct and rerun.
OBJECT NAME DOES NOT MATCH	The lesson name on the object file does not match the lesson name on the function card.	Correct where needed and rerun.
OUTPUT MUST BE TAPE	An attempt has been made to use the ADD function for DISK output.	Correct and rerun.
UNKNOWN FUNCTION	The function indicated is not ADD, COPY, DELE, or LOAD.	Correct and rerun.

g. Sample Input. The following program control cards illustrate the use of the two major options of the General Utility program.

(1) Lesson LSN010 is compiled and loaded to a new lesson file which is initialized.

```
1      8       16

$      SNUMB
$      IDENT
$      USERID
$      LIBRARY LB
$      USE     VCCMSJ
$      ENTRY   VCCMSJ
$      EXECUTE
$      PRMFL   LB,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  10,30K,6K,10000
$      PRMFL   CS,R,S,(Cat/File-String of LSN010)
$      SYSOUT  PC
$      FILE    W1,,20
$      FILE    W2,,20
$      FILE    WR,,20R
$      FILE    FO,X1S,20
$      LIBRARY LC
$      USE     VCUTSJ
$      ENTRY   VCUTSJ
$      EXECUTE
$      PRMFL   LC,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  10,18K,,1000
$      FILE    FO,X1R
$      PRMFL   FB,R/W,R,(Cat/File-String of the Lesson File)
       DISK    INIT
LOAD   LSN010
$      ENDJOB
```

(2) A new version of LSN010 is compiled and loaded to an existing lesson file. The previous versions of LSN010 and LSN020 are deleted.

```
1      8       16

$      SNUMB
$      IDENT
$      USERID
$      LIBRARY LB
$      USE     VCCMSJ
$      ENTRY   VCCMSJ
$      EXECUTE
$      PRMFL   LB,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  10,30K,6K,10000
$      PRMFL   CS,R,S,Your/Source/LSN020
$      SYSOUT  PC
$      FILE    W1,,20
$      FILE    W2,,20
$      FILE    WR,,20R
$      FILE    FO,X1S,20
$      LIBRARY LC
$      USE     VCUTSJ
$      ENTRY   VCUTSJ
$      EXECUTE
$      PRMFL   LC,R,S,CDTSCDTS/RSTARLIB
$      LIMITS  10,18K,,1000
$      FILE    FO,X1R
$      PRMFL   FB,R/W,R,CDTSCDTS/LSN/LESSONS
       DISK
DELE   LSN010
DELE   LSN020
LOAD   LSN010
$      ENDJOB
```

A8.7 SITE UNIQUE LESSONS (SUL). The purpose of this paragraph is to identify CDT lessons which have been developed and used at local CDT sites, but which are not part of the general CDT release. Included with the information about each lesson is the address of the office of primary responsibility (OPR) which maintains the lesson. You must write the OPR to obtain a copy of the lesson. If you have a lesson which you believe should be available to the sites, please send a magnetic tape copy of the source to:

3300 TCHTW/TTGXZ
Attn: WWMCCS CDT
Keesler AFB MS 39534

The CDT Section will check your CDT logic, assign the lesson a SUL number, document it in this paragraph, and return the tape to you.

a. All lessons in the paragraph are candidates for the general CDT release if the interest/usage warrants it.

b. SUL000. This lesson is used to demonstrate the capabilities of the CDT System. It also contains utilization statistics and a trivia quiz. This lesson is updated frequently by the CDT Section. Approximately 0.75 hours of on-line time is required to take this lesson.

DEVELOPMENT CENTER PROCESSING INVENTORY

A9.1 OBJECT LIBRARY PROGRAMS. Included in all releases are two objecc libraries, RSTARLIB and RSTARTWO, under CDTSCDTS. These libraries contain, in object form, the programs listed in paragraph 9.2. With the exception of VCSRSJ (Student Activity Report), VCCMSJ (compiler), and VCUTSJ (loader), the programs in the object libraries will be used only by the WWMCCS CDTS Development Center. AFM 171-752 contains a detailed description of these programs.

A9.2 RUN INVENTORY. The following inventory is provided to give CDTS release recipients a brief explanation of the two object libraries contained on the release. The format used in the inventory is: Program name - program title - program description.

a. VCSØSJ - Old trace tape preprocessing - Modifies trace tapes created by the old version of program VCSRSJ.

b. VCS1SJ, VCS2SJ, and VCS3SJ - Trace tape event processing - Processes trace tapes received from CDTS user sites. Creates event records.

c. VCS5SJ - Event merge - Merges event records.

d. VCSMSJ - Event retrieval - Extracts information from the master event file.

e. VCCRSJ - Display critique file - Displays the answers to the student critique.

f. VCCØSJ - Initialize critique file - Initializes the critique file.

g. VCTØSJ - Trace tape merge - Merges ØØ type trace records.

h. VCT2SJ and VCT3SJ - Trace tape lesson validation - Creates a listing used to validate CDTS lessons.

i. VCT4SJ and VCT5SJ - Unexpected response reporting - Creates a listing of unexpected responses.

j. VCTSSJ and VCTTSJ - Validation totals reporting - Creates the detail and summary totals listings.

k. VCXASJ - CDTS executive program - Drives the Computer Directed Training System.

l. VCCMSJ - CDTS compiler - Compiles CDTS lessons.

m. VCSRSJ - CDTS student record summary program - Produces the CDTS student utilization report on trace tape.

n. VCUTSJ - CDTS loader - Loads object form of CDTS lessons for execution by CDTS executive.

COMPUTER DIRECTED TRAINING SYSTEM (CDTS) COURSEWRITER (H6000-CDT) COURSE EXHIBITS

SYMBOL	SPECIAL PURPOSE
* (Asterisk)	Cue output by the program to indicate to the user that information may be inserted.
\ (Backslash)	Indicates to the program in the Lesson execution mode that a carriage return is to be executed whenever encountered as the last character on a line. An additional line feed is performed after the carriage return.
\\\\ (Backslashes)	Used by the program in Lesson execution mode to indicate a full page (approximately 13 lines of text) and a new page is needed.
: (Colon)	Indicates to the program to evaluate the preceding character in group 2 of a decision frame, group 4 of a multiple choice or question frame to determine the subsequent action. A letter indicates a legal action command may follow; whereas, the percent (%) symbol indicates a legal ITEM name may fcllow.
% (Percent)	Indicates to the program in the Lesson execution mode that a legal ITEM name may follow.
+ (Plus Sign)	Used immediately adjacent to an answer TAG in a group 3 and indicates that the answer is to be considered correct and also indicates a positive response in a conditional statement.
- (Minus Sign)	Used immediately adjacent to an Action Command in a group 4 as an indicator to the program which statements are reserved for unanticipated responses.
- (Hyphen)	Used as a connector to group frames in conditional statements.
= (Equal)	Means equivalency in mathematical expressions or means replace the contents of the ITEM on the left with the evaluation of the expression on right.
, (comma)	Used in conditional statements as a separator between components comprising a cluster of responses.
SPACE	Used to separate entities comprising legal expressions.

EXHIBIT CDT010-1

SYMBOL	SPECIAL PURPOSE
A-Z (Alphabetic)	Used as tags for answers in groups 3 and 4.
0-9 (Numeric)	Indicates to the program that a legal switch setting (KEYWORD, ORDER, or PHONETIC) may follow which affects the processing of the inserted response.
F: (Action Command)	(Feedback)
R: (Action Command)	(Wrong, Try Again)
C: (Action Command)	(The correct answer is ....)
B: (Action Command)	(Branch to ....)
%:ITEM	(ITEM SET)

EXHIBIT CDT010-1 (Cont)

ACTION COMMAND FEEDBACK MESSAGES

MESSAGE	POSITIVE	NEGATIVE
CORRECT	1/8	
FALSE		1/8
GOOD	1/8	
INCORRECT		1/8
NO		1/8
NOT TRUE		1/8
RIGHT	4/8	
TRUE	1/8	
WRONG		4/8
YES	1/8	

Frequency of Occurrence;
e.g., 1 out of 8 times

EXHIBIT CDT010-2

CDTS PARAMETERS

LESSON

9999 frames (0.1 to 999.9) maximum for each lesson. The number of frame headers (group 1) equals lesson length.

FRAME

There are three (3) types of frames: multiple-choice (M), question (Q), and decision (D). All types must contain a group 1.

MULTIPLE-CHOICE (M)

May contain all four groups or it may be split to contain only 1,2; 1,3,4; 1,2,4; or 1,4.

QUESTION (Q)

Same as M frame: 1,2,3,4; 1,2; 1,3,4; 1,2,4; or 1,4.

DECISION (D)

Consists of groups 1 and 2 only. The decision frame is used to evaluate previously established conditions; dictate new conditions; and direct student's path through the lesson using the branch (B:) capability.

LABELS

Any or all frames may be labelled.

ITEMS

64 ITEM names (0 to 63) maximum for each lesson. Values can range from 0 to 2095. An item name can be set to equal a value (%:ITEM1=1); incremented (%:ITEM1=ITEM1+1); incremented to another item plus a value (%:ITEM1=ITEM2+1); decremented (%ITEM1=ITEM1-1); or displayed (%:ITEM1=P).

GROUP ONE

Frame header (all frames)

GROUP TWO

Text or questions

GROUP THREE

Answers (no punctuations permitted)

GROUP FOUR

Actions

EXHIBIT CDT010-3

GLOSSARY OF ABBREVIATIONS AND TERMS

1. Abbreviations

AFB	Air Force Base
CDT	Computer Directed Training
CDTS	Computer Directed Training System
I/O	Input/Output
D	Decision
M	Multiple Choice
Q	Question

2. Terms

ALL
A relational operator in a decision statement to indicate that a condition was satisfied.

ALPHABETICS AND NUMERICS
This term refers to the construction of a lesson name or label composed of letters or digits, the first of which must be a letter.

AND
A connective used in a decision statement to mean both.

ANSWER TAG
A letter (A-Z) used to differentiate inserted answers in group 3 and the resulting actions in group 4 if answers are matched.

B: (Branch)
An action command available to the course designer which causes a transfer of control to another frame (number or label), and alters the sequence of lesson presentation.

Branch
See B: (Branch)

C: (Correct)
An action command which instructs the program to print "THE CORRECT ANSWER IS" followed by the answer associated with the plus (+) sign. If more than one answer has a plus sign, the first plus sign answer is selected by the program.

Character Set
A string of characters (symbols, digits, letters) set apart with blanks. (Example: All these words are character sets: these are also, a65, Meanl, hex9.)

Correct
See C: (Correct)

Course
A sequence of related lessons.

D
D for Decision frame.

DECISION FRAME - D
This frame is normally used to test conditional statements which are executed depending upon responses to frames which have been flagged, or upon ITEM settings.

Indicates to the program that a legal switch setting (PHONETIC, KEYWORD, ORDER) may follow which affects the processing of an inserted response.

ELSE
A connective used in a compound decision statement which causes the non-true condition to be executed.

END
A connective used in a decision statement to separate a conditional statement from a subsequent unconditional statement.

EQ
A relational operator used in a decision statement meaning "is equal to."

EVER
A condition used in a decision statement to indicate that the condition specified occurred at least once.

F: FEEDBACK
An action command which displays appropriate messages based upon trainee response. After printing the message, the program proceeds to the next frame sequence or follows a subsequent command.

FEEDBACK
See F: (Feedback)

FLAG
Instructs the program during off-line lesson generation to associate a decision statement indicator for all "Q" and "M" frame types within the lesson table. All frames are thereby established for use by decision statements constructed within the lesson. If not used, the course designer must insert the decision statement indicator into each frame he desires to have considered for use by a decision statement.

FRAME
The unit by which course content is inserted into the computer is termed a frame. Each frame is further composed of groups, and groups are composed of lines of information.

FRAME HEADER
Comprised of the appropriate frame number, frame type, decision statement indicator, and label, if any. The terms Group 1 and Frame Header are synonymous regardless of frame type selected.

FRAME TYPES
The legal frame types are: question (Q), multiple-choice (M), and decision (D).

This command instructs the program to retrieve the requested lesson from disk.

?GOTO (Frame label, frame number, or *)
An option available in the Lesson Execution Mode which permits the trainee or the course designer (in a pseudo-trainee role) to branch unconditionally to the frame label indicated.

GQ
A relational operator used in a decision statement meaning "is greater than or equal to."

GR
A relational operator used in a decision statement meaning "is greater than."

See FRAME HEADER.

IF
A connective used to mark the beginning of a decision statement.

ITEM
A single unit to which a value may be assigned. The value can be increased or decreased or the unit can be examined (tested) to determine its current value. Based on the value, additional control of the trainee's path through a sequence of frames can be established. There are 64 (0-63) items within the CDTS.

KEYWORD OFF
Causes the program to make an exact character-for-character (word by word) comparison to determine if the trainee's response matches a course designer's answer. No extraneous words may appear in the trainee's response for a match to occur. The correct response is that which matches the answer associated with the (+) sign. Keyword off is in effect unless specifically changed by using the KEYWORD ON command.

KEYWORD ON
A service function which causes the program to match a trainee's response to that which the course designer has specified as correct by disregarding any extraneous words in the trainee's response.

LAST
A condition used in a decision statement to establish a conditional statement based upon the trainee's final response for a specific frame.

LETTERS (A-Z)
Characters used as tags for answers. The tag associated with the correct answer is saved with the Trainee Record.

LQ
A relational operator used in a decision statement meaning "is less than or equal to."

LS
A relational operator used in a decision statement meaning "is less than."

M
Code for multiple-choice frame.

MULTIPLE-CHOICE FRAME - M
This frame is normally used to present course content material and alternate answer choices. The inserted answers are printed for selection.

NONE
Used as a relational operator in a decision statement to indicate the condition specified did not occur.

Operative
A secondary command which occurs in a legal expression and must follow a primary command. The operative cannot be inserted separately and may or may not be an optional entry.

OR
A connective used in a decision statement to mean either or both.

ORDER OFF
Causes the program to treat the trainee's response as a standard English statement. The words in the trainee's response must be in the same order as specified by the course designer. This is the default setting unless specifically changed by the ORDER ON command.

ORDER ON
A service function which causes the program to insure that the word(s) specified by the course designer appear somewhere in the trainee's constructed response--independent of order. This capability operates only during the execution of the particular frame in which specified and is restricted to use in a question (Q) frame.

PHONETIC OFF
Permits the course designer to turn off the encoding capability.

PHONETIC ON
A service function which causes all words in both the course designer's answer(s) and the trainee's response(s) to be encoded by the program and the encoded messages compared for a possible match. This permits the misspelling

of words by the trainee. If a misspelled word is within the encoding criteria, the response is considered correct by the program. The function is restricted to use within the particular question (Q) frame in which it is specified.

PUNCTUATION
Commas, periods, question marks, and apostrophes are not to be included in a group 3 response unless they are parc of the syntax of a computer language.

Q
Code for question frame.

QUESTION FRAME - Q
This frame is normally used to present course content material, to inform, or to require a constructed response.

R: (REPEAT)
An action command which permits the course designer to specify a feedback message requiring another response. If none is specified the program prints "WRONG TRY AGAIN" and waits for another response.

REPEAT
See R: (Repeat)

RIGHT
A condition used in a decision statement to indicate correctness of response.

SET
An optional function which operates in conjunction with PHONETIC, KEYWORD, and ORDER. If these functions are "SET" on, they remain in that state until their status is changed by the course designer with a subsequent declaration. SET inhibits the program from automatically turning PHONETIC, KEYWORD, and ORDER off when advancing from one frame to another.

SET KEYWORD ON
See SET

SET ORDER ON
See SET

SET PHONETIC ON
See SET

UNANTICIPATED ANSWERS
These are responses entered by trainees for which the course designer has not constructed correct or incorrect answers to be matched in Group 3.

WRONG
A condition used in a decision statement to indicate an incorrect response.

SUGGESTED ISD AND CDT STANDARDS

ISD STANDARDS

1. Every lesson should begin with an INTRODUCTION. This should be motivating and be written in a personal, informal format. It should not be stilted.

2. If the lesson is the first lesson of a course, it should identify the contents of the course including all of the lesson titles and lesson numbers, and the total on-line and off-line times necessary to complete the course.

3. The introduction should contain the Lesson Title and Number, Lesson Content, Criterion Objective(s), any required manuals or other necessary materials, and the approximate time for the lesson.

4. Development of the objectives and test should be in accordance with AFM 50-2, Chapter 4.

   a. Analyze education/training requirements to develop objectives. An objective is a precise statement which answers the question: What must students do to show that they have learned what they were expected to learn? An objective should specify:

      (1) Behavior to be shown or performance to be done;

      (2) Conditions under which behavior is shown or performance is done;

      (3) Standards of performance or the proficiency expected (if there are no standards stated, then the implied standard is "without any errors").

   b. Tests should be developed which meet needs and measure the objectives. Some types of tests are:

      (1) Criterion-referenced test - evaluates individual students in relation to the standards of performance set in the objectives.

      (2) Diagnostic test - measures the supporting skills and knowledges that contribute to the student's ability to perform the objectives.

      (3) Survey test - designed to determine what a prospective student knows or can do before any instruction (not a bypass test).

5. Each lesson should have its own objective(s) and each objective should be tested, but not necessarily in the final test.

6. Give the number of questions and the pass/fail point for each test before starting the test.

7. Provide transitions within lessons between major subjects and from one lesson to the next within a course. Mention required manuals, etc.

EXHIBIT CDT010-5

8. Use questions as often as possible to make the student interact with the lesson. Do not pool questions unless it is a test or review quiz. (This does not mean that a question is required for every screen/frame.)

9. Use questions to relate to exhibits. Make the student react to the lesson and to what they are trying to learn.

10. Summaries should be placed in the lesson at major break points and at the end of the lesson.

11. Maintain information in the exhibits, manual, or the lesson on who to contact about the validity of questions or technical facts that are presented in the lesson.

CDT STANDARDS - GENERAL

1. A lesson should require between 30 and 90 minutes for the completion of a logical unit of instruction.

2. Longer lessons should contain progress reports at strategically located points to let the student know how much material remains in the lesson. It isn't necessary to mention time, but comments like "You are now halfway through the lesson" or "You have three short areas remaining to complete this lesson" would be fine.

3. A lesson should contain progressively more detailed loops to aid the slower student, but branching capabilities should be provided to permit the advanced student to bypass some explanations and reviews.

4. When showing information to be used in answering questions, associate the student with either a VIP or a teletype (if ITEM60=0 the student is on a TTY and if ITEM60=1 the student is on a VIP). NOTE: ITEM60 is set automatically by the CDTS executive.

5. Avoid "neat tricks" with CDTS since updates of the system could inadvertently cancel part of the trick.

6. Each lesson should employ Instructional System Development and Computer Directed Training techniques, follow CDTS programming rules and be grammatically accurate.

7. Use comments in your lesson (* in column 1 of the CDTS source code) for:

   a. complicated ITEM settings in group 4's.

   b. before decision frames.

   Remember, you may be responsible for maintaining a course that you did not write, so be as helpful as possible to the next person.

8. Each lesson should follow the SKELETON and contain pass and fail frames (use %:ITEM63=1 for PASS).

9. Group 3 responses of "Q" frames should not contain end punctuation such as commas, periods, question marks, and apostrophes unless they are required for a particular machine instruction. Apostrophes showing possession are always required. Also keep the extra spaces in your answer to a minimum.

10. Nasty remarks should never be included in a lesson. (How would you like to be insulted by a computer!)

11. Profanity and off-color jokes have no place in a lesson.

12. Line 1 of the lesson should contain the letters SEQ beginning in card column 15 followed by the original date in the form YYMMDD as shown in the SKELETON (Example: 810220).

13. The version date should be included in the first frame of a lesson. Enter the month of the next projected release for CDTS (i.e., 10 January 1983, 30 June 1983).

14. The source and object entries included in the comments at the beginning of the lesson must reflect the current location of the program as shown in the SKELETON.

15. Course exhibits are a part of the course, so responsibility for the course includes responsibility for the exhibits.

16. Avoid using the terminal as a page-turner for a textbook. The frequent interaction between the student and the computer, along with any necessary remedial training, will insure that the student is learning the required material.

CDT STANDARDS - LESSON ORGANIZATION

1. Start all lessons with an "L" in card column 1 of the first line of your program as shown in the SKELETON.

2. Introduce new topics and end old topics with good transitions.

3. Start new areas of instruction on a separate "page" with a heading. A good convention for headings is to surround major topics with two *'s (i.e., ** TOPIC **), and minor topics with one * (i.e., * TOPIC *). In longer lessons, use three *'s for major topics, two *'s for minor topics, and one * for subtopics. However, left justify the headings so time will not be wasted on the teletype.

4. Put a descriptive label on group 1 frames that introduce major topics and put them in the introductory label comment section of your program. These frames should be numbered 100, 200, 300, 400, 500, etc., as shown in the SKELETON.

5. Use descriptive functional labels (MOVE, GOTO) of no more than 6 characters for interior labels. Label major areas within a lesson (END, PASS, FAIL). Other areas should not be accessible to the student and therefore are usually not labeled.

6. When writing a new lesson, use the SKELETON file. Do this by bringing a copy of the SKELETON file into your current file. After signing on to TS1, enter the following information at the SYSTEM level:

```
CARD 0 CDTSCDTS/CDT/SKELETON,R
```

After the system responds READY, use the SAVE command to place the SKELETON into your file. For maintaining older lessons, reorganize them so the logic follows the top-down design of the SKELETON.

7. Lesson time is computed by dividing the number of lines in the lesson by 500-700. The exact value between 500 and 700 depends upon the difficulty of the material, number of "Q" frames, and amount of simulation. This time is shown in the introduction frame. Tell the student of the time, both on-line and off-line, needed to do the next lesson, and what supplementary material might be useful to complete that lesson. If necessary, update the time required for that lesson in the manual, the introduction of the lesson and in the course introduction.

CDT STANDARDS - TESTS

1. Show only one test question per screen.

2. Terminate final tests when student exceeds the maximum number of mistakes permitted as shown in the SKELETON.

3. As shown in the SKELETON, do not provide correct answers for questions incorrectly answered during the final test.

4. The standard pass/fail point is 70% on tests as shown in the SKELETON.

5. The use of pretests for specific areas is permissible, but do not use a bypass test for an entire lesson.

CDT STANDARDS - PAGING

1. Page at the beginning of the frame, as shown in the SKELETON, not at the end of the frame or just before a branch.

2. Limit the amount of information that is displayed at one time to avoid overwhelming the student, but be sure that there is sufficient information to answer the question asked.

3. Write the lesson to control paging rather than have the terminal control the paging.

4. Make certain that the student does not have to page twice to move to the next page. This can occur when there are too many lines on a screen.

5. Limit each screen to 14 lines of text (this includes responses to questions).

CDT STANDARDS - SPACING

1. Do not indent on the left margin, even when starting a new paragraph.

2. Indentation is used only for emphasis.

3. Single space all lines of text within a paragraph except for the purpose of emphasis.

4. Use three asterisks (***) before and after an item to show it has extra importance for the student (i.e., *** NOTE ***).

5. Provide two spaces after a period at the end of a sentence and also after a colon. Provide one space after a comma and a semicolon.

6. When enumerating in columnar format, use a number with a period, then two spaces. When using hyphens for parenthetical elements or in place of commas, use two hyphens with no spaces.

7. Leave at least one clear line between paragraphs by using the backslash (\) character.

8. Align the right margin as close as possible without excessive hyphenations. Do not hypenate a word at the right margin if it just has one or two letters to be carried to the next line, or if only one or two letters will be on the first line.

9. Avoid dressy or involved headings since they require so much time to reproduce on the teletype.

10. If indentation is required for clarity, use three spaces.

CDT STANDARDS - QUESTION, MULTIPLE CHOICE, AND DECISION FRAMES

1. Use questions frequently to keep the student involved.

2. Use only one question per frame (i.e., do not use R: to ask a different question).

3. Use a good mixture of "M" and "Q" type frames. (The subject matter you are teaching may influence your questioning technique.)

4. Multiple choice type questions are usually better than matching questions.

5. Insure that the stem, the question responses, and the student responses are displayed on the same page for VIP operation (approximately 14 lines). Use caution when formatting for the VIP.

6. In "M" frames, have three spaces between each letter designator and text of choice as shown in the SKELETON.

7. In "M" frames, have one blank line before the choices.

EXHIBIT CDT010-5 (Cont)

8. "Q" frames will normally have at least one retry for each incorrect answer (i.e., use a 4-R:, then a 4-F:). Try to give a valid hint for each retry.

9. Give scored announced and unannounced quizzes to prevent students from paging instead of answering questions.

10. Avoid more than two retries. Provide followup for each retry by explaining the question further or by providing more information.

11. Allow "T" for true, "F" for false, "Y" for yes, "N" for no, and system standard abbreviations when possible.

12. Do not flag decision frames.

13. Flagged frames should be used sparingly, but the use of two or more is needed somewhere within the lesson to insure that the student is indeed going through the lesson. These frames should be checked with a decision frame before the student reaches the test area. Check the SKELETON for the correct format.

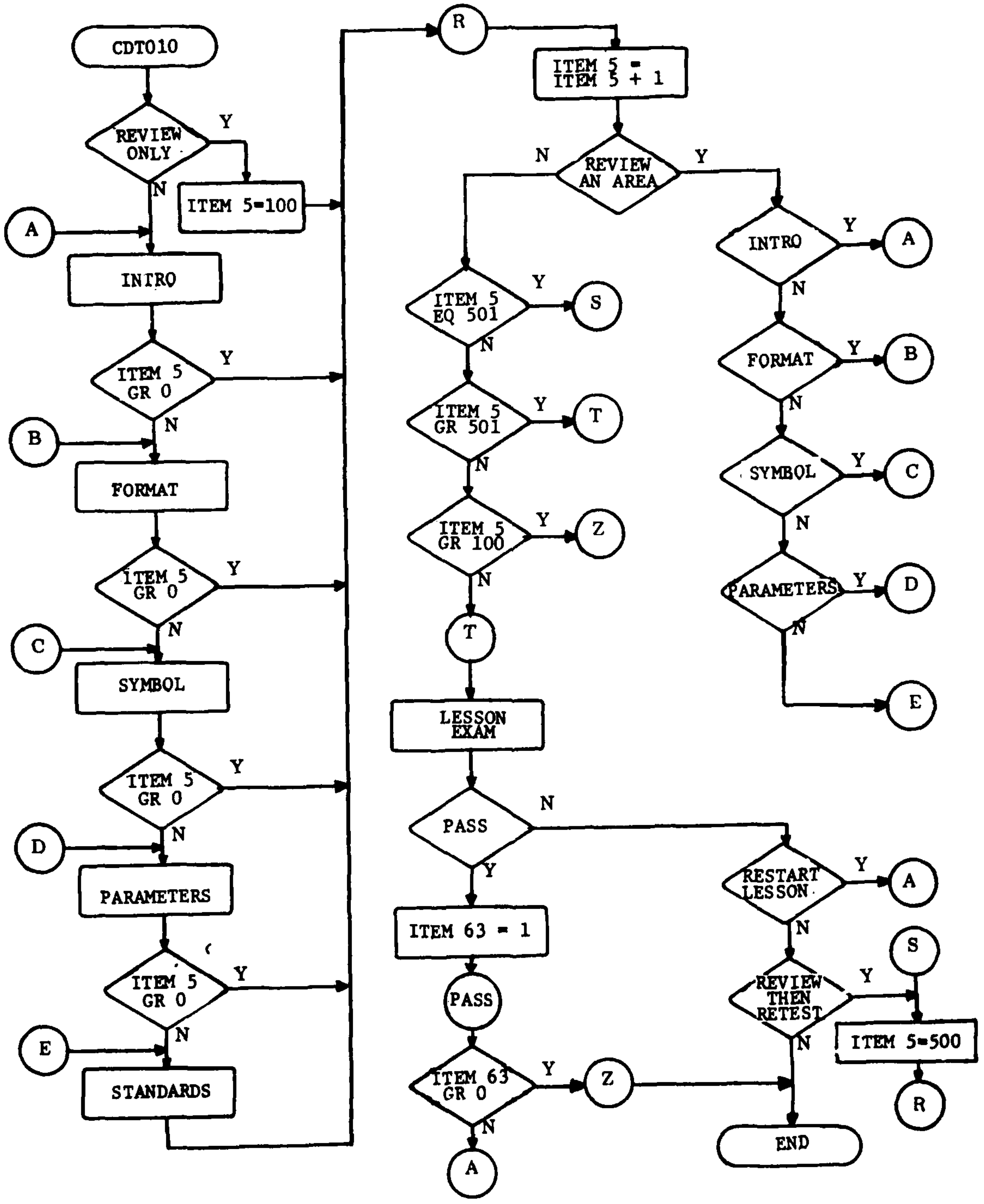
14. A frame number should be changed by .1 when the question has been changed. This allows the new question to be identified on listings from trace tapes. When writing new lessons, have an interval between the frame numbers.

15. Always vary the number of the review frame. It should be slightly before or slightly after 800 (i.e., 798.9, 799.6, 800.8, 801.4, etc.).

16. Give all of the information to the student before quizzing on that information. Also, minimize the use of scored leadoff questions.

17. Minimize the use of "canned feedback" (C:) as shown in the SKELETON.

18. If a project (on-line or off-line) is required, ask some questions concerning the completed project to insure that the work is being done.



CDT010 LESSON FLOW CHART

EXHIBIT CDT010-6

CODING SHEET

PROGRAMMER JOHN SMITH

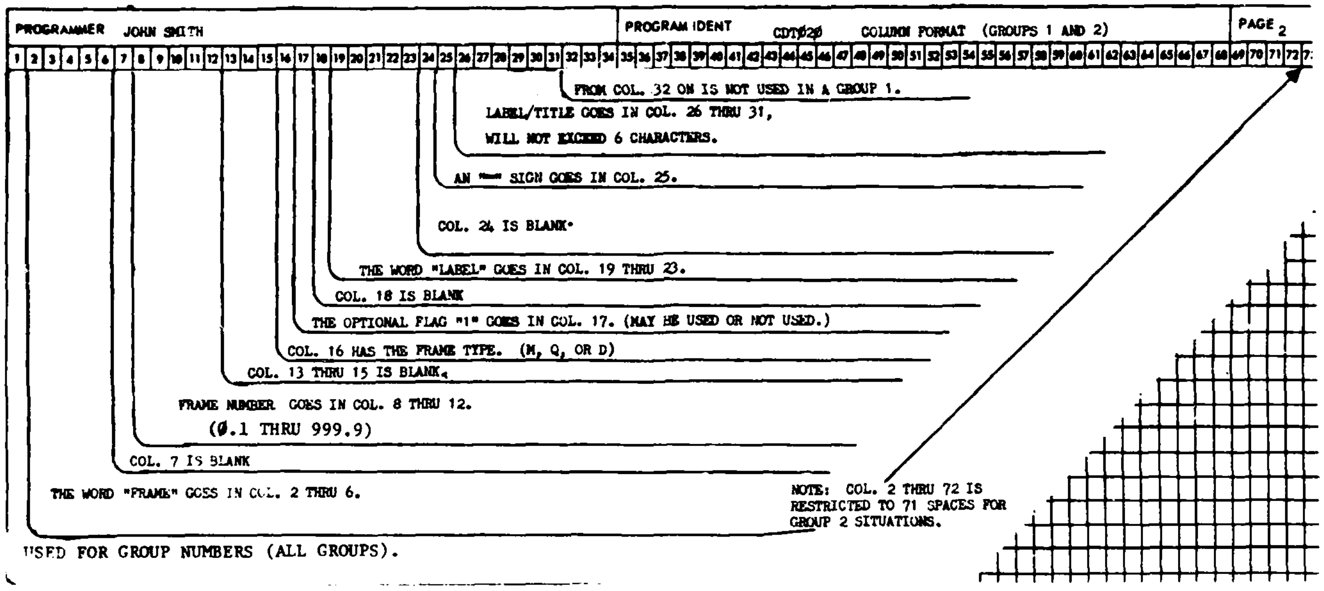
PROGRAM IDENT CDTØ2Ø MULTIPLE CHOICE FRAME

PAGE 1

```
1FRAME 001      M  LABEL - ALTER
2   THIS TYPE OF FRAME PRESENTS INFORMATION TO THE STUDENT AND GIVES HIM
 ALTERNATE SELECTIONS FROM WHICH TO CHOOSE.  WHAT TYPE OF FRAME IS THIS?
3A QUESTIUN
 B+MULTIPLE CHOICE
 C DECISION
4B F:RIGHT.
 A F:WRONG, THE ANSWER IS B.
 C F:WRONG, THE ANSWER IS B.
```

EXHIBIT CDT020-1

CODING SHEET



FROM COL. 32 ON IS NOT USED IN A GROUP 1.

LABEL/TITLE GOES IN COL. 26 THRU 31,
WILL NOT EXCEED 6 CHARACTERS.

AN "—" SIGN GOES IN COL. 25.

COL. 24 IS BLANK.

THE WORD "LABEL" GOES IN COL. 19 THRU 23.

COL. 18 IS BLANK

THE OPTIONAL FLAG "1" GOES IN COL. 17. (MAY BE USED OR NOT USED.)

COL. 16 HAS THE FRAME TYPE. (M, Q, OR D)

COL. 13 THRU 15 IS BLANK.

FRAME NUMBER GOES IN COL. 8 THRU 12.
(Ø.1 THRU 999.9)

COL. 7 IS BLANK

THE WORD "FRAME" GOES IN COL. 2 THRU 6.

USED FOR GROUP NUMBERS (ALL GROUPS).

NOTE: COL. 2 THRU 72 IS RESTRICTED TO 71 SPACES FOR GROUP 2 SITUATIONS.

EXHIBIT CDT020-2

CODING SHEET

PROGRAMMER JOHN SMITH	PROGRAM IDENT CDTØ2Ø	ACTION COMMANDS (GROUP 4)	PAGE 3

```
         1         2         3         4         5         6         7
123456789012345678901234567890123456789012345678901234567890123456789012
1FRAME 002     M
2   WHICH ACTION COMMAND WOULD BE ENTERED BY A COURSE WRITER FOR THE
 VERBAGE "WRONG, TRY AGAIN. "?
3B  B:
 C  C:
 R+ R:
4R  F:VERY GOOD.
 B  R:
 C  R:

1FRAME 003     M
2   WHICH ACTION COMMAND WOULD BE ENTERED FOR THE VERBAGE "THE CORRECT
 ANSWER IS ..."?
3B  B:
 C +C:
 K  R:
4C  F:YOU ARE RIGHT.
 B  R:
 K  R:
```

EXHIBIT CDT020- 3

GUIDELINES FOR THE OFF-LINE PROJECT

1. Use a coding sheet furnished by your Training Systems Manager or a suitable substitute as shown in AFM 50-752, exhibit CDT020-1.

2. Code your job task on these coding sheets in CDTS format.

3. Use multiple choice frames for Part I of the job task and question frames for Part II. Use as many frames as you like.

4. Do NOT complete Part II of the task until lesson CDT030 has been completed.

5. Start your frame numbering at 010 with multiples of 10, being sure to keep them in sequence.

6. Label your job task "TASK". Later it will become your TASK FILE.

7. Take the Job Task you completed from lessons CDT020 and CDT030 and insert any type of decision frames that you can utilize after completion of lesson CDT040.

8. You must leave the lesson in order to complete each part of the task.

EXHIBIT CDT020-4

JOB TASK

TASK: COMPOSE A TRIVIA QUIZ FOR USE ON CDT.

Part I: Steps 1 through 4 (From CDT020).

1. Choose a topic of interest that you are fairly familiar with. The topic should allow you to write both "Q" and "M" frame questions.

2. Check your knowledge of the subject with other sources in order to insure accuracy.

3. In order to understand the types of quizzes we are looking for, you should look at trivia quizzes written by other CDT students. Several topics are contained in SUL000, which you can take on CDTS.

4. Get a copy of the skeleton lesson so that you can practice writing questions in CDT format. To do this, BPRINT file CDTSCDTS/CDT/SKELETON.

Part II: Steps 5 and 6 (from CDT030).

5. Based on your knowledge of your chosen subject, write questions for "Q" type frames in CDT format.

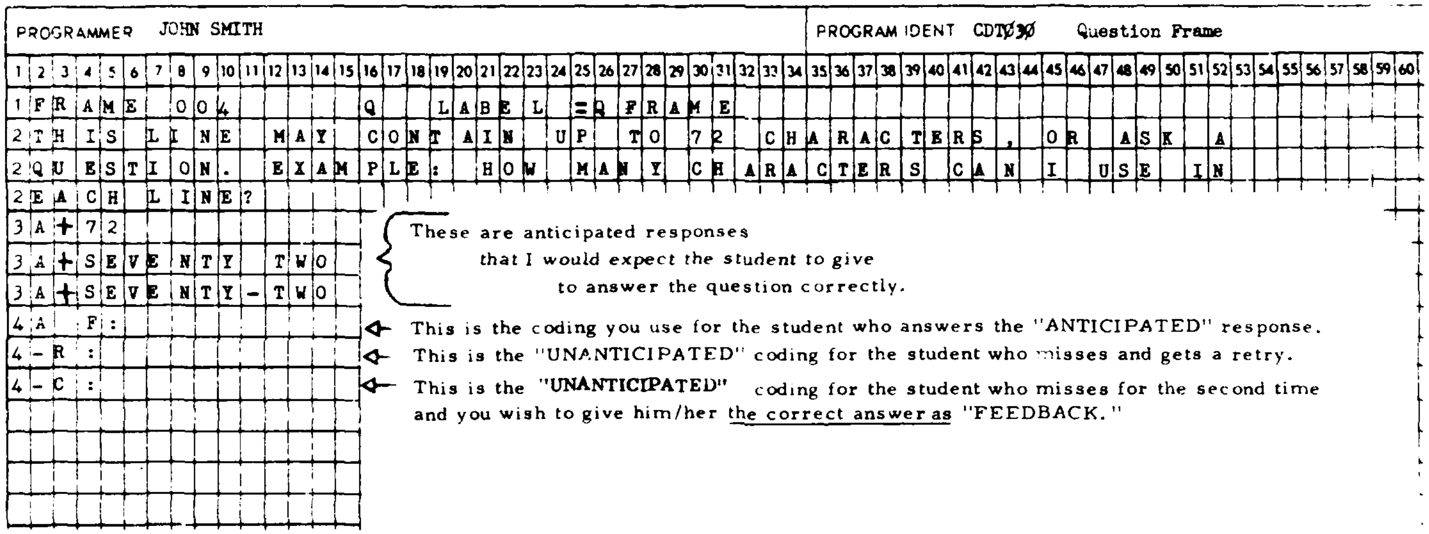
6. Review TUC040, if necessary. TUC040 tells you how to enter your project into time-sharing and create a file.

EXHIBIT CDT020-5

CDTØ3Ø GROUP 4 FORMAT (Q FRAME)

```
1234567890123456789012345678901234567
4A F:        These actions commands are LEGAL in a
4A-R:        Group 4 situation for a "⌄" frame.
4A- F:
4-F: B:

                      EXAMPLE A

4%:ITEM5=1                  Set ITEM5 EQ to 1
4-F: %:ITEM2=1              Feedback and set ITEM2 EQ to 1
4-F:WRONG. %:ITEM5=ITEM5+1  Incrementing
4-F:RIGHT. %:ITEM6=ITEM6-1  Decrementing
4%:ITEM5=P                  Print ITEM5

                      EXAMPLE B
```

EXHIBIT CDT030-1



PROGRAMMER JOHN SMITH PROGRAM IDENT CDTØ3Ø Question Frame

```
1FRAME 004     Q  LABEL =QFRAME
2THIS LINE MAY CONTAIN UP TO 72 CHARACTERS, OR ASK A
2QUESTION. EXAMPLE: HOW MANY CHARACTERS CAN I USE IN
2EACH LINE?
3A+72
3A+SEVENTY TWO
3A+SEVENTY-TWO
4A F:
4-R :
4-C :
```

These are anticipated responses that I would expect the student to give to answer the question correctly.

This is the coding you use for the student who answers the "ANTICIPATED" response.

This is the "UNANTICIPATED" coding for the student who misses and gets a retry.

This is the "UNANTICIPATED" coding for the student who misses for the second time and you wish to give him/her the correct answer as "FEEDBACK."

EXHIBIT CDTO 30-2

DECISION FRAME CONVENTIONS

1. LOGICAL CONNECTIVES - Establishes the type of decision statement.

IF The first word in a decision statement and asks whether a condition was met.

AND Used to put more than one condition in a decision statement. To be true, all conditions must be met.

OR Test to see if at least one condition has been met.

ELSE Used to account for all unspecified conditions following an IF, AND, OR statement. If the previous conditions in the decision statement were not met, the else condition will be met and executed.

END This marks the end of a group of action commands under the control of an IF statement and separates a conditional statement from an unconditional statement.

2. CONDITIONALS - Specifies under what conditions the performance of the student will meet the criteria of the D Frame.

NONE The condition specified never occurred.

ALL Includes all responses and indicates that the condition was satisfied as specified.

RIGHT Indicates that a right response criteria is to be met.

WRONG Indicates that a wrong response criteria is to be met.

SEEN Indicates that a frame or group of frames in a specified sequence within a lesson have been accessed.

EVER Indicates the condition specified occurred at least once in the progression through the lesson.

LAST Establishes a conditional statement based upon a final response for a specified frame.

EXHIBIT CDT040-1

3. RELATIONAL OPERATORS - Establishes minimum and maximum performance criteria.

LS	is less than
LQ	is less than or equal to
EQ	is equal to
GR	is greater than
GQ	is greater than or equal to

RULES OF THE D FRAME

More than one decision statement can be used in the same frame.

All lines in a D frame must start with IF, AND, OR, ELSE, END, F:, B:, or %:. This is where conditional statements, feedback, item testing/setting, and branching may be accomplished. The action commands C: and R: cannot be used in the D frame.

More than one conditional may be used in the same statement.

The hyphen (-) used between referenced frame numbers (Example: 205-210) represents multiple frames.

ITEMS are automatically set to 0 at the beginning of each lesson by the system.

ITEMS are set, incremented, and decremented in group 4 of the M and Q frames and group 2 of the D frame.

An ITEM can be set to itself plus a value. Use the format: ITEMX=ITEMX+D (%:ITEM5=ITEM5+2).

ITEMS can be set to another item plus a value. Use the format: ITEMX=ITEMY+D (%:ITEM5=ITEM10+5).

ITEMS can be set, etc., in conjunction with other action commands or alone.

ITEMS can be set before any action has been evaluated by placing the command at the beginning of the group.

More than one frame can be evaluated by letter value only. (EXAMPLE: IF 11,a 13,b B:16)

Several frames can be evaluated by making decisions concerning the + or - indicators. (EXAMPLE: IF +7,+9,+11, F:You need help. Let's review. B: 26)

Frames can also be evaluated by a combination of + or - and letter responses. (EXAMPLE: IF 5a,+7,-12, F: Great! B:18)

The D frame can be used to evaluate only the last 250 flagged frame numbers.

EXHIBIT CDT040-2

SIMULATION INSTRUCTIONS FOR CARDIN AND BPRINT

This exhibit is designed for use with the simulation project in lesson CDT050 (simulation will start at system level). Follow the instructions carefully for successful completion of this project.

```
SYSTEM        RESPONSE

*             CARDIN or CARD

OLD OR NEW    NEW

READY         AUTOX 20,20

*0020         #1FRAME 1       Q  LABEL =CARDIN
```

a. Sequence numbers must be on Automatic.

b. Information must be placed in card position 1.

c. Starting sequence number will be *0020.

d. Each card image file will be incremented by 20 (0020, 0040, 0060, etc.).

```
*0040  #2////

*0060  #2The CARDIN subsystem allows the user to create card image files.

*0080  #2Files with records having up to 80 characters can be created.

*0100  #2Files with records exceeding 80 characters cannot be created.

*0120  #2The character limitation when using CARDIN is 80 characters.

*0140  #2(Enter True or False)

*0160  #3A+True

*0180  #3B False

*0200  #4A F:Correct.
```

EXHIBIT CDT050-1

```
*0220  #4B F: Wrong.

*0240  #4-R:Enter true or false, please.

*0260  #4-F:Wrong, the statement is true.
```

NOTE: Each of the above lines represents a single card image file. After the entire frame has been entered, "SAVE" the file before requesting your BPRINT listing.

```
*0280          (hit null return)

*              SAVE CARDIN

DATA SAVED-CARDIN

*              BPRINT CARDIN

$IDENT         ABC DEFG114

LABELS ?       MOVE

TAB CHARACTERS AND SETTING?  (hit null return)

SNUMB 12347    DONE

*              BYE
```

EXHIBIT CDT050-1 (Cont)

CDTS OFF-LINE INSTRUCTIONS

1. This exhibit is to be used when accomplishing the Part II project described in lesson CDT050.

2. The project may be accomplished on the teletype (TTY) or a visual information projection device (VIP).

3. Step-by-step procedures for the operation of the TTY or VIP can be found in lessons TUC010, TUC011, or TUC012, respectively.

4. Task

   a. Using the lesson that was coded in CDT format in lessons CDT020, CDT030, and CDT040, enter the lesson into a temporary file through the CARDIN subsystem. This lesson will then be saved to a permanent file and a BPRINT listing requested. The listing will be desk checked with all corrections to be made noted in ink. Reenter the CARDIN subsystem and make the necessary corrections to the lesson on file. Resave the lesson and request another listing. A corrected BPRINT listing will serve as as successful completion of Part 2.

   b. File Setup

      (1) Starting line number will be 0020.

      (2) Increment line numbers by 20.

      (3) LABEL LESSON "TASFIL."

      (4) Data must be placed in card column one (1).

      (5) Sequence numbers will be on automatic.

      (6) After all material has been entered, SAVE the lesson to a permanent file.

      (7) Request a BPRINT listing.

      (8) Desk check lesson.

      (9) Show listing to monitor.

      (10) Reenter CARDIN and make corrections.

      (11) Resave the lesson.

      (12) Request a corrected BPRINT listing.

      (13) Show this lesson to your monitor for credit of this course.

DECK SETUP FOR COMPILE

```
1      8      16

$      IDENT   (LOCAL)
$      USERID  (LOCAL)   (Note 1)
$      SELECT  (LOCAL)
$      EXECUTE
$      LIMITS  10,30K,6K,5000
$      DATA    CS
        .
        .      input source lesson
        .
$      SYSOUT  PC
$      FILE    FO, NULL
$      FILE    W1,,20
$      FILE    W2,,20
$      FILE    WR,,20R
$      ENDJOB
***EOF         (Note 1)
```

NOTE 1: This card may not be used in CARDIN.

EXHIBIT CDT060-1

CREATE LESSON FILE USING FILSYS JOB

```
1      8      16

$      IDENT   (LOCAL)
$      USERID  (LOCAL)                               (Note 1)
$      FILSYS
USERIDƀuserid$password                           (user supplied)
FCREATEƀcat/file string of lesson file,BLOCKS/n,n/,   (Note 2)
MODE/RAND/,FCLASS/UNC/
$      ENDJOB
***EOF
```

NOTE 1: This card may not be used in CARDIN.

NOTE 2: The lesson file size in blocks replaces <u>both</u> appearances of "n" on this card.

EXHIBIT CDT060-2

ACCESS DIALOGUE

NOTE: Your responses are underlined. The symbol Ⓡ denotes a carriage return. None of your responses may contain embedded blanks.

* ACCESS Ⓡ

FUNCTION? CF Ⓡ

CATALOG FILE STRING TO WORKING LEVEL?
(enter the full catalog string down to but not including the file name) Ⓡ

FILE NAME, SIZE, MAX SIZE? Ⓡ Where "n" is the file size in blocks

GENERAL PERMISSIONS? Ⓡ

SPECIFIC PERMISSION? Ⓡ

SUCCESSFUL. FILE NAME, SIZE, MAX SIZE Ⓡ

FUNCTION? Ⓡ

*

ACCESS SHORT FORM

SYSTEM? ACCESS Ⓡ

FUNCTION? CF,(full cat/file name),B/n,n,R/ Ⓡ

SUCCESSFUL.

Where "n" again is the lesson file size in blocks.

EXHIBIT CDT060-3

DECK SETUP FOR COMPILE AND LOAD

```
1      8      16

$      IDENT  Local
$      USERID  "             (Note 1)
$      SELECT  "
$      EXECUTE
$      LIMITS 10,30K,6K,5000
$      DATA   CS
        .
        .
        .
$      SYSOUT PC
$      FILE   FO,XIS,20
$      FILE   W1,,20
$      FILE   W2,,20
$      FILE   WR,,20R
$      SELECT LOCAL
$      EXECUTE
$      LIMITS ,18K
$      DATA   I*
       DISK   INIT
DELE   LSNXXX                     (Note 2)
LOAD   LSNXXX                     (Note 2)
$      FILE   FO,X1
$      PRMFL  FB,W,R,(cat/file name of lesson file)
***EOF
```

NOTE 1: This card may not be used in CARDIN submitted jobs.

NOTE 2: "LSNXXX" indicates the lesson name.

EXHIBIT CDT060-4

CDTS COMMANDS

Purpose: Before your course is released, you should take each lesson as though you were a student. It is suggested that you take as many possible paths through the lesson as time permits. To assist you in checking the lesson, here are some CDTS commands and how they are used.

?GET LSNXXX — (OLD or NEW) Gets a lesson from the lesson file and prepares it for execution. The "NEW" parameter (which may be shortened to just "N") indicates the lesson has not been started. If it has, CDTS will give you the following message with a question: "LESSON ALREADY STARTED, RESTART?" to which you reply "yes" or "no". The "OLD" is used to retrieve a lesson that has already been started previously.

?GOTO * — Starts the lesson at the location where the lesson pointer is (at the beginning for a new lesson, or where you left off for a previously started lesson).

?DONE — Closes out a lesson (sets a pointer indicating where you are in the lesson) and returns you to the "ENTER" command level.

?CLOSE — Closes out a lesson, like ?DONE command, and terminates CDTS by returning you to the system select level within time-sharing.

Other Versions:

?GOTO N — Where 'N' is any frame number in the lesson. This command may be entered at any point during execution of the lesson (as can all CDTS commands except ?LIST) and will branch you to that frame number and execute the lesson from that point.

?GOTO LABEL — Where 'LABEL' is any frame label within the lesson. The results of this command are analogous to those of the "?GOTO N" command for a frame number (N).

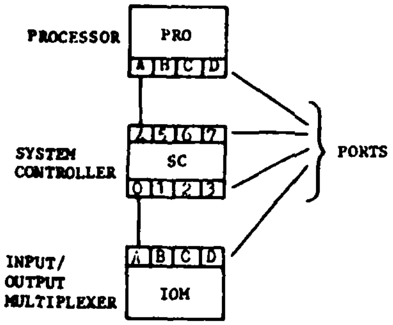
?NEW — Will take you back to the "lesson file" level to enable you to retrieve a different lesson file, but retain the same student file.

?LIST — Causes CDTS to display all lesson numbers available in the course specified, along with the lesson (if any) currently in the student's lesson file.

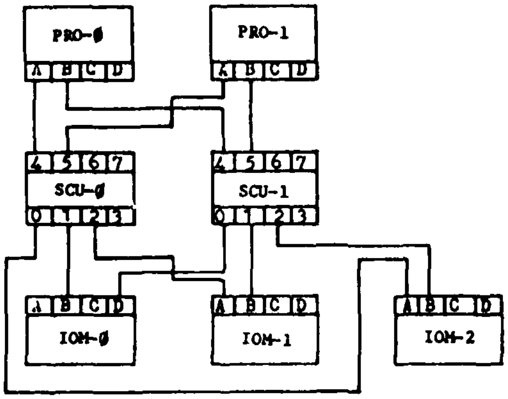
?PASS — Lists all lessons that have been completed successfully by the student.

EXHIBIT CDT060-5

GENERAL MACRO ASSEMBLY PROGRAMMING (GMP)
(H6000-CDT) COURSE EXHIBITS



HIS 6000 Mainframe, Minimum Configuration



HIS 6000 Functional Modularity

EXHIBIT GMP010-1

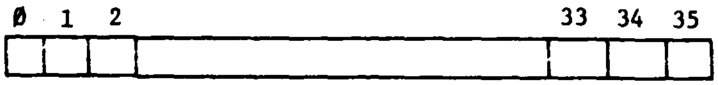


Figure 1. Machine Word

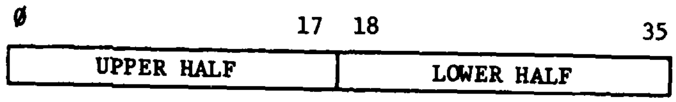


Figure 2. Half-Word Division

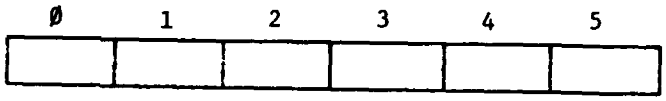


Figure 3. BCD Characters (6 Per Word)

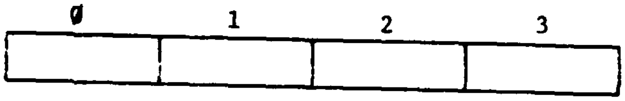


Figure 4. ASCII Characters (4 Per Word)

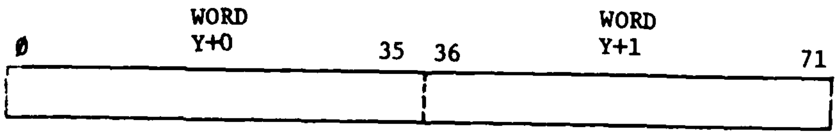


Figure 5. Double Precision
(Y-Pair)

EXHIBIT GMP040-1

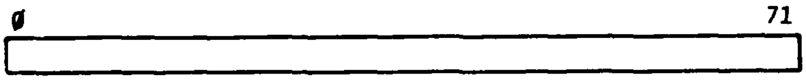


Figure 6. AQ-Register

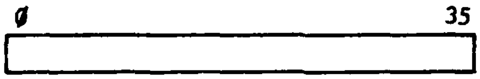


Figure 7. A-Register

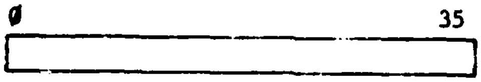


Figure 8. Q-Register

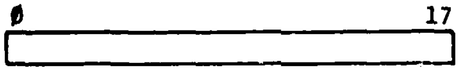


Figure 9. Index Register

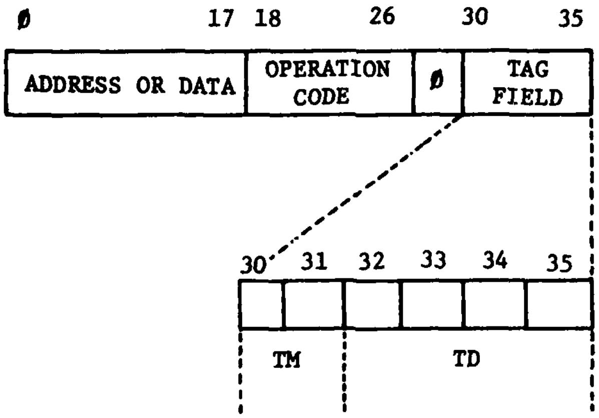
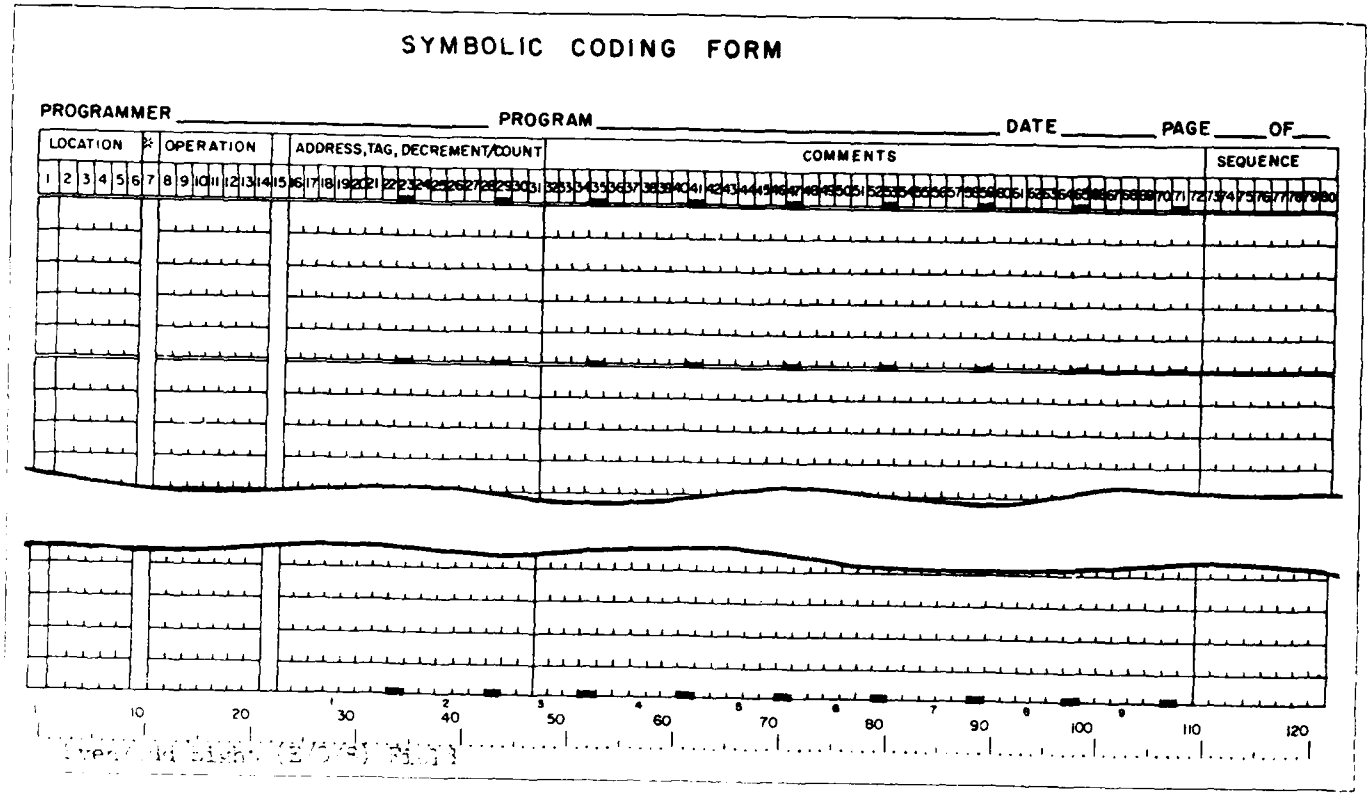


Figure 10. Instruction Word Format

EXHIBIT GMP040-2



SYMBOLIC CODING FORM

PROGRAMMER ____________ PROGRAM ____________ DATE ______ PAGE ____ OF ____

LOCATION					*	OPERATION								ADDRESS, TAG, DECREMENT/COUNT	COMMENTS	SEQUENCE
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	73 74 75 76 77 78 79 80

10 20 30 40 50 60 70 80 90 100 110 120

EXHIBIT GMP050-1

```
(1)  (2)                   (3)      (4)        (5)       (6)
$$   6817T ENTERED ATC620 AT 10.091 FROM T/S     0-08-03

(7)   (8)      (9)                                                        (10)
0001  $        SNUMB   6817T
0002  $        COMMENT MJRG              TSS CARDIN
0003  $$       USERID  MJRG$/ZZZ
0004  $        IDENT   FF700 0J/31,3-7/COTS ,          DUIF AND ASSE#6.3.2    00000010
0005 A$        GMAP    NDECK,NXEC                                             00000020
0006 A$        EXECUTE DUMP                                                   00000170
0007  $        ENDJOB                   (11)                                  00000180
  TOTAL CARD COUNT THIS JOB = 000021

      (12)        (13)      (14)       (15)          (16)
* ACTY-01   $CARD #0005   GMAP  (18) 10/11/78 (19) S#=211040000000
  * NORMAL TERMINATION (17)  AT 002440 I=5020 S#=211040000000 (20)

   (21)                                   (23)
     START 10.093     LINES     51 (22)  PROC  0.0001         I/O   0.001 (24)  IU  5 (25) MEMORY   24K (26)
 (27)STOP  10.095     LIMIT  10000 (28)  LIMIT 0.0400 (29)    LIMIT        (30) CU  5 (31) M*T      170 (32)
 (33)SWAP  0.000   (35)(36)(37)   (38)     (39)      (40)       (41)   (42)    (43) (44)
 (34)LAPSE 0.002     FC D TYPE      BUSY     IP/AT     FP/IT     IS/#C  MS/#E   ADDRESS T#

                     G* R D181 *     25        0         1          1      1    0-08-06
                     P*  SYJOT
                     R*  SYJOT
                     C*  SYJOT
                     *1 R D131 *     53        0         0         48     48    0-08-05
                     B* S D181 *     34        0         1         24     24    0-0d-09
     (45)
     LIST       31 LINES

   (46)
* ACTY-02   $CARD #0006   GELOAD   10/11/78   S#=400000000000
  * USERS OK MME GEBORT     AT 037775 I=0020 S#=400000000000

     START 10.095     LINES     64       PROC  0.0002         I/O   0.001       IU  5  MEMORY   16K
     STOP  10.096     LIMIT   5000       LIMIT 0.0500         LIMIT             CU  5  M*T       71
     SWAP  0.000
     LAPSE 0.001     FC D TYPE      BUSY     IP/AT     FP/IT     IS/#C  MS/#E   ADDRESS T#

                     G* R D181 *     23        1         1         24     24    0-08-09
                     R* R D181 *      9        0         0          1      1    0-08-06
                     P*  SYJOT
                     L* R D181 *    152        0         0        800   800K    0-08-07
                     *L R D181 *     33        0         0        250   250K    0-08-08

     LIST       64 LINES
  SNUMB = 68201, ACTIVITY # = 01, REPORT CODE = 74, RECORD COUNT = 000051 (47)
```

EXHIBIT GMP100-1

Definition of Fields in the Execution Report

Index Definition

1 - Report code for the execution report. Indicates that the execution report will follow.

2 - SNUMB

3 - System identification contains six alphanumeric characters. Can contain the command and software version (e.g., ATC400). First digit - major release from Honeywell; second digit - sub-release; and third digit - WWMCCS change.

4 - Time in hours and thousandths of hours.

5 - Input source (card reader, tape, remote, etc.).

6 - IOM, channel (PUB) number, and device number.

7 - Sequence number of control card images.

8 - A $ identifies the activity defining control cards. A $$ indicates that information has been omitted from the printout for security reasons (e.g., the password for the $ USERID control card). $$ also indicates that the file system software had to be called to process the control card. An asterisk (*) following the sequence number indicates that the control card resided on a permanent file (e.g., *A$).

9 - Control card name.

10 - Columns 73 through 80 of the control cards.

11 - Total card count for this job.

12 - Activity numoer within the job.

13 - Sequence number of activity defining control card.

14 - Activity type (FORTRAN, GMAP, COBOL, GELOAD, etc.).

15 - Month/Day/Year.

16 - Status of the program switch word at the beginning of the activity. See Exhibit GMP100-2 for further explanation of the program switch word.

17 - Abort message texts and/or reason codes appear in place of the NORMAL TERMINATION Portion of the termination message.

18 - Memory location at activity termination.

EXHIBIT GMP100-1 (Cont)

Index Definition

19 - Contents of processor indicator register at termination of the activity. Format of the indicator register may be found in the GMAP manual.

20 - Status of program switch word at the end of the activity.

21 - Time activity started in hours and thousandths of an hour.

22 - Total number of printed lines and cards punched for the activity (includes system output).

23 - Processor time consumed by activity in hours and ten thousandths of an hour.

24 - I/O time for this activity in hours and thousandths of an hour. (Does not include print or punch time.)

25 - Urgency level initially assigned to the activity.

26 - Core allocated to this activity (not including slave service area).

27 - Time activity stopped in hours and thousandths of an hour.

28 - Limit on number of lines of output for the activity. Does not include the number of lines output for the system.

29 - Limit on processor time for this activity in hours and ten thousandths of an hour.

30 - Limit on I/O time for the activity in hours and thousandths. If this limit is not specified on the $ LIMITS card, the I/O time is not limited and this field is blank on the execution report.

31 - Current urgency is the urgency attained by the activity before being allocated peripheral and core requirements. Urgency level is in^remented regularly as an activity waits for core space; length of wait is directly related to amount of core required.

32 - Core usage is equal to the number of 1024-word blocks used multiplied by the time for which they were used. Expressed in block-seconds. Used for charging time and core used to the user (the algorithm for this computation is complex).

33 - Total time activity was swapped out of core storage.

34 - Total elapsed time for activity in hours and thousandths of an hour. Equal to STOP time minus START time.

Index Definition

35 - File codes used in the activity.

36 - Disposition Code:

R - Release
S - Save
D - Dismount
C - Continue
P - Release and purge

37 - Media type according to the following list. A P or * to the right of the device type indicates a System Input created data file (*) or a cataloged permanent file (P).

Media Type	Device
D180	DSS180 Removable Disk
D181	DSS181 Removable Disk
D190	DSS190 Removable Disk
TAPE	7-track Magnetic Tape
TAP9	9-track Magnetic Tape
READ	Card Reader
PNCH	Card Punch
PRT3	PRT300 Printer
P203	PRT203 Printer
P303	PRT303 Printer
PRNT	PRT201 Printer
PPT	Paper Tape Reader
TYPE	Console Typewriter
SCC	System Control Center
CSS	Communication Subsystem

38 - Amount of time the peripheral device channel was used in the activity. If less than or equal to 999999, time is in milliseconds. If greater than 999999, time is in hours.

39 - Initial file position at start of activity. This field contains the alternate tape unit address when an alternate tape unit is specified on the $ TAPE card. Format is R/FF for tape files where R is number of records passed; F is number of files passed. If format is NNN, it represents the number of 320-word blocks passed for disk/drum files.

40 - Final file position at end of activity. Expressed same as 39.

41 - Initial size and mode of disk file (R-random, L-linked) or if tape file, number of times the file was accessed. Size is in 320-word blocks.

42 - Final size and mode of disk file, or if tape file, number of errors encountered. Size is in 320-word blocks.

EXHIBIT GMP100-1 (Cont)

Index Definition

43 - IOM number/channel number/device number (or disk pack number).

44 - Link number (if mass storage) or tape serial number. If the device is a fixed disk pack, the initial link number will be shown in the last column under L#/T#. If a removable disk pack is specified, the disk pack link number will be shown. (Later releases do not reflect link numbers.)

45 - Lines printed for the activity.

- Compressed deck cards punched (COMDK).

- Object cards punched (DECK).

- May indicate a second report with a code of XX has been requested by a $ REPORT control card (RC-XX).

46 - One of the following messages may appear in this space:

1. ***FILE XX IS DEFECTIVE*** (Part of file, number XX, could not be read because of defective space.)

2. Name of the activity deleted, if the activity was deleted.

47 - This line advises that a certain number of lines (record count) of output follow and that they are assigned a certain report code by the system software. For example, everything printed following the report code $$ (the execution report) was assigned the report code $$. A new report code (74) designates that another report will follow.

PROGRAM SWITCH WORD

0 4	5	6 17	18 35
System Options	E x e c	System Options	User Switches

The 36 bits of the program switch word are broken up as shown above.

System Options - Bits 0-4 and 6-17

Bits are set only when activity is being executed. Associated bit is set to 1 when system option is being used. Set an allocation of each activity. Bits 0-2 and 6-11 are set after encountering language processor or $ EXECUTE control cards. System options with associated bits are:

Bit	Bit=1	Bit=0
0	DUMP Option	NDUMP Option
1	FORTRAN, GMAP or PL/I	Neither FORTRAN, GMAP, nor PL/I
2	SYMTAB/XREF	NSYMTAB/NXREF
3	PURGE implicit file	NPURGE
4	CLEAR Memory	NCLEAR
5	Execute activity follows	No execution activity follows
6	COMDK/ON1	NCOMDK
7	DECK/ON2	NDECK
8	LSTOU/ON3	NLSTOU
9	UPDATE/ON4	No UPDATE
10	LSTIN/ON5	NLSTIN
11	DEBUG,NGMAC, STAB/ON6	NSTAB,GMAC,NDEBUG
12	(Bit 1=0) Abort Subactivity	(Bit 1=0) No Abort Subactivity
	(Bit 1=1) MAP	(Bit 1=1) NOMAP
13	GMAP/System Editor Interface	No GMAP Interface
14	(Bit 1=0) COPY/FORTRAN or COBOL created G* file	(Bit 1=0) NCOPY/Standard G*
	(Bit 1=1) FORM	(Bit 1=1) NFORM
15	(Bit 1=0) GESAVE	(Bit 1=0) No GESAVE
	(Bit 1=1) LNO	(Bit 1=0) NLNO
16	(Bit 1=0) EXTEND	(Bit 1=0) No EXTEND
	(Bit 1=1) BCD	(Bit 1=1) ASCII
17	(Bit 1=0) Reserved	(Bit 1=0) Reserved
	(Bit 1=1) OPTZ	(Bit 1=1) NOPTZ

Execution - Bit 5

If Bit 5=1, an execution activity follows. If Bit 5=0, no execution activity follows.

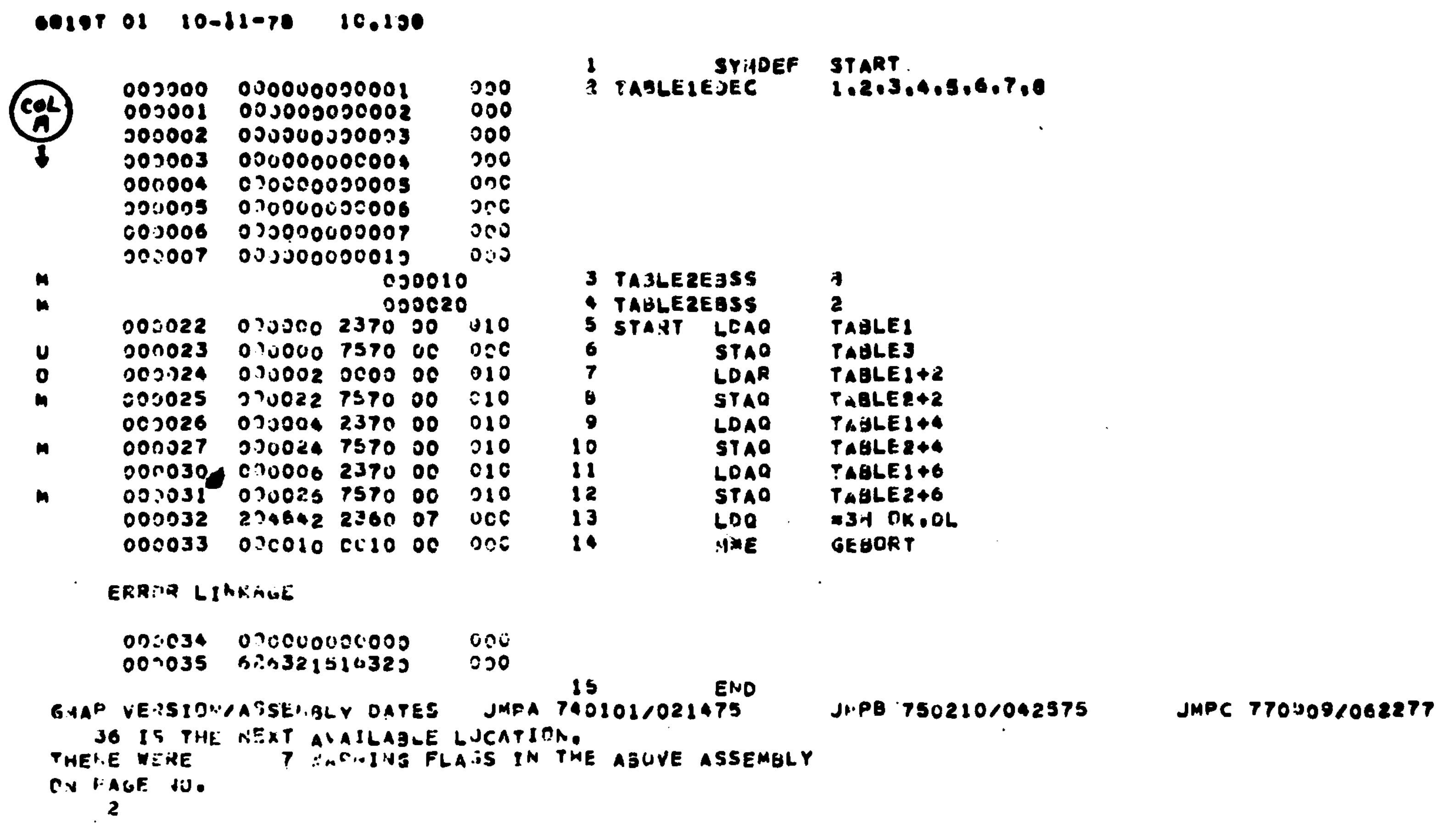
User Switches - Bits 18-35

These are switches which may be set by the programmer whenever needed in a job; they are retained between activities.

EXHIBIT GMP100-2

```
0822T 01  10-11-78   10.126
                                              ②
      ①       .    ③                         1           SYMDEF   START
     000000  000000000001    000              2 TABLE1   DEC      1,2,3,4,5,6,7,8
     000001  000000000002    000
     000002  000000000003    000
     000003  000000000004    000
     000004  000000000005    000
     000005  000000000006    000
     000006  000000000007    000
     000007  000000000010    000
                     000010                   3 TABLE2   BSS      8
     000020  000000 2370 00  010              4 START    LDAQ     TABLE1
     000021  000010 7570 00  010              5          STAQ     TABLE2
     000022  000002 2370 00  010              6          LDAQ     TABLE1+2
     000023  000012 7570 00  010              7          STAQ     TABLE2+2
     000024  000004 2370 00  010              8          LDAQ     TABLE1+4
     000025  000014 7570 00  010              9          STAQ     TABLE2+4
     000026  000006 2370 00  010             10          LDAQ     TABLE1+6
     000027  000016 7570 00  010             11          STAQ     TABLE2+6
     000030  204642 2360 07  000             12          LDQ      =34 OK,DL
     000031  000010 0010 00  000             13          MME      GEBORT

   ENTRY LINKAGE

     000032  000000000000    000
     000033  626321516320    000
                                             14          END
 GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475      JMPB 750210/042575      JMPC 770909/062277
    34 IS THE NEXT AVAILABLE LOCATION.
 THERE WERE     NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP100-3

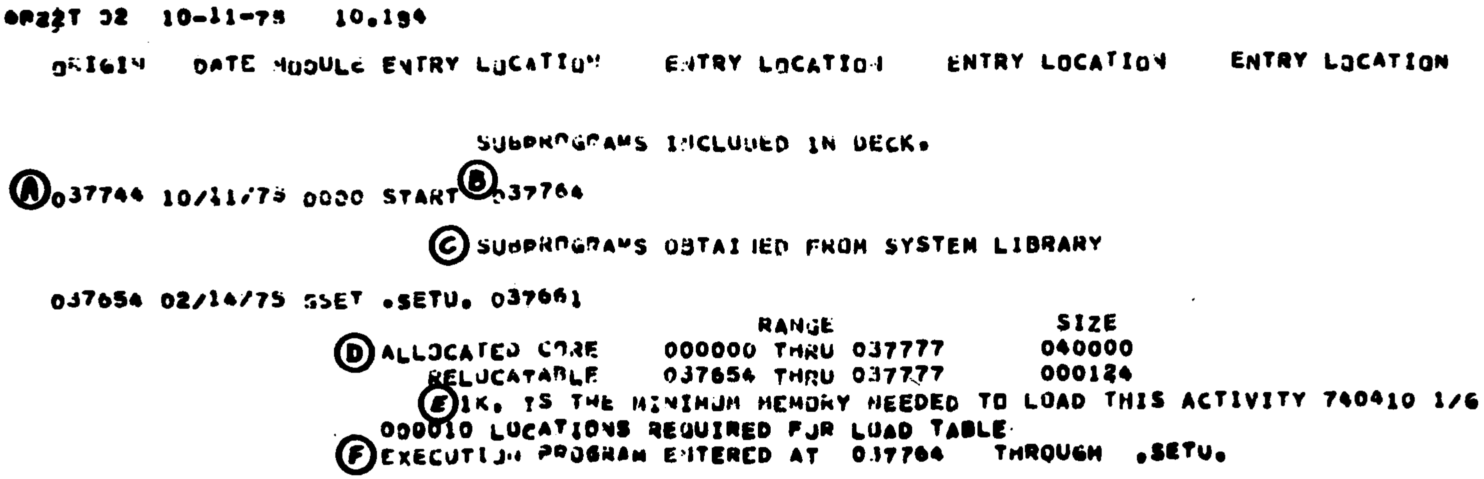


```
60197 01  10-11-78   10.130

                                                    1         SYMDEF   START
COL   000000  000000000001      000             2 TABLE1EDEC      1,2,3,4,5,6,7,8
 A    000001  000000000002      000
      000002  000000000003      000
      000003  000000000004      000
      000004  000000000005      000
      000005  000000000006      000
      000006  000000000007      000
      000007  000000000010      000
M                       000010                  3 TABLE2EBSS      8
M                       000020                  4 TABLE2EBSS      2
      000022  000000 2370 00    010             5 START   LDAQ    TABLE1
U     000023  000000 7570 00    000             6         STAQ    TABLE3
O     000024  000002 0000 00    010             7         LDAR    TABLE1+2
M     000025  000022 7570 00    010             8         STAQ    TABLE2+2
      000026  000004 2370 00    010             9         LDAQ    TABLE1+4
M     000027  000024 7570 00    010            10         STAQ    TABLE2+4
      000030  000006 2370 00    010            11         LDAQ    TABLE1+6
M     000031  000026 7570 00    010            12         STAQ    TABLE2+6
      000032  204642 2360 07    000            13         LDQ     =3H OK,DL
      000033  000010 0010 00    000            14         MME     GEBORT

    ERROR LINKAGE

      000034  000000000000      000
      000035  626321516325      000
                                               15         END
 GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475     JMPB 750210/042575      JMPC 770909/062277
    36 IS THE NEXT AVAILABLE LOCATION.
 THERE WERE      7 WARNING FLAGS IN THE ABOVE ASSEMBLY
 ON PAGE NO.
     2
```

EXHIBIT GMP100-4

```
 68_2T 01  10-11-76   10.143
①OCTAL  ②SYMBOL  ③REFERENCES BY ALTER NO.

     10   GEBORT              13      4
     20   START          4     1
      0   TABLE1         2     2      4     6     8    10
     10   TABLE2         3     3      5     7     9    11
  ** 20K LIMITS NEEDED FOR THIS ASSEMBLY.


          SNUMB = 6822T, ACTIVITY # = 02, REPORT CODE = 74, RECORD COUNT = 000084
```

EXHIBIT GMP100-5



```
@P22T 32  10-11-75   10.194
   ORIGIN   DATE MODULE ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                              SUBPROGRAMS INCLUDED IN DECK.
Ⓐ037744 10/11/75 0000 START Ⓑ037764
                           Ⓒ SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

   037654 02/14/75 SSET .SETU. 037661
                                             RANGE                  SIZE
                          Ⓓ ALLOCATED CORE    000000 THRU 037777       040000
                              RELOCATABLE     037654 THRU 037777       000124
                             Ⓔ 1K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY 740410 1/6
                            000010 LOCATIONS REQUIRED FOR LOAD TABLE
                          Ⓕ EXECUTION PROGRAM ENTERED AT  037764   THROUGH  .SETU.
```

EXHIBIT GMP100-6

```
 EI 204442230007  OI 000010001000  IC 037776  IR 002001  BA 444044  EQ 000  AR 000000000007  QR 000000204442  TR 00000200
 X0 000000  X1 000000  X2 000000  X3 000000  X4 000000  X5 000000  X6 000000  X7 000000

*EIS REGISTERS*

000000      000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010      000000000000  000000000000  000000000000  000000000000  000000000000  000000000000. 000000000004  000000000000
```

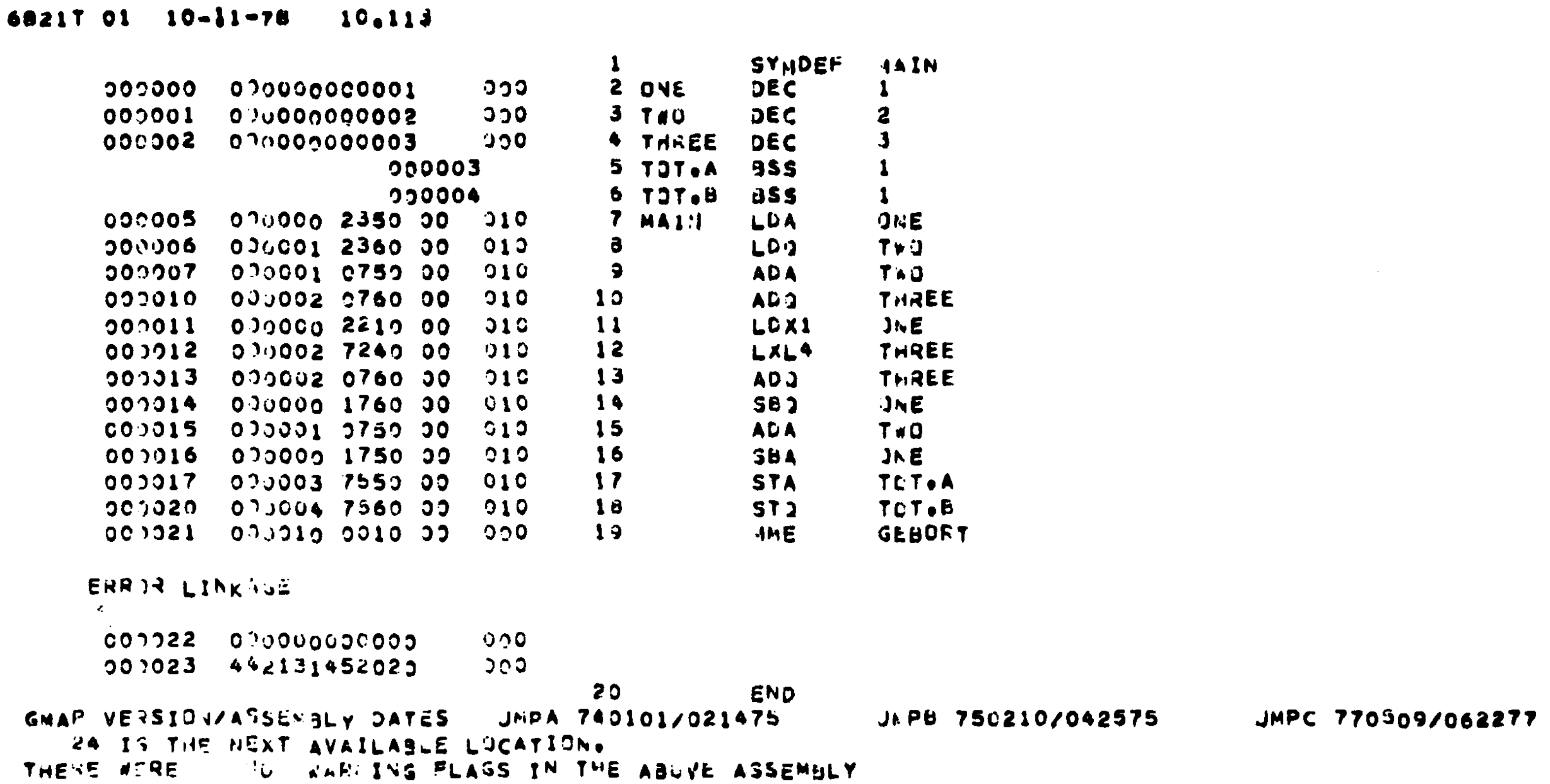
EXHIBIT GMP100-7

ⓒ
```
000000      000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000
000010      000000000000 000000000000 000000000000 037778024642 000000000000 000048074714 000000000000 000000000000
000020      000000000000 000000000000 037775002001 000000000000 037764000000 000000000000 000000000000 000000000000
000030      000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 080610010763 000101037000
000040      002350002081 002350463753 777212464000 000021000000 013666200200 000175000001 776000000000 000035330000
000050      000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000
000060      000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 202020264707 000020000101
000070      030173035207 612324636220 732020202020 202020202024 644447202145 242021620225 660633033302 202020200000
000100      000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000 000000000000
037650*     000000000000 000000000000 000000000000 000000000000 000101037653 000000000000 037654000000 000000000000
037660      037764000000 000003580004 777772222004 000037742000 464040635000 000011735000 777777624001 000002740000
037670      777764721004 037777101003 000002602004 000000621014 000037441000 777763724004 000024605004 777765121000
037700      000001636011 000011772000 777755235004 000001302003 000005600004 000000755012 000001022012 001000130007
037710      000012600004 000000620006 002000136007 000000623002 777743236004 000000520202 000000757012 000001120003
037720      000000011000 777774605004 000007710004 000100622000 000007621000 777775741004 777730236004 777700630000
037730      777751601004 777727235004 777723236004 000024757000 000000621000 034200520201 000000450011 014200520001
037740      000041450011 000000073000 000031450000 777715710024 000000000001 000000000002 000000000003 000000000004
037750      000000000005 000000000006 000000000007 000000000010 000000000001 000000000002 000000000003 000000000004
037760      000000000005 000000000006 000000000007 000000000010 037744237000 037754757000 037746237000 037756757000
037770      037750237000 037760757000 037752237000 037762757000 204642236007 000010001000 000000000000 620321510000
*  UPPER SSA
```

EXHIBIT GMP100-8

```
$$   6821T ENTERED ATC620 AT 10.10R FROM T/S     0-08-03


0001  $        $SNUMB   6821T
0002  $        COMMENT  MURG          TSS CARDIN
0003  $$       USERID   MURGS/ZZZ
0004  $        IDENT    FP700 0J/31,3-7/COTS ,     GMAP FINAL TEST   W6,3,2       00000010
0005  A$       GMAP     NDECK,NXEC                                             00000020
0006  A$       EXECUTE  DUMP                                                   00000230
0007  $        LIMITS   ,12K                                                   00000240
0008  $        ENDJOB                                                          00000250
  TOTAL CARD COUNT THIS JOB = 000028


* ACTY-01   $CARD #0005   GMAP     10/11/78   SW=211040000000
  * NORMAL TERMINATION      AT 004440 I=5020  SW=211040000000

      START  10.112      LINES     53      PROC  0.0001        I/O    0.001      IU   5     MEMORY     24K
      STOP   10.114      LIMIT  10000      LIMIT 0.0400        LIMIT              CU   5     M*T        153
      SWAP    0.000
      LAPSE   0.002      FC  D TYPE      BUSY     IP/AT      FP/RT      IS/#C  MS/#E      ADDRESS T#

                         G*  R D181 *      65         0          1          1      1      0-08-03
                         P*   SYOUT
                         K*   SYOUT
                         C*   SYOUT
                         *1  R D181 *     183         0          0         48     48      0-08-07
                         L*  S D181 *      35         0          1         24     24      0-08-07

       LIST      53 LINES


* ACTY-02   $CARD #0006   GELOAD    10/11/78   SW=400000000000
  * USERS 07 MME GELBRT     AT 027775 I=1020  SW=400000000000

      START  10.115      LINES     83      PROC  0.0002        I/O    0.001      IU   5     MEMORY     12K
      STOP   10.116      LIMIT   5000      LIMIT 0.0500        LIMIT              CU   5     M*T         82
      SWAP    0.000
      LAPSE   0.001      FC  D TYPE      BUSY     IP/AT      FP/RT      IS/#C  MS/#E      ADDRESS T#

                         B*  R D181 *      69         1          1         24     24      0-08-07
                         H*  P D181 *      30         0          0          1      1      0-08-03
                         P*   SYOUT
                         L*  P D181 *     196         0          0        800   800R      0-08-07
                         *L  R D181 *      55         0          0        250   250R      0-08-08

       LIST      83 LINES
```

EXHIBIT GMP100-9



```
6821T 01  10-31-78   10.113

                                            1           SYMDEF   MAIN
     000000  000000000001     000           2 ONE       DEC      1
     000001  000000000002     000           3 TWO       DEC      2
     000002  000000000003     000           4 THREE     DEC      3
                         000003             5 TOT.A     BSS      1
                         000004             6 TOT.B     BSS      1
     000005  000000 2350 00   010           7 MAIN      LDA      ONE
     000006  000001 2360 00   010           8           LDQ      TWO
     000007  000001 0750 00   010           9           ADA      TWO
     000010  000002 0760 00   010          10           ADQ      THREE
     000011  000000 2210 00   010          11           LDX1     ONE
     000012  000002 7240 00   010          12           LXL4     THREE
     000013  000002 0760 00   010          13           ADQ      THREE
     000014  000000 1760 00   010          14           SBQ      ONE
     000015  000001 0750 00   010          15           ADA      TWO
     000016  000000 1750 00   010          16           SBA      ONE
     000017  000003 7550 00   010          17           STA      TOT.A
     000020  000004 7560 00   010          18           STQ      TOT.B
     000021  000010 0010 00   000          19           MME      GEBORT

    ERROR LINKAGE

     000022  000000000000     000
     000023  442131452020     000
                                           20           END
  GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475      JAPB 750210/042575      JMPC 770509/062277
     24 IS THE NEXT AVAILABLE LOCATION.
  THERE WERE     NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP100-10

```
0SST 02  10-11-78   10.176                                                                                                        PAGE

   ORIGIN   DATE MODULE ENTRY LOCATION    E ITRY LJCATIO.I    ENTRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                                             SUBPROGRAMS INCLUDED IN DECK.

   027754 10/11/78 0000 MAIN   027761

                                             SUBPROGRAMS OBTAI:IED FROM SYSTEM LIBRARY

   027664 02/14/75 6SET .SETU. 027671
                                                               RANGE                SIZE
                                 ALLOCATED CORE     000000 THRU 027777       030000
                                    RELOCATABLE     027664 THRU 027777       000114
                                      1K, IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY 740410 1/6
                                 000010 LOCATIONS REJUIRED FUR LUAD TABLE
                                 EXECUTION PROGRAM ENTERED AT  027761    THROUGH  .SETU.

  SI 027760756000  OI 000010001000  IC 027775  IR 102001  BA 274030  ER 000  AR 000000000004  QR 000000000007  TR 00007000
  X0 000000  X1 000000  X2 000000  X3 000000  X4 000003  X5 000000  X6 000000  X7 000000


0EIS REGISTERS.

000000       0C0000000000  800000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010       000000000000  800000000000  000000000000  000000000000  000000000000  000000000000  000000000004  000000000000


0000005      000000000000  800000000000  000000000000  000000000000  009000000000  000000800000  000000000000  000000000000
000010       000000000000  000000000000  000000000000  027775020007  000000000000  000045074714  000000000000  000000000000
000020       000000000000  000000000000  027775102001  000000000000  027761000000  000000000000  000000000000  000000000000
000030       000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  080610020863  000101020003
000040       092350002025  002350273753  777212274000  000025000000  013666200200  000175000001  776000000000  000075700000
00005C       000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
00006C       000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  202020264707  000020000101
000070       030173035207. 612324636220  732020202020  274421472026  314521432063  256263202020  660633033302  202020200000
000100       000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
027660       000000000000  000000000000  000000000000  000000000000  009101027663  000000000000  027664000000  000000000000
027670       027761000000  000003550004  777772222004  000037742000  274030635000  000011738000  777777624001  000002700004
027700       777764721004  027777151003  000002602004  000000021014  000037441000  777763726004  000026606004  777775121004
027710       000001636011  000011772000  777755235004  000001302003  000008600004  000000758012  000001022012  001000130007
027720       000012600004  000000620006  002000136007  000000023002  777743236004  000000520202  000000757612  000001150003
027730       000000011000  777774605004  000007710004  000100622000  000007621000  777775741004  777730236004  777790620003
027740       777751601004  777727235004  777723236004  000024757000  000000621000  034200520201  000000480011  014200600001
027750       000041450011  000000073000  000031450000  777715710024  000000000001  000000000008  000000000003  000000000004
027760       000000000007  027754235000  027755236000  027755075000  027756076000  027754221000  027756720000  027756070000
027770       027754176000  027755075000  027754175030  027757755000  027760756000  000010001000  000000000000  442131400000
```

EXHIBIT GMP100-11

PART 1 - PSEUDO-OPS AND DATA MANIPULATION PROGRAM

1. Create a table using appropriate pseudo-ops labeled "TABLE1" and start on a word address evenly divisible by eight. The table should be eight words long and contain the following numbers in this order:

   -1,-2,-3,-4,5,6,7,8

2. Create an empty table area, 16 words long, starting on a word address evenly divisible by eight. Label this area TABLE2.

3. Move the contents of TABLE1 into TABLE2 in reverse order, skipping every other word in TABLE2. For example:

```
TABLE2+0 = 8 (TABLE1+7)
TABLE2+1 = 0
TABLE2+2 = 7 (TABLE1+6)
        . . .
TABLE2+14=-1 (TABLE1+0)
TABLE2+15= 0
```

4. Terminate the program with an abort and obtain a dump.

5. Circle, in the dump, the following items:

   a. TABLE1 (eight locations).

   b. TABLE2 (sixteen locations).

   c. First executed instruction (probably a load).

   d. Last executed instruction (MME GEBORT).

6. On a blank space in the loadmap, show what computation was done to find the dump addresses.

EXHIBIT GMP110-1

PART 2 - ARITHMETIC INSTRUCTIONS PROGRAM

1. Using the program from Part 1, set up a cumulative total of the numbers in TABLE2 after they are moved from TABLE1.

2. The odd locations in TABLE2 will contain the subtotals as the even locations are accumulated. For example:

```
TABLE2+0 = 8  From TABLE1
TABLE2+1 = 8  Subtotal
TABLE2+2 = 7  FROM TABLE1
TABLE2+3 = 15 Subtotal
           . . .
TABLE2+14=-1  From TABLE1
TABLE2+15=16  Final Total
```

3. Make the coding for the totals a separate section of code from the portion that does the movement from TABLE1 to TABLE2.

4. Terminate the program with an abort and obtain a dump.

5. Find and circle, in the dump, the following locations:

   a. TABLE1+0 through TABLE1+7.

   b. TABLE2+0 through TABLE2+15.

   c. The second executed instruction (probably a store).

   d. The first instruction in the second part.

   e. The next to last executed instruction.

EXHIBIT GMP110-2

```
        7479T ENTERED ATCRIC AT 13.400 FROM TSS/S   0-08-09

 0001  $        SNUMB   7479T
 0002  $        COMMENT KCAT                  TSS CARDIN
 0003  $$       USERID  KCAT$/ZZZ
 0004  $        IDENT   FE776A0J/31,3-7/G1201,GMP120-1
 0005 A$        GMAP    NDECK,NXEC                                      W6.3.2      00000010
 0006 A$        EXECUTE DUMP                                                        00000020
 0007  $        LIMITS  .5K                                                         00000150
 0008  $        ENDJOB                                                              00000160
  TOTAL CARD COUNT THIS JOB = 000020                                                00000170

* ACTY-01  $CARD 00005  GMAP      12/06/78   SW=211040000000
  * NORMAL TERMINATION      AT 002440 I=5020 SW=211040000000

    START 13.401       LINES     46       PROC  0.0001        I/O   0.001     IU  5   MEMORY    24K
    STOP  13.403       LIMIT  10000       LIMIT 0.0400        LIMIT           CU  5   M*T       163
    SWAP   0.000
    LAPSE  0.002   FC  TYPE        BUSY     IP/AT      FP/RT     IS/#C MS/#E    ADDRESS T#

                   G*  D181 *       25        0          1          1     1    0-08-03
                   F*  SYOUT
                   C*  SYOUT
                   C*  SYOUT
                   *1  D181 *      193        0          0         48    48    0-08-01
                   Z*  D181 *       74        0          1         24    24    0-08-08

    LIST     45 LINES AT STA.  U3

* ACTY-02  $CARD 00008  GELOAD    12/06/78   SW=400000000000
  * USERS OO MME GEBORT     AT 011775 I=0020 SW=400000000000

    START 13.403       LINES     62       PROC  0.0002        I/O   0.001     IU  5   MEMORY    5K
    STOP  13.404       LIMIT   5000       LIMIT 0.0500        LIMIT           CU  5   M*T       37
    SWAP   0.000
    LAPSE  0.001   FC  TYPE        BUSY     IP/AT      FP/RT     IS/#C MS/#E    ADDRESS T#

                   G*  D181 *       57        1          1         24    24    0-08-08
                   F*  D181 *        9        0          0          1     1    0-08-03
                   P*  SYOUT
                   L*  D181 *      138        0          0        800   800R   0-08-07
                   *L  D181 *       62        0          0        250   250R   0-08-08

    LIST     45 LINES AT STA.  U3
```

EXHIBIT GMP120-1

```
7,7?1 01  12-06-7?    13.?02                                                            PAGE    2

                                         1          SYMDEF   START                       00000030
   000000  000000000001     000          2 TAB      DEC      1,2,3,4,5                   00000040
   000001  000000000002     000
   000002  000000000003     000
   000003  000000000004     000
   000004  000000000005     000
   000005  000000000000     000          3 TOTAL    DEC      0                           00000050
   000006  000005 2350 00   010          4 START    LDA      TOTAL                       00000060
   000007  000000 0750 00   010          5          ADA      TAB                         00000070
   000010  000001 0750 00   010          6          ADA      TAB+1                       00000080
   000011  000002 0750 00   010          7          ADA      TAB+2                       00000090
   000012  000003 0750 00   010          8          ADA      TAB+3                       00000100
   000013  000004 0750 00   010          9          ADA      TAB+4                       00000110
   000014  000005 7550 00   010         10          STA      TOTAL                       00000120
   000015  000010 0010 00   000         11          MME      GEBORT                      00000130

  ERROR LINKAGE

   000016  000000000000     000
   000017  626021516320     000
                                        12          END                                  00000140
GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475      JMPB 750210/042575      JMPC 770509/062277
   20 IS THE NEXT AVAILABLE LOCATION.
THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP120-2

```
74797 02  12-06-78   13.403                                                                                    PAGE      1

   ORIGIN   DATE MODULE ENTRY LOCATION    ENTRY LOCATION    E TRY LOCATION    ENTRY LOCATION    ENTRY LOCATION

                             SUBPROGRAMS INCLUDED IN DECK.

   011760 12/06/78 0000 START  011766

                             SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

   011670 02/14/75 GSET .SETU. 011675
                                                RANGE                  SIZE
                         ALLOCATED CORE     000000 THRU 011777       012000
                            RELOCATABLE     011670 THRU 011777       000110
                              1K. IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY 740410 1/G
                         000010 LOCATIONS REQUIRED FOR LOAD TABLE
                         EXECUTION PROGRAM ENTERED AT  011766   THROUGH  .SETU.

 EI C11765755000  OI 000010001000  IC 011775  IR 002001  BA 724012  ER 000  AR 000000000017  QR 000000000000  TR 00007373
 X0 000000  X1 000000  X2 000000  X3 000000  X4 000000  X5 000000  X6 000000  X7 000000


*EIS REGISTERS*

000000    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000004  000000000000



000000S   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010    000000000000  000000000000  000000000000  011775020000  000000000000  000045074714  000000000000  000000000000
000020    000000000000  000000000000  011775002001  000000000000  011760000000  000000000000  000000000000  000000000000
000030    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  020704071163  000101011667
000040    002350002021  002350323753  777212324000  000021000000  013666200200  000175000001  776000000000  000075253000
000050    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000060    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  202020262507  070621004161
000070    030173035207  612701020001  732744470102  005201202020  202020202020  202020202020  660633033302  202020202020
000100    000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
011670*   000101011667  000000000000  011670000000  000000000000  011760000000  000003550004  777772222004  000037742000
011700    324012635000  000011735000  777777624001  000002744004  777764721004  011777101003  000002602004  000000621014
011710    000037441000  777763724004  000024605004  777755121004  000001636011  000011772000  777755235004  000001302003
011720    000005600004  000000755012  000001022012  001000136007  000012600004  000000620006  002000136007  000000623002
011730    777743234004  000000520202  000000757012  000001123003  000000011000  777774605004  000007710004  000100622000
011740    000007621000  777775741004  777730236004  777700636002  777751601004  777727235004  777723236004  000024757000
011750    000000621000  014200520201  000000450011  014200520201  000041450011  000000073000  000031450000  777715710024
011760    000000000001  000000000002  000000000003  000000000004  000000000005  000000000017  011765235000  011760075000
011770    011761075000  011762075000  011763075000  011764075000  011765755000  000010001000  000000000000  626321516320
```

EXHIBIT GMP120-3

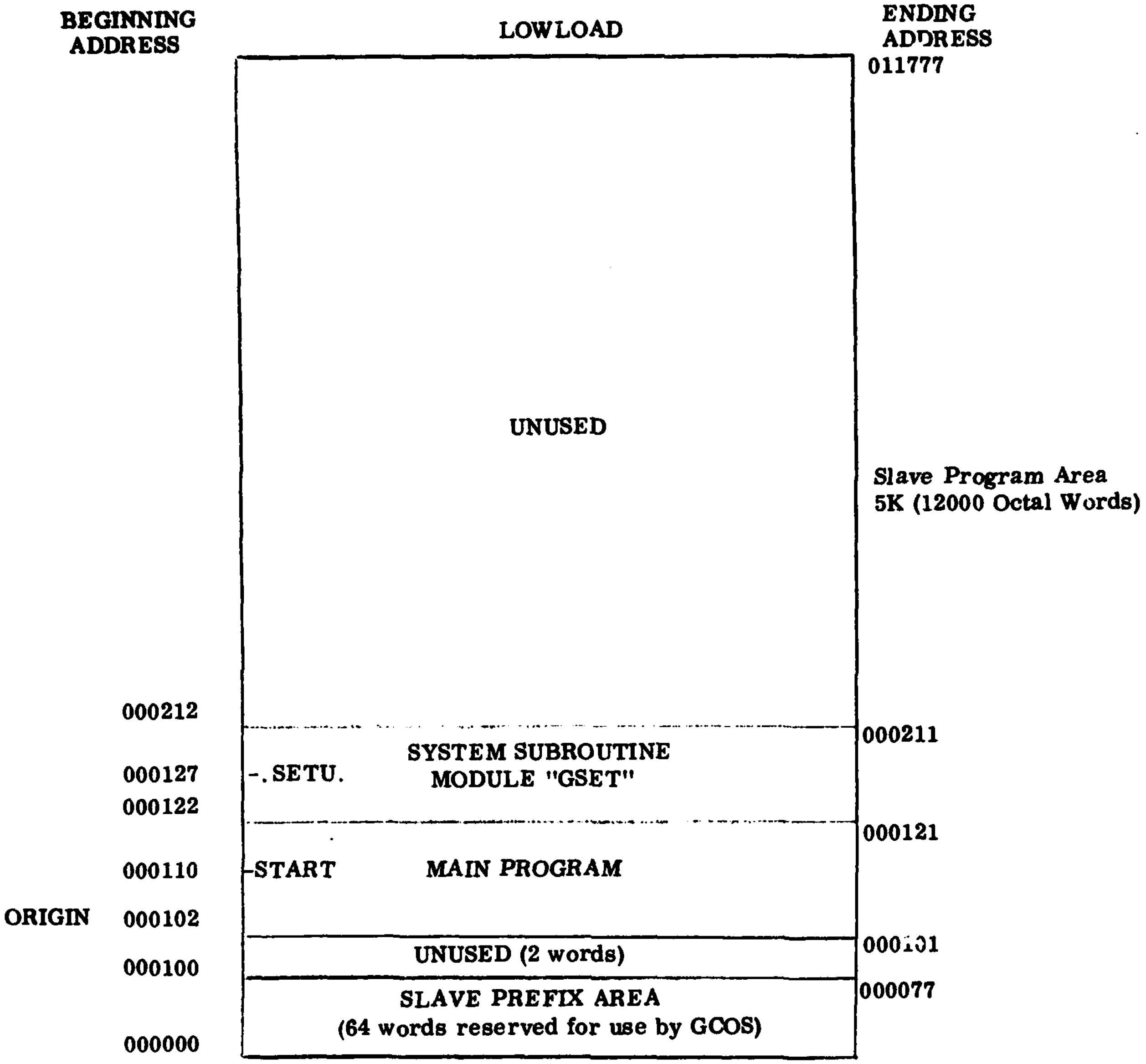
```
7,90T 01  12-06-7    13.49                                                                   PAGE     2

                                                 1           SYMDEF   START                             00000030
     00C000  000000000001       00C      2 TAB       DEC      1,2,3,4,5                        00000040
     000001  000000000002       000
     00C002  000000000003       000
     000003  000000000004       000
     000004  000000000005       000                                                         00000050
     000005  000000000000       000      3 TOTAL     DEC      0                                00000060
     000006  000005 2350 00     010      4 START     LDA      TOTAL                            00000070
     000007  000000 0750 00     010      5           ADA      TAB                              00000080
     000010  000001 0750 00     010      6           ADA      TAB+1                            00000090
     000011  000002 0750 00     010      7           ADA      TAB+2                            00000100
     000012  000003 0750 .00    010      8           ADA      TAB+3                            00000110
     000013  000004 0750 00     010      9           ADA      TAB+4                            00000120
     000014  000005 7550 00     010     10           STA      TOTAL                            00000130
     000015  000010 0010 00     000     11           MME      GEBORT

    ERROR LINKAGE

     000016  000000000000       000
     000017  620321516320       000     12           END                                      00000140
 GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475      JMPB 740210/042575      JMPC 770509/062277
   20 IS THE NEXT AVAILABLE LOCATION.
  THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP120-6

```
                        SUBPROGRAMS INCLUDED IN DECK.

                         LOCATED                                                        00000015
 000102 12/06/7- 00'0 START  000110

                        SUBPROGRAMS OBTAINED FROM SYSTEM LIBRARY

 000122 02/14/75 GS.T .SETU. 000127
                                          RANGE                SIZE
                   ALLOCATED CORE     000000 THRU 011777       012000
                     RELOCATABLE      000100 THRU 000211       000112
                        1K. IS THE MINIMUM MEMORY NEEDED TO LOAD THIS ACTIVITY 740410 1/G
                   000010 LOCATIONS REQUIRED FOR LOAD TABLE
                   EXECUTION PROGRAM ENTERED AT  000110   THROUGH  .SETU.

 EI 000107755000  OI 000010001000  IC 000117  IS C02001  BA 336012  ER 000  AR 000000000017  QR 000000000000  TR 00007372
 X0 000000  X1 000000  X2 000000  X3 000000  X4 000000  X5 000000  X6 000000  X7 000000


*FIS REGISTERS*

000000   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010   000000000000  000000000000  000000000020  000000000000  000000000020  000000000000  000000000004  000000000000

7490T 02  12-06-7.   13..71                                                                             PAGE      1

  ORIGIN   DATE MODULE  ENTRY LOCATION  ENTRY LOCATION  ENTRY LOCATION  ENTRY LOCATION  ENTRY LOCATION  ENTRY LOCATION

000005   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000010   000000000000  000000000000  000000000000  000117020000  000000000000  000045074714  000000000000  000000000000
000020   000000000000  000000000000  000117002001  000000000000  000110400000  000000000000  000000000000  000000000000
000030   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  020704110063  000212011777
000040   002350002023  002350335753  777212336000  000023000000  013666200200  000175000001  776000000000  000076030000
000050   000000000000  000000000000  000000000000  000000000000  000000003000  000000000000  000000000000  000000000000
000060   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  202020262507  070621004161
000070   030173031207  612701020001  732744470102  005201202020  202020202020  202020202020  660633033302  202020202020
000100   000000000000  000000000000  000000000001  000000000002  000000000003  000000000004  000000000005  000000000017
000110   000107231000  000102075000  000103075000  000104075000  000105075000  000106075000  000107755000  000010001000
000120   000000000000  5 6321516320  000212011777  000000000000  000100000000  000000000000  000110400000  000003550004
000130   777772222004  000037742000  336012635000  000011735000  777777624001  000002744004  777764721004  011777101003
000140   000002601004  000000621014  000037441000  777763724004  000024603004  777755121004  000001636011  000011772000
000150   777755235004  000001302003  000005600004  000000755012  000001022012  001000136007  000012600004  000000620006
000160   002000131007  000000623002  777743236004  000005202002  000000757012  000001123003  000000011000  777774605004
000170   000007710004  0 0100622000  000007621000  777775741004  777730236004  777700636002  777751601004  777727235004
000200   777723231004  000024757000  000000621000  034200520201  000000450011  014200520201  000041450011  000000073000
000210   000031450000  777715710024  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
000220   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
011770*  00000                                   000000000000  000000000000  000000000000  000000000000  000000000000  000000000000
*  UPPER FSA
```

EXHIBIT GMP120-7

EXHIBIT GMP120-3C



Map of the Slave Program Area Generated by the General Loader for the Program Shown in Exhibits GMP120-3A, B, C.

EXHIBIT GMP120-8

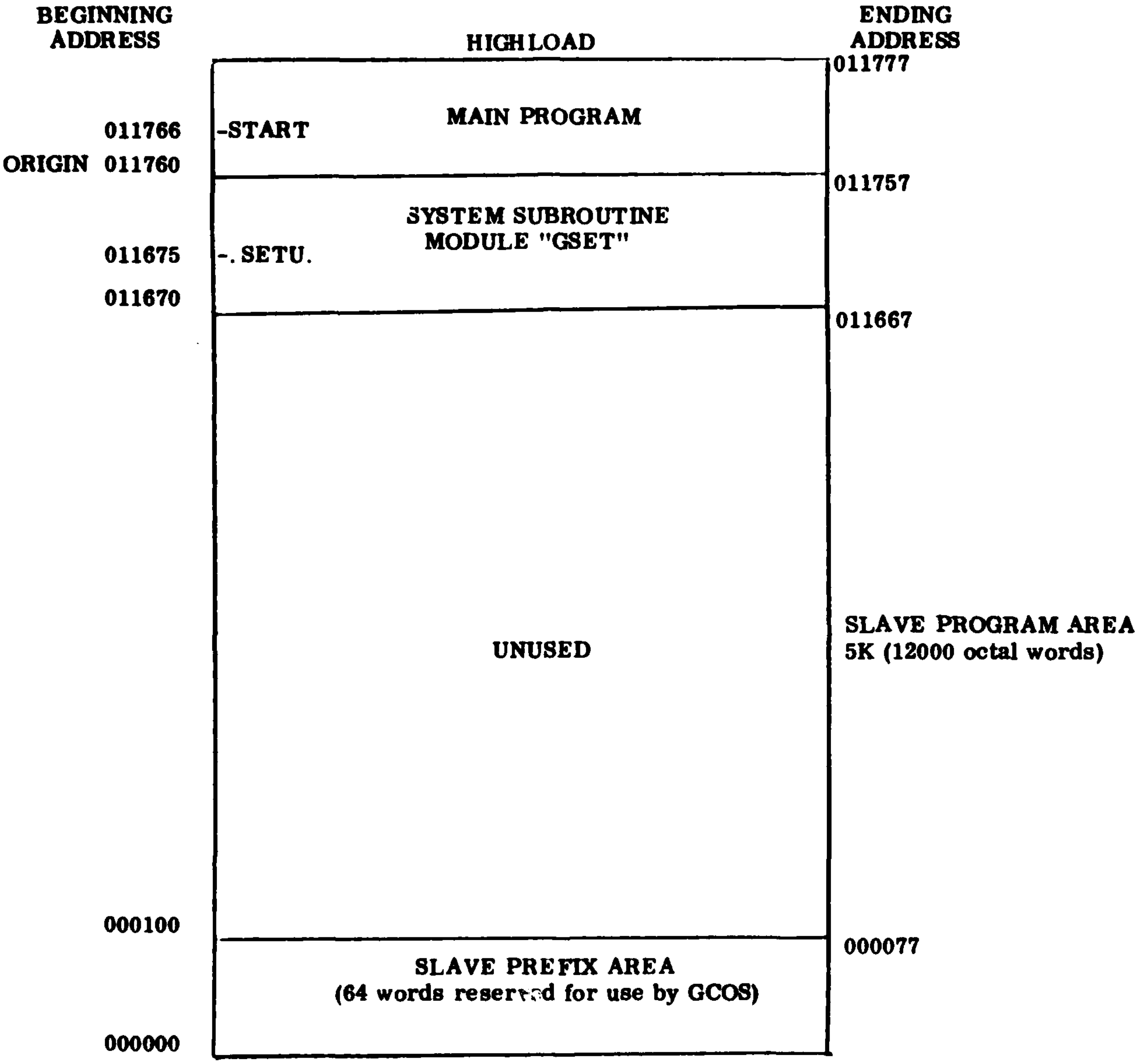
OCTAL REPRESENTATION OF BCD DATA

BCD	OCTAL	BCD	OCTAL	BCD	OCTAL
A	21	J	41	S	62
B	22	K	42	T	63
C	23	L	43	U	64
D	24	M	44	V	65
E	25	N	45	W	66
F	26	O	46	X	67
G	27	P	47	Y	70
H	30	Q	50	Z	71
I	31	R	51		

LOGICAL COMPARISON EXAMPLES

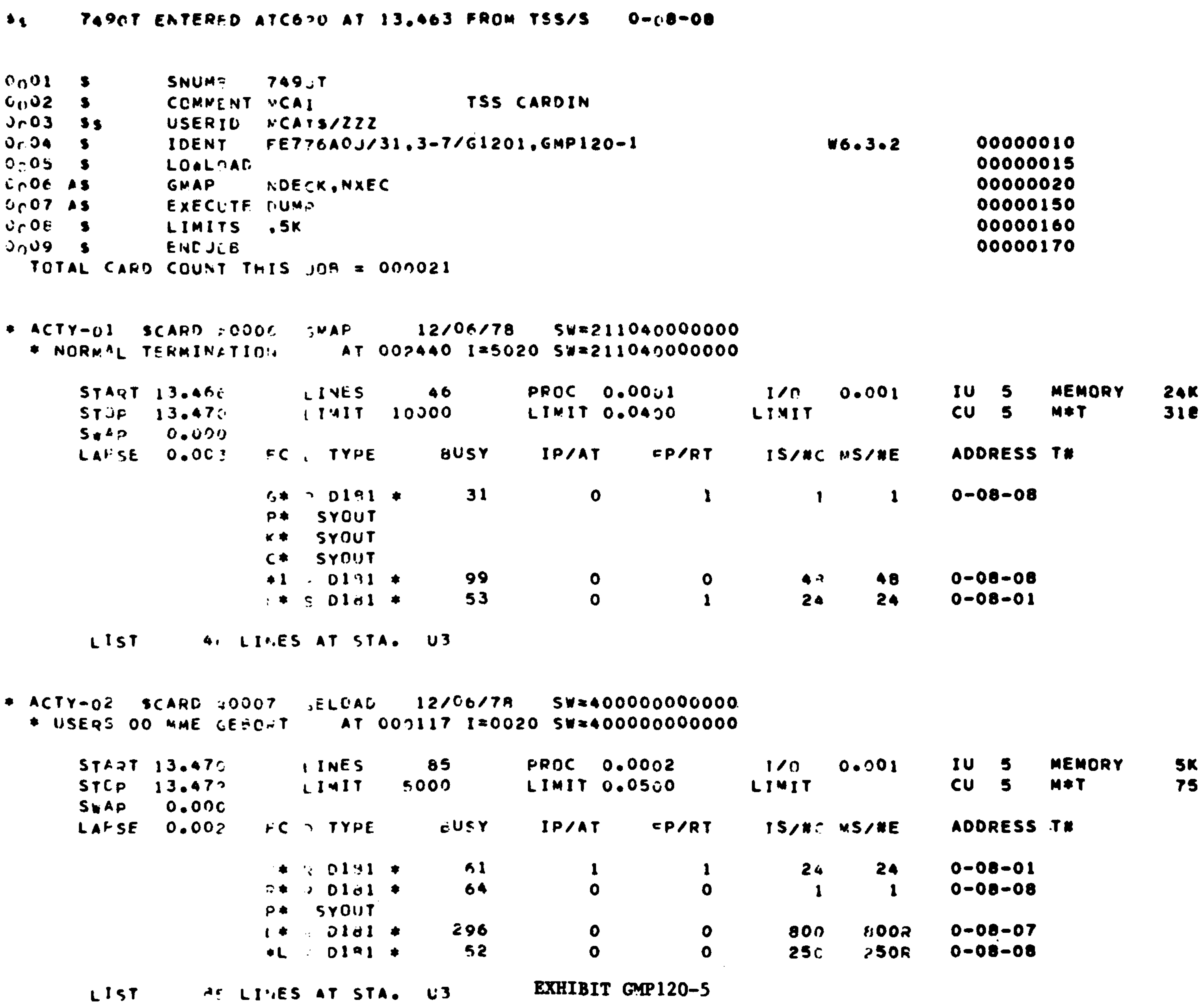
	EXAMPLE #1	EXAMPLE #2	EXAMPLE #3
	C(R) > C(Y)	C(R) = C(Y)	C(R) < C(Y)
REGISTER	Z 111001	Z 111001	I 011001
MEMORY	I 011001	Z 111001	Z 111001
REGISTER	111001	111001	011001
MEMORY COMPLEMENT	+100111	+100111	+000111
SCRATCH PAD REG	1 100000	1 000000	100000
	CARRY ON	CARRY ON	CARRY OFF
	NEGATIVE ON	NEGATIVE OFF	NEGATIVE ON
	ZERO OFF	ZERO ON	ZERO OFF

EXHIBIT GMP130-1



Map of the Slave Program Area Generated by the General Loader for the Program Shown in Exhibits GMP120-1A, B, C.

EXHIBIT GMP120-4



```
$$    749CT ENTERED ATC620 AT 13.463 FROM TSS/S    0-08-08


0001  $      SNUMB    749JT
0002  $      COMMENT  MCAI               TSS CARDIN
0003  $$     USERID   MCAIS/ZZZ
0004  $      IDENT    FE776AOJ/31,3-7/G1201,GMP120-1                W6.3.2      00000010
0005  $      LOWLOAD                                                           00000015
0006  AS     GMAP     NDECK,NXEC                                               00000020
0007  AS     EXECUTE  DUMP                                                     00000150
0008  $      LIMITS   .5K                                                      00000160
0009  $      ENDJOB                                                            00000170
  TOTAL CARD COUNT THIS JOB = 000021


* ACTY-01  $CARD 00006  GMAP     12/06/78   SW=211040000000
  * NORMAL TERMINATION      AT 002440 I=5020 SW=211040000000

      START 13.466     LINES     46      PROC  0.0001       I/O    0.001    IU  5    MEMORY    24K
      STOP  13.470     LIMIT  10000      LIMIT 0.0400       LIMIT           CU  5    M*T       318
      SWAP   0.000
      LAPSE  0.003     FC L TYPE     BUSY     IP/AT     EP/RT     IS/#C  MS/#E    ADDRESS T#

                       G* ? D191 *     31        0         1         1      1     0-08-08
                       P*  SYOUT
                       K*  SYOUT
                       C*  SYOUT
                       *1 . D191 *     99        0         0        48     48     0-08-08
                       :* S D181 *     53        0         1        24     24     0-08-01

        LIST     4. LINES AT STA.  U3


* ACTY-02  $CARD 00007  GELOAD   12/06/78   SW=400000000000
  * USERS 00 MME GEBORT     AT 000117 I=0020 SW=400000000000

      START 13.470     LINES     85      PROC  0.0002       I/O    0.001    IU  5    MEMORY     5K
      STOP  13.472     LIMIT   5000      LIMIT 0.0500       LIMIT           CU  5    M*T        75
      SWAP   0.000
      LAPSE  0.002     FC ? TYPE     BUSY     IP/AT     EP/RT     IS/#C  MS/#E    ADDRESS T#

                        * ? D191 *     61        1         1        24     24     0-08-01
                       ?* ? D181 *     64        0         0         1      1     0-08-08
                       P*  SYOUT
                       ( *   D181 *   296        0         0       800   800R     0-08-07
                       *L    D191 *    52        0         0       250   250R     0-08-08

        LIST     ?? LINES AT STA.  U3
```

EXHIBIT GMP120-5

LOOPING AND TABLE INDEXING PROGRAM

1. Using the specifications from the programs listed on exhibits GMP110-1 and GMP110-2, use looping and indexing to perform the same functions as in Part 2.

2. Maintain the same modular format for the two functions:

   a. Moving table values.

   b. Summing the table entries.

3. Your program output should be in the form of a dump obtained by a "MME GEBORT".

4. In the dump, indicate the instructions which have TABLE1 or TABLE2 as operands and show where to find, in the panel of the dump, the final settings of the registers used by your program. (NOTE: the A-register, Q-register, and index registers are the only ones used.)

5. Your coded GMAP program should contain the following 5 program steps before it is a valid solution to the assigned problem:

   a. Initialization

   b. Execution

   c. Incrementation

   d. Comparison

   e. Test/Transfer

Subfield No.	1	8	16
1		FILCB	LOCSYM of file control block,
2		ETC	2-character file code,
3		ETC	LOCSYM of first buffer,
4		ETC	LOCSYM of second buffer,
5		ETC	maximum block size,
6		ETC	record form (0 = variable, 1 = fixed, 2 = mixed),
7		ETC	record size (no. if fixed, LOCSYM if mixed),
8		ETC	1 if standard BSN not used,
9		ETC	LOCSYM of error routine,
10		ETC	1 if no standard labels,
11		ETC	recording mode (0 = binary, 1 = BCD, 2 = mixed, 3 = 9-track tape),
12		ETC	recording density (0 = high, 1 = low),
13		ETC	1 if file is on multifile reel,
14		ETC	retention period in days,
15		ETC	LOCSYM of preheader label routine,
16		ETC	LOCSYM of postheader label routine,
17		ETC	LOCSYM of pretrailer label routine,
18		ETC	LOCSYM of posttrailer label routine,
19		ETC	file name (12-character maximum),
20		ETC	Exception Processing override (0-7),
21		ETC	1 if partitioned records

SUBFIELDS OF FILE CONTROL BLOCK

EXHIBIT GMP200-1

	0 5 6 17	18 29	30 35	
-14	PREHEADER LABEL EXIT	▨	MBZ	4
-13	POSTHEADER LABEL EXIT	▨	MBZ	
-12	PRETRAILER LABEL EXIT	▨	MBZ	
-11	POSTTRAILER LABEL EXIT	▨	MBZ	
-10	FIRST SIX CHARACTERS OF FILE-NAME			3
-9	LAST SIX CHARACTERS OF FILE-NAME			
-8	RETENTION PERIOD	REEL SEQUENCE NO.		1
-7	FILE SERIAL NUMBER		▨	
-6	BLOCK COUNT	FILE COUNT		
-5	ERROR ROUTINE EXIT		▨	
-4	▨ LINE - ID	MBZ FILE CODE		
-3	IOS - STATUS RETURN WORD			
-2	IOS - STATUS RETURN WORD			
-1	FCB POINTER			
LOCSYM	CURRENT RECORD INDEX		MBZ	
+1	RECORD SIZE	▨	▨	
+2	LOC. OF 1ST BUFFER	LOC. OF 2ND BUFFER		2
+3	WORKING STORAGE WORD			
+4	BLOCK SIZE	▨	▨	
+5	RECORD SIZE ROUTINE LOC.	▨	MBZ	5

File Control Block

EXHIBIT GMP200-2

```
1      8       16
______________________________________________________________
       VFD     18/fcb,1/io,1/open,2/close,1/prime,1/size,1/req,1/res,10/file
```

fcb = Symbolic location of the file control block

io = Input/output indicator value

open = Rewind-on-open option value

close = Rewind-on-close option value

prime = Prime input buffer option value

size = Put-size on output option value

req = Required file option

res = Reserved for system use

file = File number

The subfields must account for all 36 bits (by count) of the file designator word. Commas separate the subfields, with the last subfield terminated by a blank. No blanks are permitted within the field.

The format for the file designator word is as follows:

Bits

0-17 Location of the file control block.

18 Input/output indicator:
- 0 Input file.
- 1 Output file.

19 Rewind-on-open option:
- 0 Either do not rewind the file or this option is not appropriate to the device.
- 1 Rewind the file when opened.

This option is ignored in the time-sharing environment. Mass storage files are always in the "rewind-on-open" condition.

20-21 Rewind-on-close option:
- 0 This option is not appropriate to this device.
- 1 Lock the file when closed. Release via MME GERELS.
- 2 Do not rewind and do not lock the file when closed.
- 3 Rewind but do not lock the file when closed.

EXHIBIT GMP200-3

22 Prime input buffer option:

- 0 Normal operation.
- 1 This is a buffered input file and the buffer is not be filled when the file is opened.

23 Put-size on output option:

- 0 Normal operation.
- 1 This is a buffered output file for which a CALL PUTSZ with record size equal zero is to be issued when the file is closed.

24 Optional/required status on open:

- 0 Normal operation.
- 1 Abort if file is not present (is ignored in the time-sharing environment).

25 Reserved for System use

26-35 File positioning value for open:

- 0 Honor the rewind option as given in bit 19 but do not otherwise position the device.
- n This file is the nth file on a multifile tape reel. Rewind the tape and space forward over n-1 files.

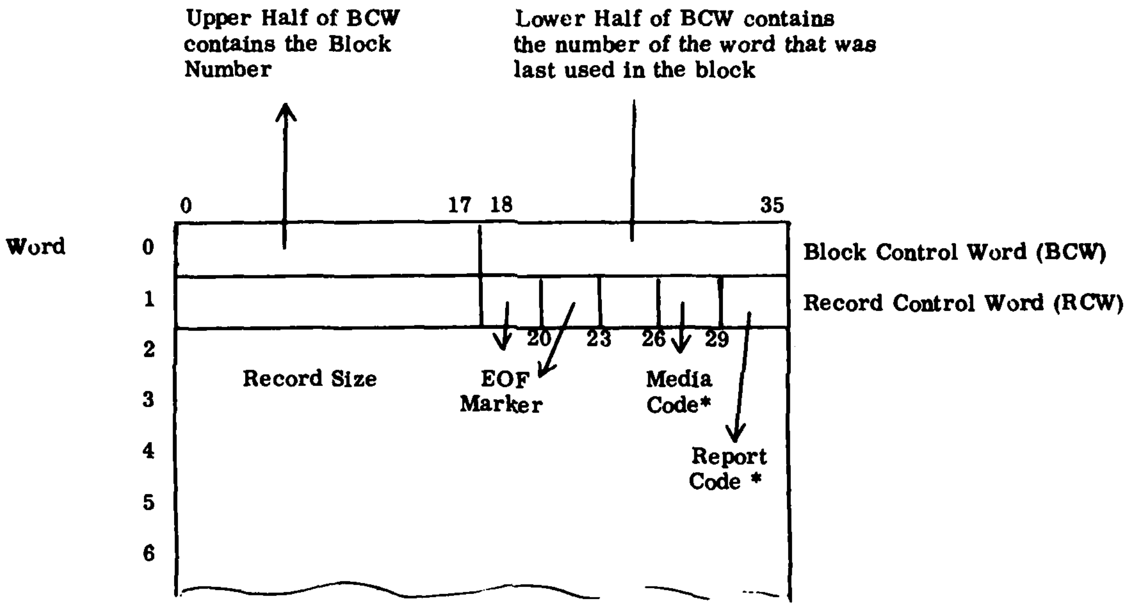
File Designator Word (FDW)

EXHIBIT GMP200-3 (Cont)

```
                  1
1       8         6    --CARD COLUMNS

        FILCB     INPUT,IN,INBUF1,INBUF2
        FILCB     OUTPUT,OT,OTBUF1,,,,22
FILES   VFD       18/INPUT,18/0
        VFD       18/OUTPUT,1/1,17/0
INBUF1  BSS       321
INBUF2  BSS       321
OTBUF1  BSS       321
```

SAMPLE INPUT/OUTPUT FILE CONTROL BLOCK INSTRUCTION

EXHIBIT GMP200-4


Upper Half of BCW contains the Block Number
Lower Half of BCW contains the number of the word that was last used in the block
0
17
18
35
Word
0
1
2
3
4
5
6
Block Control Word (BCW)
Record Control Word (RCW)
20
23
26
29
Record Size
EOF Marker
Media Code*
Report Code *

* MEDIA CODES

1 - Binary Card Image
2 - Hollerith Card Image (BCD)
3 - Print Line Image
5 - ASCII Image

* REPORT CODES

74 - Execution Report
75 - Compressed Deck
76 - Object Deck
77 - Alter Input List

Six words for Block Showing the Block Control Word and Record Control Word Formats

EXHIBIT GMP200-5

```
WORD 0      000002000470
     1      000016000200
     2      532020202020
     3      202744214720
     4      202020452425
     5      234220202020
     6      202020202020
     7      202020202020
     8      202020202020
     9      202020202020
```

TEN WORDS OF A BLOCK OF DATA

EXHIBIT GMP200-6

CONTENTS OF PRNTLN

+7	+8	+9	+10	+11	+12
	THIS I	S THE	FIRST		

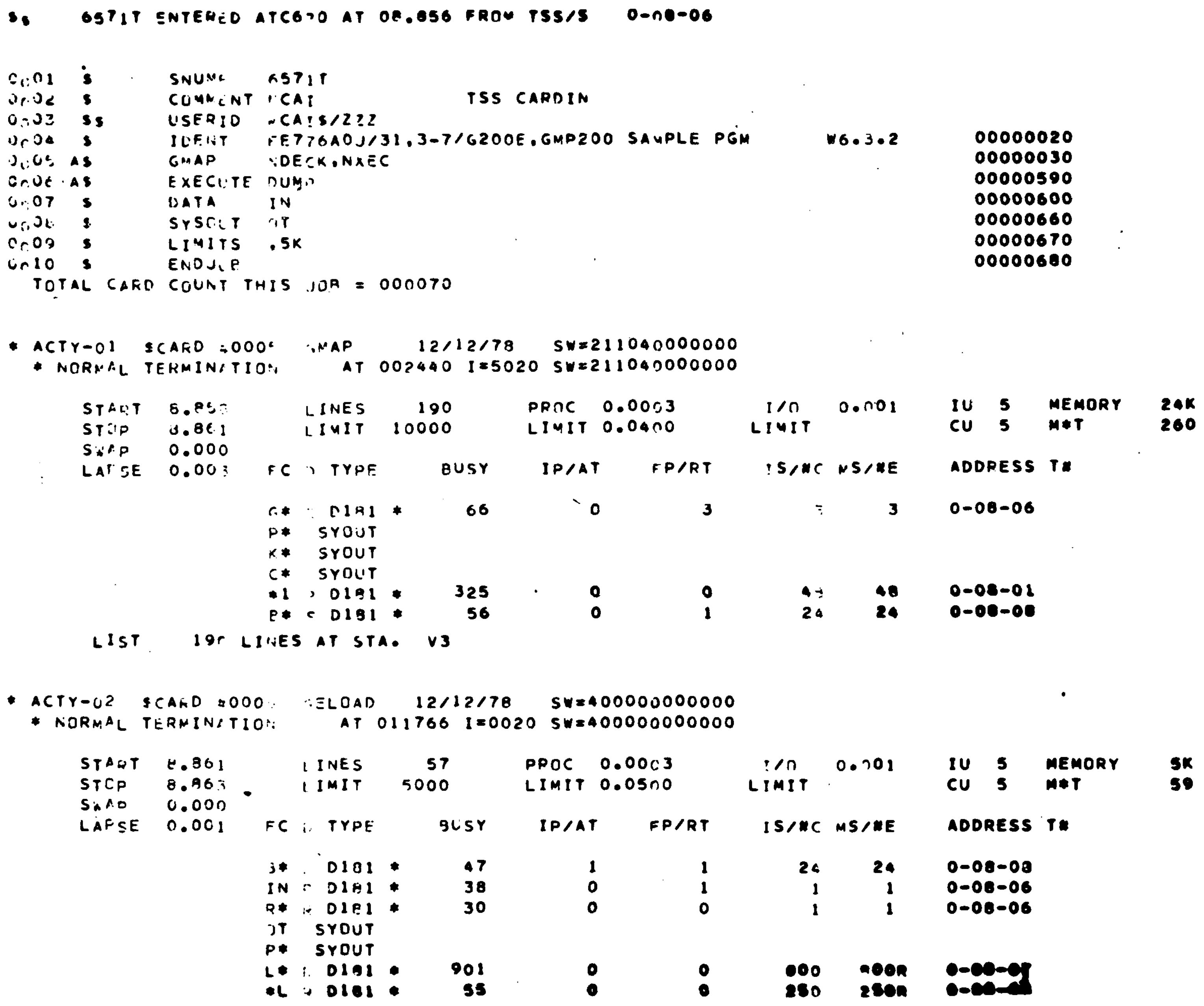
EXHIBIT GMP200-7

+7	+8	+9	+10	+11	+12
	THIS I	T IS	FIRST		

EXHIBIT GMP200-8

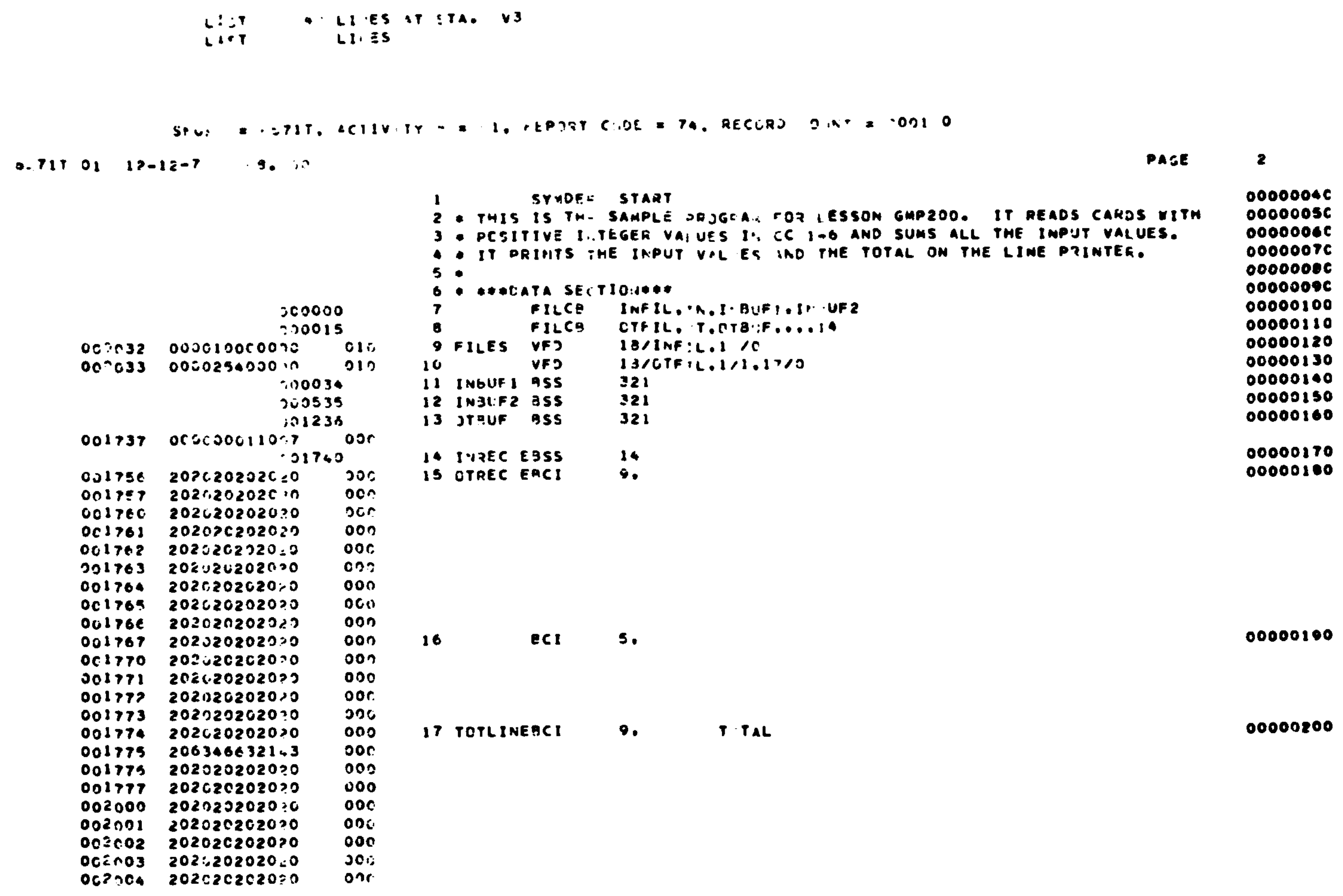
+7	+8	+9	+10	+11	+12
	THIS I	T IS			

EXHIBIT GMP200-9



```
$$    6571T ENTERED ATC6?0 AT 08.856 FROM TSS/S   0-08-06

0001  $      SNUMB    6571T
0002  $      COMMENT  FCAI              TSS CARDIN
0003  $$     USERID   .CAIS/ZZZ
0004  $      IDENT    FE776A0J/31,3-7/G200E,GMP200 SAMPLE PGM      W6.3.2      00000020
0005  A$     GMAP     NDECK,NXEC                                               00000030
0006  A$     EXECUTE  DUMP                                                     00000590
0007  $      DATA     IN                                                       00000600
0008  $      SYSOUT   OT                                                       00000660
0009  $      LIMITS   ,5K                                                      00000670
0010  $      ENDJOB                                                            00000680
  TOTAL CARD COUNT THIS JOB = 000070


* ACTY-01  $CARD #0005  GMAP      12/12/78   SW=211040000000
  * NORMAL TERMINATION      AT 002440 I=5020 SW=211040000000

     START  8.858      LINES    190     PROC  0.0003      I/O   0.001    IU  5   MEMORY    24K
     STOP   8.861      LIMIT  10000     LIMIT 0.0400      LIMIT          CU  5   M*T       260
     SWAP   0.000
     LAPSE  0.003   FC D TYPE      BUSY     IP/AT      FP/RT     IS/#C  MS/#E   ADDRESS T#

                    G*   D181 *      66         0          3         3      3   0-08-06
                    P*  SYOUT
                    K*  SYOUT
                    C*  SYOUT
                    *1 D D181 *     325         0          0        48     48   0-08-01
                    B* C D181 *      56         0          1        24     24   0-08-08
      LIST    190 LINES AT STA.  V3

* ACTY-02  $CARD #0007  GELOAD    12/12/78   SW=400000000000
  * NORMAL TERMINATION      AT 011766 I=0020 SW=400000000000

     START  8.861      LINES     57     PROC  0.0003      I/O   0.001    IU  5   MEMORY     5K
     STOP   8.863      LIMIT   5000     LIMIT 0.0500      LIMIT          CU  5   M*T        59
     SWAP   0.000
     LAPSE  0.001   FC D TYPE      BUSY     IP/AT      FP/RT     IS/#C  MS/#E   ADDRESS T#

                    3*   D181 *      47         1          1        24     24   0-08-08
                    IN C D181 *      38         0          1         1      1   0-08-06
                    R* R D181 *      30         0          0         1      1   0-08-06
                    OT  SYOUT
                    P*  SYOUT
                    L* L D181 *     901         0          0       800   800R   0-08-07
                    *L P D181 *      55         0          0       250   250R   0-08-0[illegible]
```

EXHIBIT GMP200-10



```
                   LIST       * LINES AT STA.  V3
                   LIST           LINES


                SPU:  = .571T, ACTIVITY = 1, REPORT CODE = 74, RECORD COUNT = 001 0

 .71T 01  12-12-7     9. 0                                                                    PAGE       2

                                     1          SYMDEF   START                                                    0000004C
                                     2 * THIS IS TH- SAMPLE PROGRAM FOR LESSON GMP200.  IT READS CARDS WITH      0000005C
                                     3 * PCSITIVE INTEGER VALUES IN CC 1-6 AND SUMS ALL THE INPUT VALUES.       0000006C
                                     4 * IT PRINTS THE INPUT VALUES AND THE TOTAL ON THE LINE PRINTER.          0000007C
                                     5 *                                                                       0000008C
                                     6 * ***DATA SECTION***                                                    0000009C
                       000000        7          FILCB    INFIL,IN,INBUF1,INBUF2                                  00000100
                       000015        8          FILCB    OTFIL,OT,OTBUF,...,14                                   00000110
   000032  000010000000     010      9 FILES    VFD      18/INFIL,1 /0                                           00000120
   000033  000025400000     010     10          VFD      18/OTFIL,1/1,17/0                                       00000130
                       000034       11 INBUF1   BSS      321                                                     00000140
                       000535       12 INBUF2   BSS      321                                                     00000150
                       001236       13 OTBUF    BSS      321                                                     00000160
   001737  000000011007     000
                       001740       14 INREC    EBSS     14                                                      00000170
   001756  202020202020     000     15 OTREC    EBCI     9.                                                      00000180
   001757  202020202020     000
   001760  202020202020     000
   001761  202020202020     000
   001762  202020202020     000
   001763  202020202020     000
   001764  202020202020     000
   001765  202020202020     000
   001766  202020202020     000
   001767  202020202020     000     16          BCI      5.                                                      00000190
   001770  202020202020     000
   001771  202020202020     000
   001772  202020202020     000
   001773  202020202020     000
   001774  202020202020     000     17 TOTLINEBCI      9.        TOTAL                                           00000200
   001775  206346632143     000
   001776  202020202020     000
   001777  202020202020     000
   002000  202020202020     000
   002001  202020202020     000
   002002  202020202020     000
   002003  202020202020     000
   002004  202020202020     000
```

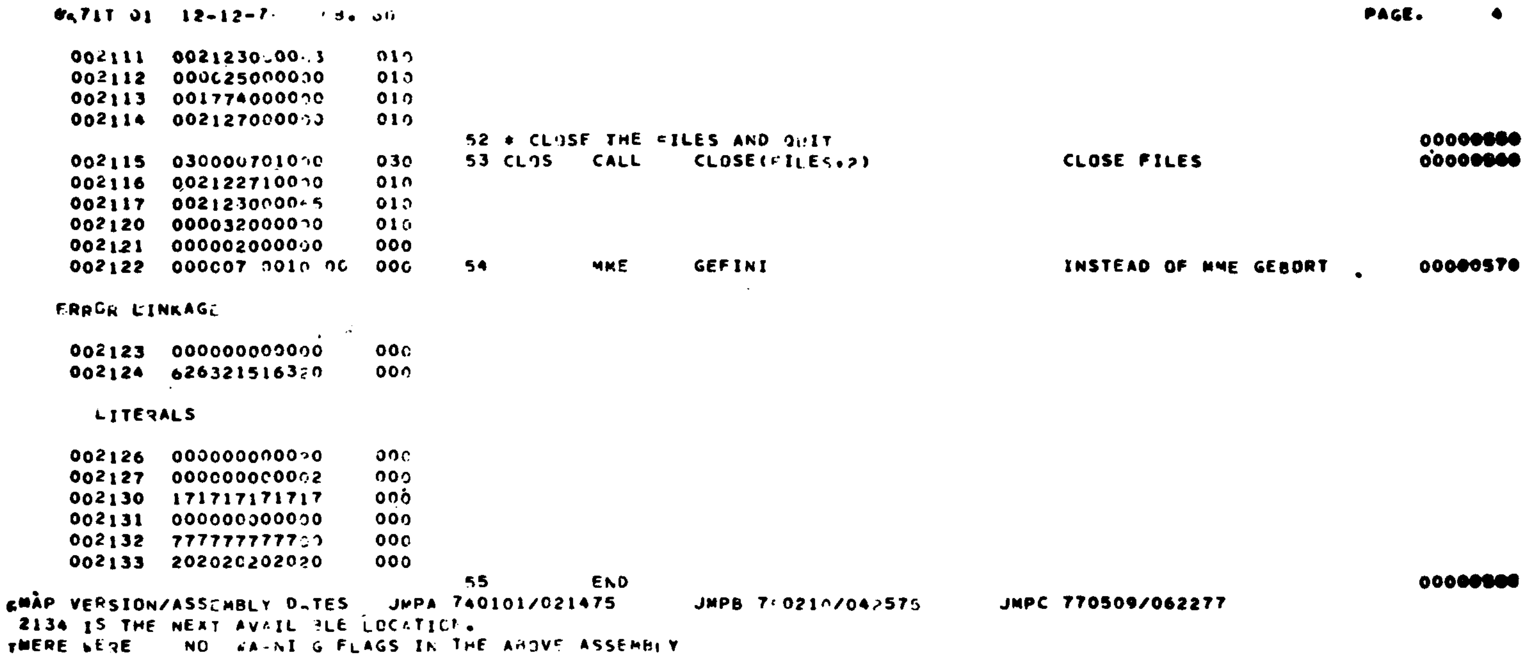
EXHIBIT GMP200-10 (Cont)

```
     002005  202020202020   000    18        PCI                                                           00000210
     002006  202020202020   000
     002007  202020202020   000
     002010  202020202020   000
     002011  202020202020   000
     002012  274447020000   000    19 HDGLINEBCI     9,GMP200 SAMPLE PROGRAM OUTPUT                        00000220
     002013  206221444743   000
     002014  252047514627   000
     002015  512144204664   000
     002016  634764632020   000
     002017  202020202020   000
     002020  202020202020   000
     002021  202020202020   000
     002022  202020202020   000
     002023  202020202020   000    20        BCI     5.                                                    00000230
     002024  202020202020   000
     002025  202020202020   000
     002026  202020202020   000
GMAP 01  12-12-76   08.160                                                                         PAGE     3

     002027  202020202020   000
                    002030         21 TOTAL  BSS     1                                                     00000240
                                   22 * ***EXECUTABLE INSTRUCTIONS***                                      00000250
     002031  000000 0110 00  000   23 START  NOP                                                           00000260
     002032  020000701000   030    24        CALL    OPEN(FILES,2)                OPEN FILES               00000270
     002033  002037710000   010
     002034  002123000030   010
     002035  000032000000   010
     002036  000002000000   000
     002037  050000701000   030    25        CALL    PRINT(OTFIL,HDGLIN,=16)      PRINT HEADING            00000280
     002040  002045710000   010
     002041  002123000031   010
     002042  000025000000   010
     002043  002012000000   010
     002044  002126000000   010
     002045  010000701000   030    26 RFED   CALL    GET(INFIL,EOF,INREC)         READ A CARD              00000290
     002046  002053710000   010
     002047  002123000032   010
     002050  000010000000   010
     002051  002073000000   010
     002052  001740000000   010
     002053  001740 2350 00  010   27        LDA     INREC                        MOVE NUMBER FOR PRINT:   00000300
     002054  001756 7550 00  010   28        STA     OTREC                                                 00000310
     002055  050000701000   030    29 PRT    CALL    PRINT(OTFIL,OTREC,=2)        PRINT INPUT NUMBER       00000320
     002056  002063710000   010
     002057  002123000035   010
     002060  000025000000   010
     002061  [illegible]    01
     002062  [illegible]    01
```

EXHIBIT GMP200-10 (Cont)

```
                                          30 * CONVERT INPUT NUMBE  TO BINARY AND ADD TO TOTAL FOR A SINGLE WORD ONLY 00000330
002063  001756 2360 00  010               31 CONV    LDQ     QTREC                                                    00000340
002064  002130 3760 00  010               32         ANQ     =O171717171717                                           00000350
002065  002131 2350 00  010               33         LDA     =0                                                       00000360
002066  060000701000    030               34         CALL    .GBNRY                       BINARY VAL PLACED IN Q-REG  00000370
002067  002071710000    010
002070  002123000042    010
002071  002030 0560 00  010               35         ASQ     TOTAL                        ADD VALUE TO TOTAL          00000380
002072  002045 7100 00  010               36         TRA     REED                         GET ANOTHER CARD            00000390
                                          37 * CONTROL IS TRANSFERRED TO EOF WHEN NO CARDS REMAIN.  AT THAT TIME.     00000400
                                          38 * WE CONVERT TOTAL TO BCD AND PRINT IT.                                  00000410
                                                                                                                      00000420
002073  002030 2350 00  010               39 EOF     LDA     TOTAL
002074  040000701000    030               40         CALL    .GBCD                        BCD VAL PLACED IN Q-REG     00000430
002075  002077710000    010
002076  002123000050    010
002077  001774 7560 00  010               41         STQ     TOTLIN                                                   00000440
                                          42 *GET RID OF LEADING ZEROS                                                00000450
002100  002132 2360 00  010               43         LDQ     =O777777777700                                           00000460
002101  001774 3160 00  010               44         CANQ    TOTLIN                                                   00000470
002102  002105 6000 00  010               45         TZE     *+3                                                      00000480
002103  000006 7360 00  000               46         QLS     6                                                        00000490
002104  002101 7100 00  010               47         TRA     *-3                                                      00000500
002105  002133 3760 00  010               48         ANQ     =6H                                                      00000510
002106  001774 2560 00  010               49         ORSQ    TOTLIN                                                   00000520
                                          50 *PRINT THE TOTAL LINE AFTER MASKING OFF LEADING ZEROS                    00000530
002107  050000701000    030               51         CALL    PRINT(QTFIL,TOTLIN,=2)       PRINT TOTAL                 00000540
002110  002115710000    010
```

EXHIBIT GMP200-10 (Cont)



```
G471T 01  12-12-7    18. 30                                                                                  PAGE.    4

  002111  002123000003     010
  002112  000025000000     010
  002113  001774000000     010
  002114  002127000000     010
                                  52 * CLOSE THE FILES AND QUIT                                               00000000
  002115  030000701000     030    53 CLOS    CALL     CLOSE(FILES,2)            CLOSE FILES               00000000
  002116  002122710000     010
  002117  002123000045     010
  002120  000032000000     010
  002121  000002000000     000
  002122  000007 0010 00   000    54         MME      GEFINI                    INSTEAD OF MME GEBORT  .  00000570

ERROR LINKAGE

  002123  000000000000     000
  002124  626321516320     000

    LITERALS

  002126  000000000020     000
  002127  000000000002     000
  002130  171717171717     000
  002131  000000000000     000
  002132  777777777700     000
  002133  202020202020     000
                                  55         END                                                              00000000
GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475     JMPB 740210/042575     JMPC 770509/062277
 2134 IS THE NEXT AVAILABLE LOCATION.
THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP200-10 (Cont)

EXHIBIT GMP200-6 (Cont)

```
5171T 01  12-12-7-   08.30                                                                            PAGE     4

      002111  002123000013    010
      002112  000025000000    010
      002113  001774000000    010
      002114  002127000000    010
                                   52 * CLOSE THE FILES AND QUIT                                           00000550
      002115  030000701000    030  53 CLOS    CALL    CLOSE(FILES,2)               CLOSE FILES             00000560
      002116  002122710000    010
      002117  002123000045    010
      002120  000032000000    010
      002121  000002000000    000
      002122  000007 0010 00  000  54         MME     GEFINI                       INSTEAD OF MME GEBORT   00000570

     ERROR LINKAGE

      002123  000000000000    000
      002124  626321516320    000

       LITERALS

      002126  000000000000    000
      002127  000000000002    000
      002130  171717171717    000
      002131  000000000000    000
      002132  777777777770    000
      002133  202020202020    000
                                   55         END                                                          00000580
GMAP VERSION/ASSEMBLY DATES   JMPA 740101/021475     JMPB 7?0210/042575     JMPC 770509/062277
 2134 IS THE NEXT AVAILABLE LOCATION.
THERE WERE    NO  WARNING FLAGS IN THE ABOVE ASSEMBLY
```

EXHIBIT GMP200-10 (Cont)

GMAP PROGRAMMING PROJECT III

1. Read in a deck of cards using file and record control subroutines.

   a. The cards are formatted as follows:

      (1) CARD COLUMNS 3-8 . . . FIELD A

      (2) CARD COLUMNS 9-12 . . . FIELD B

      (3) CARD COLUMNS 13-18 . . . FIELD C

      (4) CARD COLUMNS 25-30 . . . FIELD D

   b. Fields A, B, C, and D contain positive integers, unsigned.

   c. Each numeric field is right-justified within its own area.

   d. Input is in BCD form.

   e. There are no leading zeros in the input.

2. Compute: FIELD A + FIELD B + FIELD C - FIELD D = ANSWER

3. Print the following on the line printer:

   a. Print a heading line with a label for each of the fields, the answer column, and two areas labeled "NEG" and "GR 10".

   b. Print one line for each card consisting of the four fields, the answer derived from the computation, and a flag if the answer is negative or greater than 10.

4. Punch the following card deck for input to your program:

```
         111111111122222222223
123456789012345678901234567890      --CARD COLUMNS

      10   2     3          15
      10   3     2          20
       2   3    10           3
       8   4     3           9
       9   9     9          29
       6   4     4           0
     320  10    50         370
      42   6     8          10
       0   0     0           9
       3   8     8         100
```

EXHIBIT GMP210-1

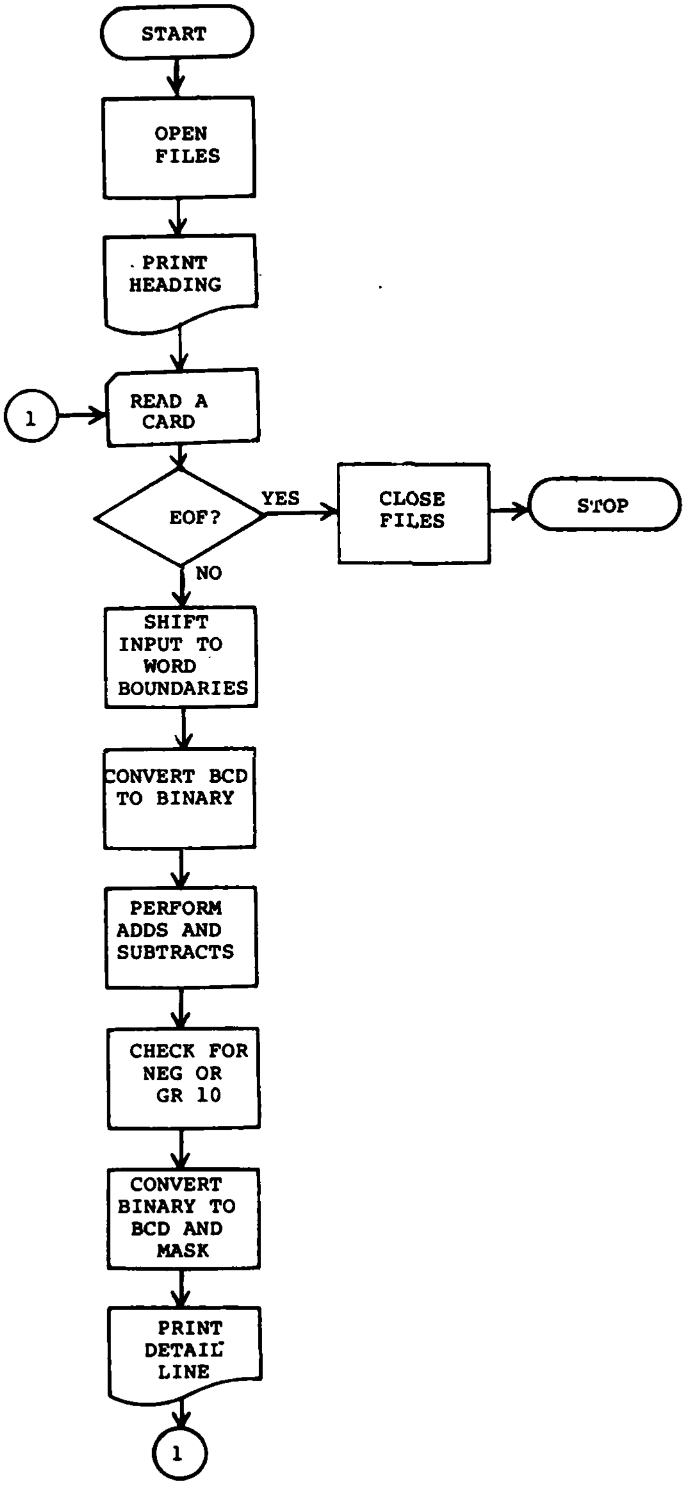


EXHIBIT GMP210-2

BCD TO BINARY SUBROUTINE

```
                1
1      8        6     <-- CARD COLUMNS

 Actual Conversion Routine              (Note)
       LDQ      BCDVAL
       ANQ      =0171717171717
       LDA      =0
       CALL     .GBNRY
```

NOTE: At this point, the binary value is in the Q-register. You can store it or do anything you wish with it.

BINARY TO BCD SUBROUTINE

```
                1
1      8        6     <-- CARD COLUMNS

 Actual Conversion Routine          (Note)
       LDA      BINVAL
       CALL     .GBCD
 BCD value is in the Q-register     (Note)
       STQ      BCDVAL
 Get rid of leading zeros           (Note)
       LDQ      =0777777777700
       CANQ     BCDVAL
       TZE      *+3
       QLS      6
       TRA      *-3
       ANQ      =6H
       ORSQ     BCDVAL
```

NOTE: You may substitute any locsym you wish in the routines to to replace the locsyms "BINVAL" AND "BCDVAL". All the other information, with the exception of the comments, is required and must be coded exactly as stated.

EXHIBIT GMP210-3

H700 TEST AND DIAGNOSTICS (H6000-CDT)
COURSE EXHIBITS

MAIN FRAME PROCEDURES

700-7200 PROCESSOR/700-1210 MEMORY (PRELIMINARY)

STEP 1 - Load the AB16-CCT4 card deck using the normal program loading procedure with the boot loader. The following teletype message should occur:

```
AB16 CCT4 APR 24, 73 REV H
SENS SW TO A REG 8 TIMES
```

STEP 2 - Depress A on REGISTER SELECT. Depress REGISTER CLEAR. Depress SENSE SWITCH 4. Depress START. Only Bit 4 of the DISPLAY REGISTER should be illuminated. Depress SENSE SWITCH 3. Depress START. Only Bits 3 and 4 should be illuminated. Depress SENSE SWITCH 2. Depress START. Only Bits 2, 3, and 4 should be illuminated. Depress SENSE SWITCH 1. Depress START. Only Bits 1, 2, 3, and 4 should be illuminated. Reset SENSE SWITCH 4. Depress START. Only Bits 1, 2, and 3 should be illuminated. Reset SENSE SWITCH 3. Depress START. Only Bits 1 and 2 should be illuminated. Reset SENSE SWITCH 2, depress START. Only Bit 1 should be illuminated. Reset SENSE SWITCH 1. Depress START. No bits should be illuminated and the following teletype message should occur:

```
DONE
CP is 716 MMSZ 8K
```

The program automatically starts a diagnostic test which is executed in about 100 seconds. If the program detects an error, it stops. This must be detected by the operator by either a time out or observance of non-flickering of the DISPLAY REGISTER indicators and a dark RUN indicator. At a successful completion of the test this teletype message occurs:

```
PS 0001
```

Put STOP/RUN in STOP. Depress Y on REGISTER SELECT.

STEP 3 - Load the AB16-TIME 1 card deck using the normal program loading procedure with the boot loader. The following teletype message should occur:

```
AB16 - TIME 1 DOC NO 70185768000 REV A
INSTRUCTION TIMING DEMONSTRATION PROGRAM
TYPE R (RTC) or T (60 SEC) and CR.
```

STEP 4 - Enter an R and a RETURN on the teletype. The teletype will output

```
WHAT IS THE REAL TIME CLOCK FREQ IN Hz
```

EXHIBIT TAD700-1

STEP 5 - Enter the number 60 and a RETURN on the teletype. The program will execute and output a teletype message similar to the following:

```
INSTRUCTION   CONDITION      TIME

    NOP       NO SKIP        800 ±  40
    LDA                     1600 ±  80
    LDA       INDIRECT      2400 ± 120
    STA                     1600 ±  80
    SZE       A = 0         1100 ±  55
    CAS         =           1900 ±  85
    CAS         <           2200 ± 110
    ALR       1 SHIFT       1100 ±  55
    ALR       2 SHIFTS      1400 ±  70
    SKS       ASR-INT       2700 ± 135
    MPY                     4050 ± 203
    DIV                     6700 ± 335
END OF PROGRAM
ST
```

NOTE: The times here will not be printed in this format but the times that are printed should fit within these ranges.

STEP 6 - Load the AB16-11T1 card deck using the normal program loading procedure with the boot loader. The following teletype message should occur:

```
AB16-11T1  REV E JAN 5, 72
```

The program automatically starts a diagnostic test which is executed in about 55 seconds. The program will half if an error is detected. Operator actions are the same as STEP 2 in this case. At a successful completion of the test (no errors) the following teletype message occurs:

```
END OF PASS 0000050000
```

Put STOP/RUN in STOP.

STEP 7 - Load the AB16-CMT5 card deck using the normal program loading procedures with the boot loader. The following teletype message should occur:

```
AB16-CMT5  JUN 7, 73 REV E
CP IS 716
```

The program automatically starts a diagnostic test. If no errors are detected, a teletype output similar to the following occurs about every 8 seconds.

```
PS 0001   001673   017777
PS 0002   000021   016116
```

If the above teletype messages occur and there are no error messages, put STOP/RUN in STOP. The processor test is completed.

EXHIBIT TAD700-1 (Cont)

CONSOLE TELETYPE PROCEDURES

700-5307 ASR-33 TELETYPE SUBSYSTEM

STEP 1 - Load the AG16-TWT1 card deck using the normal program loading procedure with the boot loader. All switches level on the CPU. Depress MASTER CLEAR. Select A on REGISTER SELECT. Depress BIT 7 causing '001000 to be set into the A Register. Select Y on REGISTER SELECT. Key in Y = '0010000. Depress STOP/RUN to RUN. Depress START. The following teletype message should occur:

```
AG16-TWT1      3 NOV  71      REV. B
 TYPE XOFF
```

STEP 2 - While holding down the CTRL key, depress the XOFF/S key on the teletype. This must be done within 15 seconds. The following teletype message should occur:

```
FULL DUPLEX FUNCTION TEST AA
TYPE XOFF
```

While holding down the CTRL key, depress the XOFF/S key on the teletype. The following teletype message should occur:

```
                          A
TYPE XOFF
```

While holding down the CTRL key, depress the XOFF/S key on the teletype. The following teletype message should occur:

```
                          A
TYPE XOFF
```

While holding down the CTRL key, depress the XOFF/S key on the teletype. The following teletype message should occur:

```
                          A
TYPE XOFF
```

While holding down the CTRL key, depress the XOFF/S key on the teletype. The following teletype message should occur:

```
                          A
PAGE PRINTER TEST
```

STEP 3 - Place the STOP/RUN switch on the CPU in the STOP position. Depress A on REGISTER SELECT. Depress REGISTER CLEAR. Depress BIT 13 causing '000010 to be set into the A Register. Depress Y on REGISTER SELECT. Depress STOP/RUN to RUN. Depress START. A teletype message similar to that in Exhibit TAD700-3 should occur. It consists of a rotating printout of eight lines of all printable symbols followed by a full line for each symbol appearing in the first line of printout (including blanks). This printout takes about nine minutes. In the first eight lines, the correct printout can be observed by scanning

EXHIBIT TAD700-2

semi-vertically along a line of similar characters. During the printout of a line for each symbol, observe that the carriage return mechanism operates correctly and starts each line in the same place with no skipped characters. Note that certain lines of the sample printout contain this error condition. This malfunction tends to become worse and worse until messages are difficult to read. The print quality should also be observed for possible need to change the ribbon, clean the mechanism, etc. After the page printer output, the following message should occur:

```
KEYBOARD INPUT
```

STEP 4 - Enter the following characters on the teletype:

```
1234567890:-QWERTYUIOPASDFGHJKL:ZXCVBNM,./!"#$%&'()*=<-@
[\+↑]<>?
CHARACTER COUNT = 63
FULL DUPLEX KEYBOARD TEST
```

STEP 5 - Enter a W on the teletype. No character should be printed. Enter an M on the teletype; a W should be printed. Enter an X on the teletype; an M should be printed. Depress RETURN on the teletype; an X should be printed followed by the message:

```
TEST FOR ANSWER BACK DRUM
```

The teletype chatters during this test. The program will halt if an error is detected. The operator must observe this by either a time-out or observance of non-flickering DISPLAY REGISTER indicators and a dark RUN indicator. If no errors are detected, the following teletype message should occur in about six seconds:

```
NULL CHARACTER TEST - NO CHARACTER SHOULD PRINT
```

A row of null characters should be output to the teletype (the operator should observe that no characters are printed) followed by the teletype message:

```
DONE
PUNCH/READER TEST
TURN ON ASR33 PUNCH - THEN PRESS START
```

STEP 6 - Load a blank paper tape in the paper tape punch following the procedures in the equipment operator's guide. Depress ON on the paper tape punch. Depress START on the CPU. A teletype message should be printed out while the paper tape is being punched consisting of all the symbols in various combinations.

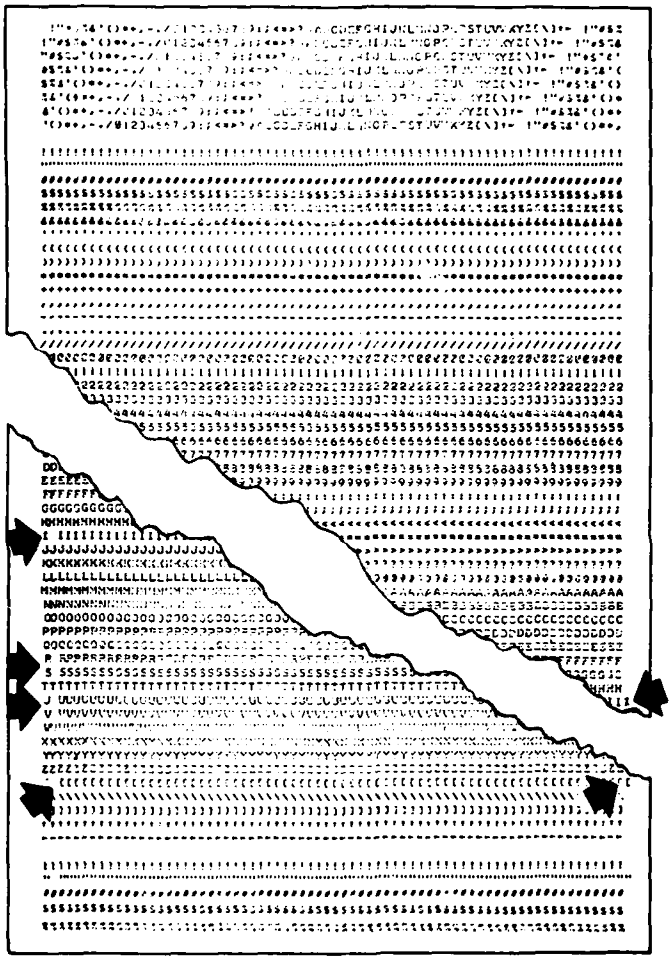
STEP 7 - Depress OFF on the paper tape punch. Carefully tear off the paper tape which was punched and load it into the paper tape reader following procedures in the equipment operator's guide. Depress START on the CPU and START on the paper tape reader. The paper tape should be read: a halt of the paper tape input signifies an error. If the tape was punched and read correctly, the following teletype message should occur:

EXHIBIT TAD700-2 (Cont)

```
DONE
FULL DPLX RDR PUNCH TEST
TURN ON ASR33 PUNCH - THEN PRESS START
```

It is not necessary to exercise this feature since simultaneous paper tape punch and read operations are not required operationally. The test is complete.

EXHIBIT TAD700-2 (Cont)



Page Printer Output with Errors

EXHIBIT TAD700-3

LINE PRINTER PROCEDURES

700-5515 LINE PRINTER SUBSYSTEM/700-5517 96-132 COLUMN UPGRADE

STEP 1 - Remove the operational vertical format tape from the line printer and load the test control vertical format tape for AB16-55T3. The TYPE ROLL ON and START switches on the line printer should be illuminated.

STEP 2 - Load the AB16-55T3 card deck using the normal program loading procedure with the boot loader. The following teletype message should occur:

```
AB16-553T3 REV G NOV 72
LPM?:
```

Enter number 200 and RETURN on the teletype. The following teletype message should occur:

```
COL?:
```

Depress RETURN on the teletype. The following teletype message should occur:

```
BUS?:
```

Depress RETURN on the teletype. The following teletype message should occur:

```
CHN?:
```

Depress RETURN on the teletype. The following teletype message should occur:

```
PI?:
```

Depress RETURN on the teletype. The following teletype message should occur:

```
DEV ADR?:
```

Depress RETURN on the teletype. The following teletype message should occur:

```
MSK BIT?:
```

Depress RETURN on the teletype. (If the line printer is not in a ready status, an error message will occur on the teletype and the test must be restarted. It is not necessary to reload the card deck to restart the test; initialize the line printer and place the STOP/RUN switch on the CPU in STOP. With all switches level on the CPU, depress MASTER CLEAR. Key in Y = '001000. Depress STOP/RUN to RUN. Depress START. The header format at the beginning of STEP 2 should occur on the teletype.) The line printer test should begin to execute; if there are no errors, two lines of data should be output on the printer. The first line should consist of the message I/O BUS NON-INTERRUPT TEST followed by a space and the print font: @ABCDEFGHIJKLMNOPQRSTUVWXYZ[\] -!" #$%&'()*+,-./0123456789:;<=>@ABCD EFGHIJK. The second line should consist of the symbols 1W. The following teletype message should occur:

OCP '203 W/O '103 - LINE SHOULD BE BLANK

The line printer should then output two more lines of data on the first page below that printed above. The first line should consist of the message I/O BUS INTERRUPT TEST followed by a space and the print font. The second line consists of the message STATUS WORD + SKS TEST ON I/O BUS followed by the print font. The printer should then advance and print at the top of the next page the message HEAD-OF-FORM followed by the print font. The printer should then advance again and repeat this output. This operation occurs very quickly. The three pages of printout should be examined after the test is completed. The following teletype message should occur:

HIT STOP, FORMS AND START

STEP 3 - On the line printer depress the STOP switch, the FORM/SPACE switch, and the START switch in that order. The following teletype message should be repeatedly output:

OPEN TYPE ROLL HOUSING

STEP 4 - Raise the cabinet assembly on the line printer to allow access to the internal printer controls. Release the latch on the front door assembly and open the door. The Fault indicator should be illuminated. The teletype output should then change to the following repeated message:

CLOSE TYPE ROLL HOUSING

If the teletype continues to output OPEN TYPE ROLL HOUSING, this operational check is in error.

STEP 5 - Close and relatch the front door assembly. The Fault indicator should extinguish in about 5 seconds. After this, depress the START switch on the line printer. The following teletype message should occur:

NEXT?:

If the teletype message continues to repeat CLOSE TYPE ROLL HOUSING rather than NEXT?, this operational check is in error.

STEP 6 - Enter the message MIX followed by depressing RETURN on the teletype. The line printer should then output the following:

128 lines, each composed of all E's

132 lines of characters consisting of a pattern containing a complete line of symbols for each symbol in the font @ABC...? (including the space) in that order repeated twice followed by a line of @'s, a line of A's, a line of B's, and a line of C's.

132 lines of characters in which common symbols appear from line to line along a diagonal caused by stepping the character print positions one place to the left as each line is advanced.

266 lines of asterisks in a pattern consisting of lines of decreasing length, repeated completely twice and then partially.

Two passes through the channel advance control which results in two identical printouts containing the line advance and channel advance printouts followed by the complete print font on each line.

Twenty-two outputs of a three-line canned text.

The following teletype message should occur:

NEXT?:

STEP 7 - Remove the test control vertical format tape and install the operational vertical format tape. Restore the line printer to a "ready" condition. A full-size standard printout should be maintained with these test procedures for comparison with the test results. Compare the test output of *he line printer with the retained reference printout. Correspondence and adequate print quality indicate a successful test.

AB16-55T3 VERTICAL FORMAT TAPE PREPARATION/INSTALLATION

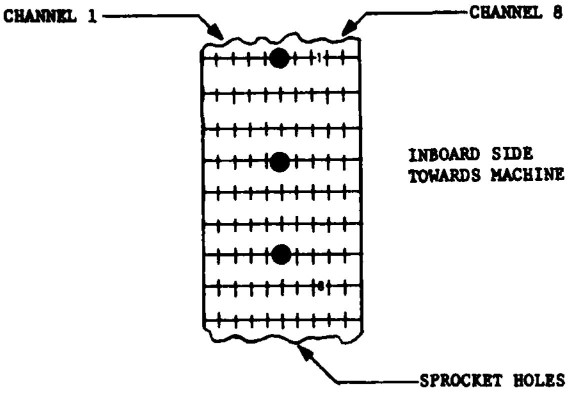
PREPARATION OF VERTICAL FORMAT TAPE FOR AB16-55T3 DIAGNOSTIC TEST

STEP 1 - Obtain a strip of blank vertical format tape about 12 inches long (either 6 LPI or 8 LPI depending on the density commonly used at the site). The tape must be manually punched with the appropriate hand punch. The channel number coding to be used is indicated in Table III.

Channel Number Coding for Test Tape

Line 1	Channel 1	Head of Form
8	3	
20	4	
29	5	
38	6	
47	7	
61	2	End of Form
67	1	Coincident with Line 1

Starting at the top of the strip at a line with a sprocket hole, write the number 1 on the line. Then, count and label the lines according to the above Channel Number Coding diagram (the next numbered line, 8, etc.).



Preparation of Test Tape

EXHIBIT TAD700-5

STEP 2 - Place the tape in the hand punch so that line 1 can be punched. Punch a hole in channel 1. Advance the tape so that line 8 can be punched. Punch a hole in channel 3. Punch the remainder of the holes according to the table.

STEP 3 - Cut off the extra paper tape about 1/4" before line 1 and 1/4" after line 67. Glue the strip in a loop so that lines 1 and 67 coincide (with the printing on the outside of the loop).

Installation of AB16-55T3 Diagnostic Test Vertical Format Tape

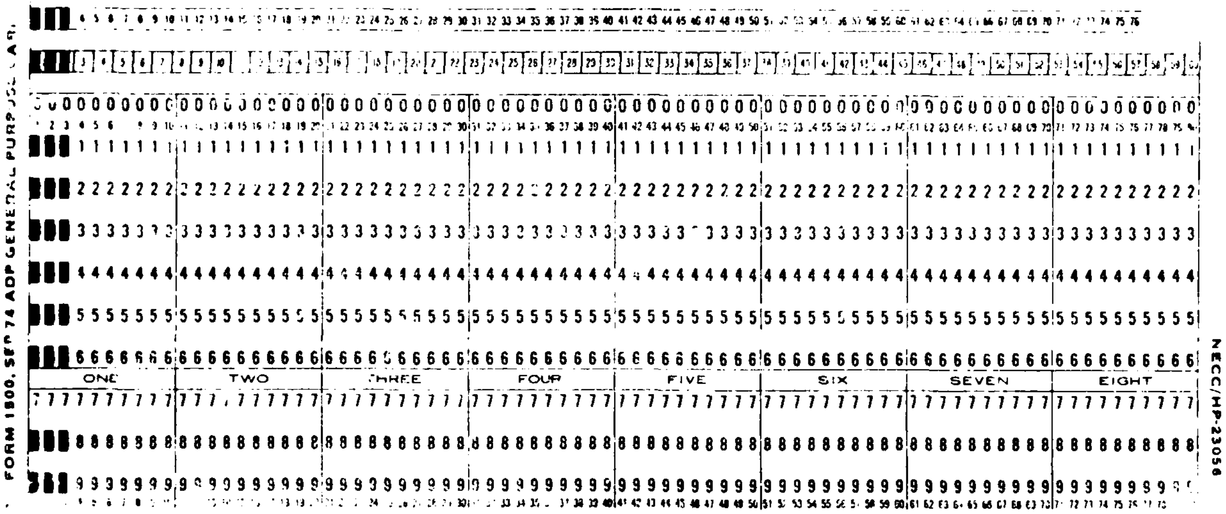
STEP 1 - Raise the top housing on the line printer and open the access door on the left side of the machine at the front. Swing the tape access door down.

STEP 2 - Relax the tension of the tape loop in the machine by gripping the locking lever fingers on either side of the H-shaped bar slide assembly and moving the assembly upward. Disengage the operational vertical format tape loop from the upper sprocket roller and slide the loop out.

STEP 3 - Insert the diagnostic test vertical format tape loop over the upper sprocket roller engaging the sprocket pins and insuring that the tape is routed between the paper tape guide and the phototransistor diode assembly. Slip the bottom of the loop over the lower sprocket wheel and gently increase the tension on the loop by gripping the locking level fingers on the bar slide assembly and moving the assembly downward. Do not put excess tension on the vertical format tape. Swing the tape access door up and close the machine.

TYPICAL TEST CARDS (CONCLUDED)

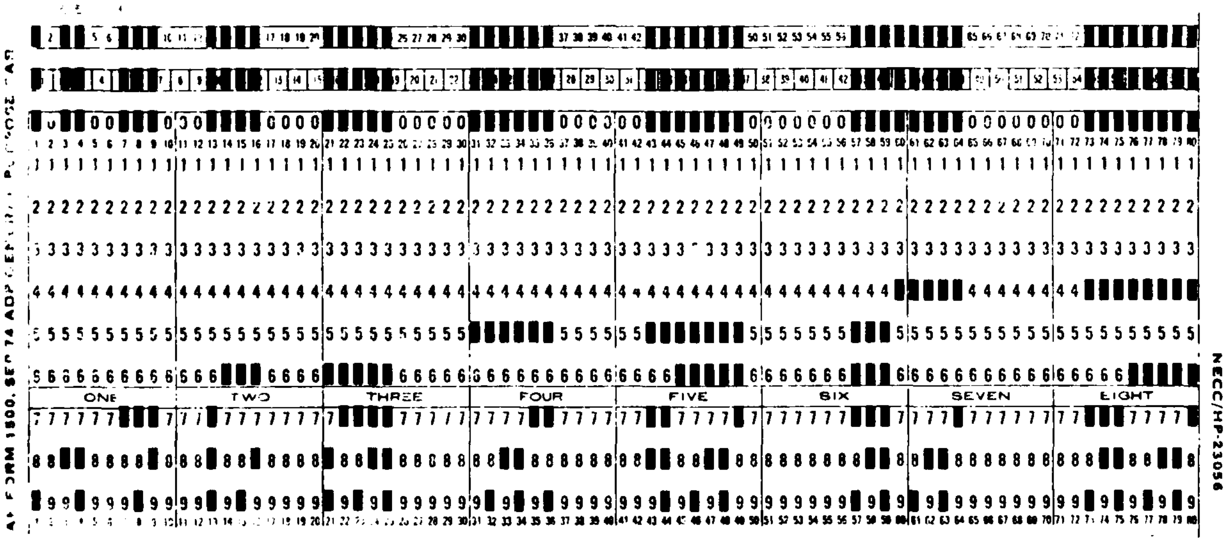
CARD #1



CARD PUNCH TEST

SIX PUNCH OUTPUT (STEP 8)

CARD #4



TYPICAL TEST CARDS

Card Reader Test

Hollerith Test Deck (STEP 4)

- 4 Blank Cards
- 1 Card punched per HSB Table
- 1 Card, Col. 1, 11-8-6; Col 2, 1-2
- 52 Blank Cards

Random Card Deck 200 Cards

AA16 - 51XXT6 Deck (64 Cards)

- 10 Cards punched per HSB Table

RCP-7XX Deck (n cards)

- 10 Cards punched per HSB Table
- Blank cards to fill out a total of 200

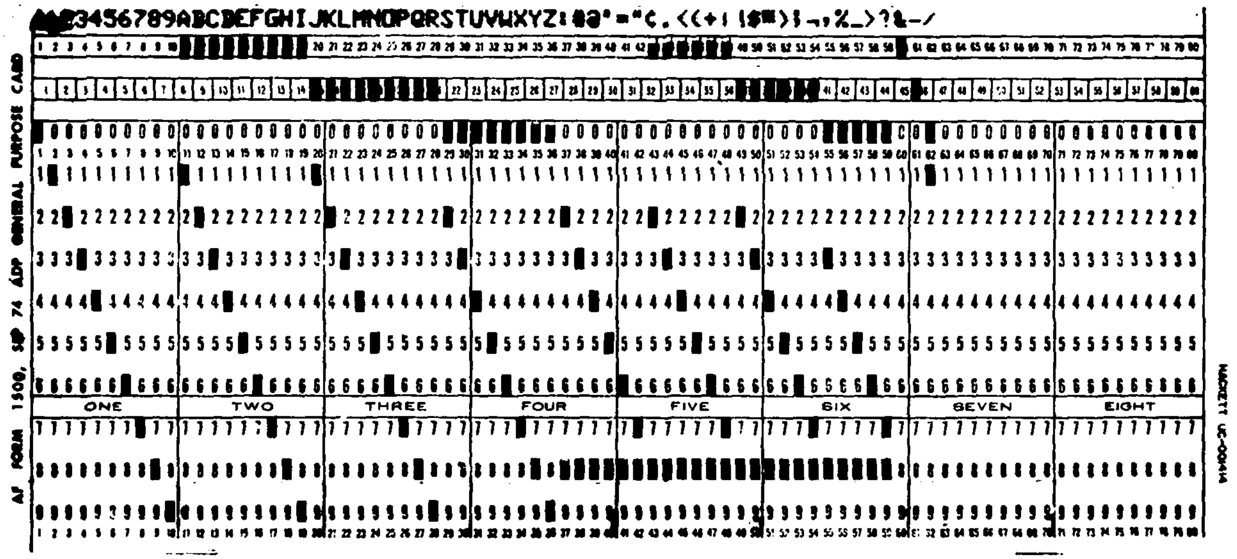

AA23456789ABCDEFGHIJKLMNOPQRSTUVWXYZ:#@'="¢.<(+|!$*);¬,%_>?&-/
AF FORM 1500, SEP 74 ADP GENERAL PURPOSE CARD
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT

EXHIBIT TAD700-6 (Cont)

CARD READER/PUNCH PROCEDURES

700-5140 CARD READER/PUNCH SUBSYSTEM

PROCEDURE - CARD DECK INPUT SOURCE

STEP 1 - Load the AB16-51XXT6 card deck using the normal program loading procedure with the boot loader. Depress STOP and RUNOUT on the card reader/punch. The following teletype message should occur:

```
AB16-51XXT6, REV D AUG 06, 74
NEXT?:
```

STEP 2 - Enter an A on the teletype. The following teletype message should occur.

```
ADR?:
```

Enter the number 6 and RETURN on the teletype. The following teletype message should occur:

```
NEXT?:
```

Enter an M on the teletype. The following teletype message should occur:

```
MSK?:
```

Enter the number 13 and RETURN on the teletype. The following teletype message should occur:

```
NEXT?:
```

Enter an S on the teletype. The following message should occur?

```
TIME 200 CARDS?:
```

Enter a Y on the teletype.
In the following steps all references to STOP, RUNOUT, and START apply to the card reader/punch switches. Exhibit TAD700-6 contains a description of the input and output test decks.

STEP 3 - Place a deck of at least 200 blank cards in the input hopper. Press START, noting the time to the second when the first card is read. Note the time to the second when the last card is read. The elapsed time should be between 27 and 33 seconds. Depress STOP and RUNOUT. The following teletype message should occur:

```
TIME OK?:
```

Enter a Y on the teletype. The following message should occur:

```
OK
NEXT?:
```

Enter an H on the teletype. The following teletype message should occur:

```
RUNOUT
```

STEP 4 - Place the Hollerith test deck in the input hopper and press START. A card should be read and the following teletype message output:

```
RUNOUT
```

Depress STOP and RUNOUT. After the card is ejected, depress START. Two more cards should be read, the RECHECK indicator should be illuminated and the following message should occur:

```
     OK
RUNOUT
```

Depress STOP and RUNOUT. After the cards are ejected, depress START. Four more cards should be read and two ejected. The RECHECK and VALIDITY indicators should be illuminated. The following teletype message should occur:

```
     OK
RUNOUT
```

Depress STOP and RUNOUT. After the two cards are ejected, depress START. The remainder of the deck will be read. Depress RUNOUT. The following teletype message should occur:

```
     OK
OFFSTK OK?:
```

Verify that only the following cards are offset in the output hopper: 1, 3, 4, 7, 8, and the last card. Enter a Y on the teletype. The following teletype message should occur:

```
     OK
NEXT?:
```

Enter a D on the teletype. The following teletype message should occur:

```
DECKSIZE NOT TO EXCEED 659 CARDS
?:
```

Enter the number 200 and RETURN on the teletype. The following teletype message should occur:

```
NEXT?:
```

STEP 5 - Enter an R on the teletype. The program will cause a carriage return on the teletype. Place the random card deck in the input hopper. Depress START. The deck should be read. Depress STOP and RUNOUT. The following teletype message should occur:

```
NEXT?:
```

Enter a V on the teletype. The following teletype message should occur:

```
FULL CHECK?:
```

Enter a Y on the teletype. The following teletype message should occur:

```
RAN.DLY?:
```

Enter a Y on the teletype. The program will cause a carriage return to occur on the teletype.

STEP 6 - Remove the random card deck from the output hopper. Replace the first card of the deck with a blank card and place the deck in the input hopper. Depress START. The first card should be read followed by the teletype message:

```
CRD NO. 1 - BITS D-98
ROW - 9, 8, 7, 6, 5, 4, 3, 2,
COL - 1, 10, 13, 16, 18, 20, 21, 24, 25, 38
```

The remainder of the deck should then be read but at an irregular rate followed by the teletype message:

```
NEXT?:
```

Depress STOP and RUNOUT. Enter a V on the teletype. The following teletype message should occur:

```
FULL CHECK?:
```

Enter a Y on the teletype. The following teletype message should occur:

```
RAN.DLY?:
```

Enter an N on the teletype. The program will cause a carriage return to occur on the teletype.

STEP 7 - Remove the random card deck from the output hopper and replace the first card (blank) with the original card. Place the random card deck in the input hopper and depress START. The deck should be read at an even rate followed by the teletype printout:

```
NEXT?:
```

Depress STOP and RUNOUT and remove the random card deck from the output hopper. The card reader test is now completed; to test the card punch, continue.

STEP 8 - Load the input hopper of the card reader/punch with at least 206 blank cards. Enter a P on the teletype. The following teletype message should occur:

```
RUNOUT
```

Depress RUNOUT and START. A card should be punched and output; the RECHECK indicator on the card reader/punch should be illuminated; and the following teletype message should occur:

```
     OK
RUNOUT
```

Depress STOP and RUNOUT. Two blank cards should be output. Depress START. A card should be punched and output and the following teletype message should occur:

```
     OK
TIME 200 CARDS?:
```

Depress STOP and RUNOUT: two more blank cards should be output. Remove the cards in the output hopper. Cards 1 and 4 should be duplicates of the standard cards shown in Exhibit TAD700-6. Cards 2, 3, 5, and 6 should be blanks.

STEP 9 - Enter a Y on the teletype. Depress START and note the time to the second when the first card is punched (this occurs almost simultaneously with the START depression). Note the time to the second when the last card is punched. The elapsed time should be between 68 and 82 seconds. The following teletype message should be output:

```
TIME OK?:
```

Enter a Y on the teletype. The following teletype message should occur:

```
     OK
RUNOUT-REMOVE STACK
```

Depress STOP and RUNOUT and remove the punched deck, which should consist of 200 cards punched all rows in columns 37-44.

COMMUNICATIONS CONTROLLER SUBSYSTEMS

700-6312 SYNCHRONOUS SINGLE LINE CONTROLLER

STEP 1 - Depress Sense Switch 4 on the CPU. Load the AA16-6312T3 card deck using the normal program-loading procedure with the boot loader. This teletype message should occur:

```
AA16-6312T3 SSLC TEST REV. A 6-29-73
COMMAND LIST:
A=AUTO TST
C=DEV ADRS
D=DEBUG
  C=CONFIG
  D=DATA
  I=IDNTFY
  L=LP TST
  N=NON-LP TST
  Q=QUIT
S=SUBTST
  DA  CID  CODE
  60  231   8 BIT CRTS
```

STEP 2 - Enter the number 60 and a RETURN on the teletype. The following teletype message should occur:

```
NEXT?:
```

STEP 3 - Enter an A on the teletype. An SSLC test should occur and cause the following teletype message in about 3 seconds:

```
PASS 00000001 DA=60 ERRORS=0000
NEXT?:
```

Any other output indicates an SSLC failure.

STEP 4 - Enter a D on the teletype. The following teletype message should occur:

```
DBG?
```

Enter a D on the teletype. The following teletype message should occur:

```
DATA?:
```

Enter the number 123456 on the teletype followed by a CARRIAGE RETURN. The following teletype message should occur:

```
DATA?:
```

Depress CARRIAGE RETURN. The following teletype message should occur:

EXHIBIT TAD700-8

DBG?:

STEP 5 - The program is now initialized to execute a loopback test on the remote communications link at any point where the transmit and receive lines are interconnected. This can be at a remote modem with a digital loopback switch, a multiplexor with loopback capability, or at a terminal board with manual interconnection. The characteristics of each communications link should be determined and the appropriate instructions added to this procedure to accomplish the desired loopback. In most cases, the first loopback will be at the remote modem.

STEP 6 - Enter an N on the teletype. A test message will be sent from the H700 (SSLC) over the communications link to the loopback point and back in several seconds. The program will verify correct message receipt which will be indicated by the same teletype message as output in STEP 3. Any other output indicates an error condition. This step should be repeated for each loopback point. After the completion of this test, all modem switches, etc., should be returned to their normal operating positions.

LOAD PROGRAM AA16-6312T1

AA16-63RT1 SSLC TEST REV. D 6-29-73

```
COMMAND LIST

  A AUTO TEST
  C RE-ENTER DEV ADOR
  D DEBUG MODE
  C CONFIG
  D DATA
  I IDENT PARMS
  L LOOP MODE TEST
  N NON-LOOP TEST
  Q QUIT DEBUG
  R RUBOUT = QUIT
  S SUBTEST #N

DA     CID   CODE
60     231   8 BIT GRTS

DA?:  60

NEXT?:  A

PASS  00000050  DA = 60  ERRORS = 0000
PASS  00000100  DA = 60  ERRORS = 0000
PASS  00000150  DA = 60  ERRORS = 0000
PASS  00000161  DA = 60  ERRORS = 0000
NEXT?:  A
PASS  00000001  DA = 60  ERRORS = 0000
NEXT^:  D
DBG?:  1
2 DBG?:
DBG?:  D
DATA?:  123456
DATA?:                        (NOTE:  PLACE MODEM IN LOCAL
DBG?:  N                       AT SITE)

PASS  00000001  DA = 60  ERRORS = 0000

          (NOTE:  AT THIS POINT PUT MODEM IN LINE AT SITE,
          CALL HOST AND HAVE THEIR MODEM PLACED ON REMOTE
          TEST.)

DBG?:  N

ERR93 @  '011435 TEST 62@ '001264
  'SB' = 160000
  'IS' = 040000
  INA 02 ='036000 INA 10 = '000000 INA 12 = '036000
  INA 01 = '043503 INA 11 = '000231 INA 13 = 040000

PASS 00000001  DA = 60  ERRORS = 0001
```

EXHIBIT TAD700-9

(NOTE: AT THIS POINT PUT MODEM IN LINE A SITE, CALL HOST AND HAVE THEIR MODEM PLACED ON LOCAL.)

```
DBG?:  N

ERR93 @ '011435  TEST 62 @ '001264

  'SB' = 160000
  'IS' = 040000

INA 02 = '036000 INA 10 = '000000 INA 12 = '036000
INA 01 =  043503 INA 11 =  000231 INA 13 = '040000

PASS 00000001 DA = 60  ERRORS = 0001
```

TERMINATE TEST AT THIS POINT AND
NOTIFY FE IN CASE OF ERRORS

EXHIBIT TAD700-9 (Cont)

TERMINAL USER (H6000-CDT) COURSE EXHIBITS

The Operation of the Teletype

Lesson TUC010 is designed to develop the skills necessary to interact with the Computer Directed Training System (CDTS) when using a teletype terminal. This lesson is aimed at those individuals who have little or no knowledge of the operation of a teletype or CDTS. This lesson will develop your skills in the following areas:

1. Logging on to a teletype terminal.

2. Accessing the Computer Directed Training System.

3. Getting a selected Computer Directed Training (CDT) lesson.

4. Correcting typographical errors.

5. Terminating a CDT lesson.

6. Interpreting error conditions.

Before advancing, you should look closely at EXHIBIT TUC010-2 which shows the location of some of the more important keys on the keyboard--specifically the CONTROL (CTRL), SHIFT, RETURN, A, X, C, AND @ keys. Locate these keys and the LINE/OFF/LOCAL switch on your teletype.

EXHIBIT TUC010-1

Keyboard SEND/RECEIVE Device
(Teletype Terminal)

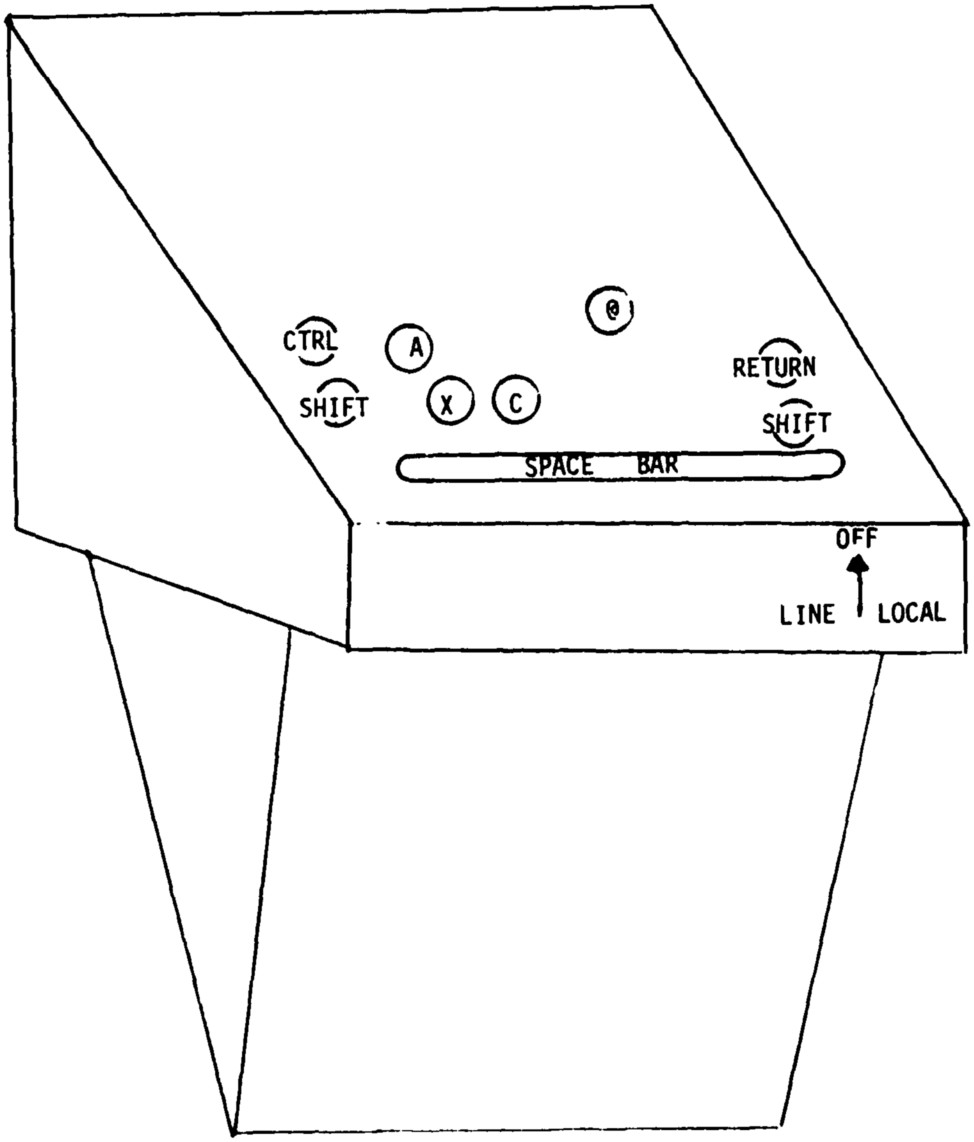


EXHIBIT TUC010-2

INSTRUCTIONS FOR BEGINNING A TRAINING SESSION

The next exhibit, TUC010-4, will show you step-by-step how to:

1. Power up a teletype.
2. Log-on to a teletype.
3. Access the Computer Directed Training System.
4. Get a CDT lesson.
5. Terminate a training session.

Read all the steps contained in TUC010-4. DO NOT attempt to activate the Computer Directed Training System until you have read all steps completely. If you have any questions, see your training monitor before beginning the sequence.

During the LOG-ON sequence, the computer system will allow you one (1) minute between required responses. If you cannot enter the correct response within the time limit, the computer system will disconnect your terminal. Once you have successfully completed your LOG-ON and accessed CDTS, you will have a ten (10) minute time limit in which to enter any required response to the course material.

In the event you should make a typographical error and notice the error prior to transmission, you must do the following:

1. Simultaneously PRESS the keys marked CTRL and X.
2. Retype the correct message.
3. PRESS the RETURN key.

If the computer system unexpectly goes down while you are under CDTS, make note of where you are in the lesson and see your training monitor. If you are aware that the computer system is going down at a given time, wait for the asterisk (*) cue from CDTS and enter ?CLOSE to terminate the lesson.

Make sure you have the following information from your training monitor before attempting to activate CDTS:

1. USER ID and PASSWORD
2. IDENT
3. In response to system prompt (*), enter - CDTS
4. Catalog/file description of your Student File
5. Catalog/file description of the Lesson File

NOTE TO TRAINING MONITOR: The Student File must be at least 2 random blocks with write permission.

If you have any questions, see your training monitor.

Should you become completely confused or lost while activating CDTS, do the following:

1. Wait until the computer is waiting for you to input a response (an asterisk).
2. Simultaneously PRESS the CTRL and C keys.
3. Secure your device according to local procedures.
4. See your training monitor.

Now read Exhibit TUC010-4.

EXHIBIT TUC010-3 (Cont)

TO GET ON/OFF THE SYSTEM

STEP	ACTION	DESCRIPTION
	Starting a Training Session	
	Powering Up the Teletype	
1	(TURN) OFF LINE LOCAL ON/OFF Switch to LINE	Teletype (TTY) is powered up.
	LOG-ON Sequence	
2	(SIMULTANEOUSLY PRESS) CTRL A	This action clears the device and starts the log-on procedure.
3	(WAIT) Message from the system PROGRAM NAME-	If nothing happens, press the RETURN key and repeat step 2.
4	(TYPE) TS1	TS1 stands for Time-sharing System. CDTS operates under the Time-sharing System.
5	(PRESS) RETURN	Pressing the RETURN key will transmit your response (message) to the computer.
6	(WAIT) Message from the system 31200 TERMINAL ID	1. Terminal information is printed.
	USERID$PASSWORD	2. USERID and PASSWORD are asked for to check and see if you are a valid System user.
	ADEHHM#%8I$%6	3. A masked line is typed.

EXHIBIT TUC010-4

STEP	ACTION	DESCRIPTION
7A	(TYPE) Your user ID and password EXAMPLE: STUDENT$CDTS (PRESS) RETURN	If you do not have an assigned user ID and password, simultaneously PRESS the CTRL and C keys to disconnect your device. Secure your device and see your training monitor.
7B	(WAIT) Message from the system IDENT?	
7C	(TYPE) Your accounting information EXAMPLE: FP7ØØ ØJ/31,3-7/JONES, (PRESS) RETURN	The accounting information you enter is the variable field of the $IDENT control card. This enables accounting of your terminal activity. The format of this information varies at every site. Check with your training monitor to get the correct information.
8	(WAIT) Message from the system CLASSIFICATION OF YOUR OUTPUT?	The system requires you to identify the security classification of output you will generate. The lesson material CDTS presents is considered output.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
9	(TYPE)	
	ZZZ	1. ZZZ stands for DO NOT MARK and is appropriate for material with no security classification.
	(PRESS)	
	RETURN	2. Some installations will require you to use UZZ so as to specifically identify the output as UNCLASSIFIED.
		3. Unless required by your facility to use UZZ, use ZZZ.
		4. Any acceptable entry other than ZZZ may cause problems in CDTS lesson presentation.
10	(WAIT)	
	Message from the system	The system requires you to identify the security classification of any files you will create. CDTS will be updating a Student File to mark your progress through a given lesson.
	CLASSIFICATION OF FILES YOU WILL CREATE?	
11	(TYPE)	
	UZZ	1. UZZ stands for UNCLASSIFIED.
	(PRESS)	2. Log-on is complete.
	RETURN	
	<u>Accessing the Computer Directed Training System</u>	
12	(WAIT)	
	(Message from the system	CDTS may now be activated.
	*	

STEP	ACTION	DESCRIPTION
13	(TYPE) CDTS (PRESS) RETURN	This places the CDT Executive into execution.
14	(WAIT)	
	Message from CDTS	1. This sequence informs you that CDTS has been activated. CDTS requires you to identify your Student File.
	CDTS STARTED	2. The asterisk (*) is the prompting cue for you to enter a response.
	STUDENT FILE- CAT/FILE STRING *	3. Refer to AFM 50-752, Attachment 5, for action to be taken if error conditions arise.
15	(TYPE)	
	Catalog/file description of your Student File	1. Enter the catalog/file description of your Student File as designated by your installation. The Student File contains information as to your progress through a lesson.
	EXAMPLE: STUDENT/JONES	
	(PRESS)	2. Type the message exactly. An incorrect entry will return you to step 14 where you may continue.
	RETURN	
16	(WAIT)	
	Message from CDTS	1. Your SSAN is part of your student identification. This is a security feature to verify your Student File before CDTS begins updating your progress.
	IAW THE PRIVACY ACT OF 1974, YOU ARE ADVISED THAT DISCLOSURE OF THIS INFORMATION IS VOLUNTARY. SEE AFM 50-752, ATTACHMENT 8, FOR MORE INFORMATION. STUDENT SSAN (9) *	
		2. Again, the asterisk (*) cues you to enter a response.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
17	(TYPE) Your SSAN EXAMPLE: 823456789 (PRESS) RETURN	1. Your SSAN must be nine (9) numerics with no spaces or dashes. 2. The student SSAN is used to output student records to the training monitor to certify completion of a course. 3. If this is the first time you have accessed CDTS, whatever entry you make for an SSAN must then be used to continue updating that Student File. In the event CDTS does not accept your SSAN entry, CDTS will terminate. To rectify this condition, the training monitor will have to erase your Student File before you can continue. 4. If you have accessed CDTS previously and your Student File is current, you will be advanced to step 26.
18	(WAIT) Optional Message from CDTS GRADE OR TITLE (6) *	This message prompts you to identify your military grade or your civilian title.
19	(TYPE) Your Grade or Title EXAMPLES: 02 GS13 PL313 E4 (PRESS) RETURN	1. This information is recorded in your Student File. 2. Your entry must not exceed (6) characters.
20	(WAIT) Optional Message from CDTS LAST NAME (30) *	This message prompts you to enter your last name.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
21	(TYPE) Your LAST Name EXAMPLE: JONES (PRESS) RETURN	1. This information is recorded into your Student File. 2. Your entry must not exceed thirty (30) characters.
22	(WAIT) Optional Message from CDTS FIRST NAME MI (30) *	This message prompts you to enter your first name and middle initial.
23	(TYPE) Your FIRST Name and MI EXAMPLE: JAMES R (PRESS) RETURN	1. This information is recorded into your Student File. 2. Your entry must not exceed thirty (30) characters.
24	(WAIT) Optional Message from CDTS ORGANIZATION (24) *	This message prompts you to identify your organization.
25	(TYPE) Your organization Example: 3390 TCHTG (PRESS) RETURN	1. This information is recorded into your Student File. 2. Your entry must not exceed twenty-four (24) characters.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
26	(WAIT)	
	Message from CDTS LESSON FILE- CAT/FILE STRING *	CDTS is now ready to begin a Training Session. Before CDTS can present a lesson, it requires the location of the CDT lessons.
27	(TYPE)	
	Catalog/File Description of the CDT Lesson File	1. Enter the catalog/file description of the Lesson File. Type the message exactly.
	EXAMPLE: CDTSCDTS/TUC/LESSONS	2. The Lesson File contains the CDT lessons.
	(PRESS) RETURN	3. An incorrect entry will return you to step 26 where you may continue.

Getting a Lesson

STEP	ACTION	DESCRIPTION
28	(WAIT)	
	Message from CDTS ENTER COMMAND *	CDTS has selected a Lesson File and is ready to begin a lesson.
29	(TYPE)	
	?GET LLLLLL NEW or ?GET LLLLLL OLD or ?GET LLLLLL	1. LLLLLL is the six (6) position lesson name of a desired lesson. To retrieve lesson TUC010, LLLLLL would be replaced with TUC010.
	EXAMPLE: ?GET TUC010 NEW	2. NEW is to be specified if you are entering a lesson for the first time.
	(PRESS) RETURN	3. OLD is the default option and is appropriate if you are returning to a lesson that was begun and terminated before the lesson was completed.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
30	(WAIT)	
	MESSAGE FROM CDTS	1. CDTS informs you that it has selected the requested lesson.
	LLLLLL READY ENTER COMMAND *	2. LLLLLL is replaced by the six (6) position lesson name.
31	(TYPE)	
	?GOTO * (PRESS) RETURN	Referring to step 29, ?GOTO * will start a NEW lesson at the beginning and restart an OLD lesson at the point at which the lesson was concluded if the lesson was incomplete.
32	(WAIT) (READ) Course material will begin printing on your device.	1. Follow the course material closely. Read all information presented. 2. CDT lessons require a great deal of student interaction and participation. Various types of questions will be asked and you will be required to respond to the questions. 3. When CDTS presents a question, it will allow you time to transmit your answer. The asterisk (*) cue will prompt you when CDTS is awaiting a response. When the asterisk (*) appears, go on to step 33. 4. As CDTS presents material and as you respond to the course material, you will advance through the lesson. At some point, you will reach the end of a lesson. When you reach the end of a lesson, CDTS will display the message "LESSON ENDED." When this occurs, go on to step 34.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
33	(TYPE)	
	Appropriate response to course material (PRESS RETURN after each response)	1. When the asterisk (*) cue appears, transmit your response to the course material. 2. Continue with steps 32 and 33 until you are ready to discontinue a lesson or until the lesson has concluded. 3. To break a lesson when the lesson has not been completed, go on to step 35.
34	(LESSON ENDED)	
	Message from CDTS LESSON ENDED ENTER COMMAND *	1. This sequence occurs when a lesson has been concluded in its entirety. You should nave reached this level from step 32, paragraph 4. 2. At this level, you may return to step 29 and select another lesson, or you may go on to step 38 and conclude the training session.
	Terminating a Lesson	
35	(WAIT)	
	Message from CDTS *	1. The asterisk (*) is your cue to enter a response. You should have reached this point from step 33. 2. At this point, CDTS is awaiting a response to course material; however, CDTS will respond to certain commands. To terminate a lesson when the lesson is incomplete, go on to step 36.
36	(TYPE) ?DONE (PRESS) RETURN	By entering ?DONE following the CDTS cue in response to a question within a lesson, CDTS will record your progress in your Student File and mark your location in your current lesson. CDTS will then restart the respective lesson at that location when you access the respective lesson at a later date.

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
	Terminating/Continuing a Training Session	
37	(WAIT)	
	Message from CDTS	1. LLLLLL will be replaced by the Lesson name.
	LLLLLL READY ENTER COMMAND *	2. At this level, you have several options. First, you may return to step 31 and continue with your current lesson. Second, you may return to step 29 and select another lesson. Third, you may go on to step 38 and terminate the training session.
38	(TYPE)	
	?CLOSE	The command ?CLOSE will terminate your training session.
	(PRESS)	
	RETURN	
39	(WAIT)	
	Message from CDTS	1. CDTS informs you that it has terminated.
	RECORDS SPAWNED 9999T CDTS ENDED	2. The message "RECORDS SPAWNED" is printed to inform you that your Student Records are being processed. The number 9999T is called a "SNUMB" and is used by the System to identify that activity. 3. Your Student Records are processed and sent to your training monitor. Your training monitor will use these records to identify problem areas or to certify completion of a course.
40	(WAIT)	CDTS has returned you to the Time-sharing System.
	Message from the system	
	*	

EXHIBIT TUC010-4 (Cont)

STEP	ACTION	DESCRIPTION
41	(TYPE)	
	BYE	BYE is the command to terminate your terminal activity.
	(PRESS)	
	RETURN	
42	(WAIT)	
	Message from the system	
	** RESOURCES USED......... **TIME-SHARING OFF AT.....	
43	(TURN)	
	OFF	1. Secure your device according to local convention.
	LINE LOCAL	2. Save course material for future reference.
	Turn teletype to OFF	

EXHIBIT TUC010-4 (Cont)

HANDOUT DEMONSTRATING

INITIALLY ACTIVATING CDTS

```
0312000

PROGRAM NAME - TS1
312000
TERMINAL ID T1

USERID$PASSWORD
ADEHLH$$$$¢&
IDENT?
FP700 0J/31,3-7/NNNNN,                (See your TM for local ident)

CLASSIFICATION OF YOUR OUTPUT?
ZZZ
CLASSIFICATION OF FILES YOU WILL CREATE?
UZZ

*CDTS

CDTS STARTED

STUDENT FILE-
CAT/FILE STRING
*

STUDENT/SMITH

IAW THE PRIVACY ACT OF 1974, YOU ARE ADVISED THAT
DISCLOSURE OF THIS INFORMATION IS VOLUNTARY
SEE AFM 50-752, ATTACHMENT 8, FOR MORE INFORMATION
STUDENT SSAN (9)
*
823456789

GRADE OR TITLE (6)
*
02

LAST NAME (30)
*
SMITH

FIRST NAME MI (30)
*
JOHN E

ORGANIZATION (24)
*
3890 TCHTG
```

EXHIBIT TUC010-5

```
LESSON FILE-
CAT/FILE STRING
*
CDTSCDTS/TUC/LESSONS
ENTER COMMAND
*
?GET TUC010 NEW
TUC010 READY
ENTER COMMAND
*
?GOTO *


                              COMPUTER
                          DIRECTED TRAINING

CONGRATULATIONS ON ESTABLISHING CONTACT WITH ME AND WELCOME TO

TUC010, THE FIRST CDT LESSON IN THE TERMINAL USER'S COURSE.  THIS

LESSON WILL TEACH YOU THE OPERATING PROCEDURES FOR A TELETYPE

TERMINAL AND WILL TAKE ABOUT 45 MINUTES TO COMPLETE.


IS THIS WHAT YOU HAD EXPECTED FROM THIS LESSON?  (YES OR NO)


*
?CLOSE
RECORDS SPAWNED 6817T
CDTS ENDED


*BYE
**RESOURCES USED $  1.22, USED TO DATE $    243.51 = 5%
** TIME-SHARING OFF AT 10.217 ON  8/88/82
```

EXHIBIT TUC010-5 (Cont)

THE OPERATION OF THE 786 VISUAL INFORMATION PROJECTION (VIP)

Lesson TUC011 is designed to develop the skills necessary to interact with the Computer Directed Training System (CDTS) when using a Visual Information Projection (VIP) device. This lesson is aimed at those individuals who have little or no knowledge of the operation of a VIP and have limited knowledge of the operation of CDTS. This lesson will develop your skills in the following areas:

1. Logging on to a VIP.

2. Accessing the Computer Directed Training System.

3. Getting a Computer Directed Training (CDT) lesson.

4. Correcting typographical errors.

5. Terminating a CDT lesson.

6. Interpreting error conditions.

Lesson TUC011 follows lesson TUC010; however, TUC010 is not a prerequisite to TUC011. Individuals who have completed TUC010 will not receive any redundant information in the lesson presentation of TUC011. If the individual has completed TUC010, TUC011 will present only the information necessary to develop proficiency in the operation of a VIP.

Before advancing to EXHIBIT TUC011-3, look closely at EXHIBIT TUC011-2 and locate some of the more important keys on the keyboard--specifically the PRINT/TRANSMIT (PRT TX), LINE RETURN (LR), FORM FEED (FF), END OF TEXT (ETX), PAGE RETURN (PR), REPEAT (RPT), SHIFT, FORWARD SPACE (FS), BACK SPACE (BS), LINE FEED (LF), REVERSE LINE FEED (RLF), and the four (4) ARROW keys.

EXHIBIT TUC011-1

VISUAL INFORMATION PROJECTION
(VIP) DEVICE 786

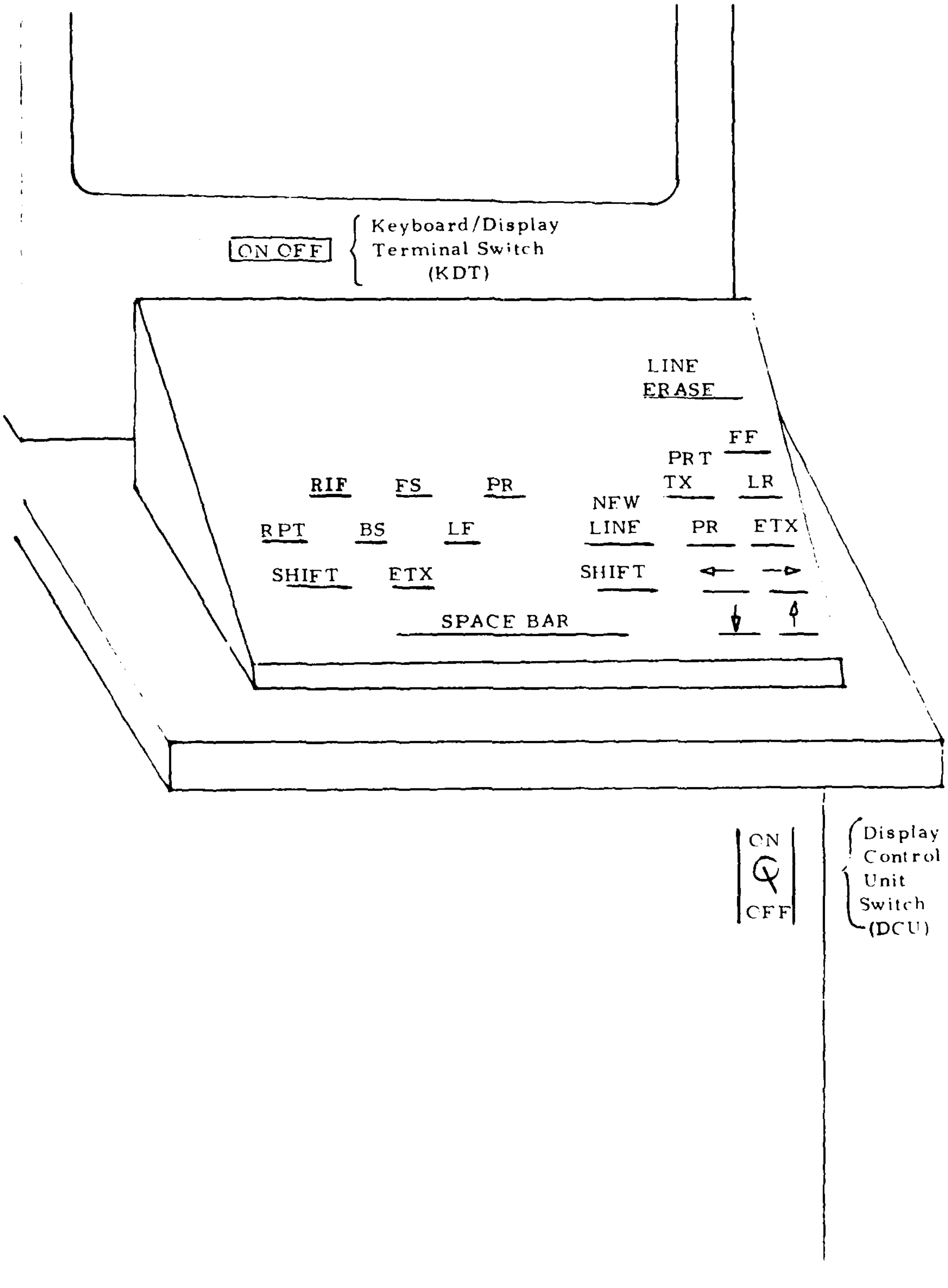


EXHIBIT TUC011-2

INSTRUCTIONS FOR BEGINNING A TRAINING SESSION

The next exhibit, TUC011-4, will show you step-by-step how to:

1. Power up a VIP.

2. Log-on to a VIP.

3. Access the Computer Directed Training System.

4. Get a CDT lesson.

5. Terminate a training session.

Read all steps contained in TUC011-4. DO NOT attempt to activate the Computer Directed Training System until you have read all steps completely. If you have any questions, see your training monitor before beginning the sequence.

During the LOG-ON sequence, the computer will allow you one (1) minute between required responses. If you cannot enter the correct response within the time limit, the computer will disconnect your terminal. Once you have successfully completed your LOG-ON and accessed CDTS, you will have a ten (10) minute time limit in which to enter any required response to course material.

In the event you should make a typographical error and notice the error prior to transmission, you must do the following:

1. PRESS the key marked NEW LINE.

2. Retype the correct message.

3. PRESS the keys ETX LR PRT/TX in sequence.

If the computer system goes down unexpectedly while you are under CDTS make note of where you are in the lesson and see your training monitor. If you are aware that the computer system is going down at a given time, wait for the asterisk (*) cue from CDTS and enter ?CLOSE to terminate the lesson.

Make sure you have the following information from your training monitor before attempting to activate CDTS:

1. USERID and PASSWORD
2. IDENT
3. In response to system prompt (*), enter CDTS.
4. Catalog/file description of your Student File
5. Catalog/file description of the Lesson File

NOTE TO TRAINING MONITOR: The Student File must be at least 2 random blocks with write permission.

If you have any questions, see your training monitor.

EXHIBIT TUC011-3

Should you become completely confused or lost while activating CDTS do the following:

1. Wait until the computer is waiting for you to input a response.

2. Type $*$DIS

3. PRESS ETX LR PRT/TX keys in sequence.

4. Secure your device.

5. See your training monitor. While using a VIP you will not have a hard copy output to show your confusion, so recall the events accurately.

IMPORTANT NOTE TO TRAINEE: When the symbols ^* or *^ appear, do the following: 1) PRESS FF key, and 2) PRESS PRT/TX key.

Now read EXHIBIT TUC011-4.

TO GET ON/OFF THE SYSTEM

STEP	ACTION	DESCRIPTION
	Starting a Training Session	
	Powering Up the 786 Series VIP Keyboard Display Station (KDS)	
1	(FLIP UP) DISPLAY CONTROL UNIT switch (DCU) to the ON position.	1. Refer to Exhibit TUC011-2 for the location of the switch. 2. The DCU is powered up. 3. Wait a few seconds before moving on to Step 2.
2	(PRESS) KEYBOARD/DISPLAY TERMINAL switch (KDT) to the ON position.	1. Refer to TUC011-2 for the location of the switch. 2. The KDT is powered up. 3. Wait until blinking cursor appears clearly on the screen.
3	(WAIT) A blinking cursor should appear. Be patient.	Blinking cursor indicates the device is ready to accept your messages.
4	(PRESS) FF key	This action will insure cursor is positioned at the upper lefthand corner of the screen.
5	(TYPE) $*$DIS	1. The keyboard of the VIP operates similarly to the keyboard of a typewriter. 2. This message, in conjunction with the next step, will insure the device is ready for the LOG-ON sequence.

EXHIBIT TUC011-4

STEP	ACTION	DESCRIPTION
6	(PRESS) ETX LR PRT/TX keys in sequence.	1. Observe the events that occur as you press each key. 2. The ETX key will generate a character that marks the end of text. 3. The LR key positions the cursor to the left hand side of the screen and marks the start of the text. 4. The PRT/TX key will transmit your message to the computer.

NOTE: Each time you TYPE a message, you must PRESS the ETX LR PRT/TX keys in sequence to transmit your response to the computer. In the steps to follow, (TRANSMIT) will mean PRESS the ETX LR PRT/TX keys in sequence.

STEP	ACTION	DESCRIPTION
7	(WAIT) Message from the system LINE TERMINATED — INC	The VIP is now cleared and ready to begin the LOG ON sequence.
8	(PRESS) FF key	The screen will be cleared and the cursor repositioned at the upper left-hand corner of the screen.
9	(TYPE) $*$LOG22L, TS1 (TRANSMIT)	This message will call up the Time-sharing system. CDTS operates under TS1.
10	(WAIT) Message from the system 312000 TERMINAL ID USERID$PASSWORD	1. Terminal information is presented. 2. The question USERID$PASSWORD is asked to check to see if you are a valid system user. 3. The question USERID$PASSWORD may be asked more than once depending upon local procedures.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
11	(TYPE) Your USERID and PASSWORD Example: STUDENT$CDTS (TRANSMIT)	Be sure to enter your USERID$PASSWORD correctly because an incorrect response will result in a security breach. If you do not have one, turn OFF the two switches you just turned on and go see your Training Monitor.
12A	(WAIT) Message from the system IDENT?	
12B	(TYPE) Your accounting information Example: FP7ØØ ØJ/31,3-7/SMITH, (TRANSMIT)	The information you enter is the variable field of a $ IDENT control card. This information will enable an accounting of your terminal activity. The format of this information varies at every base. Check with your Training Monitor to get the correct information.
13	(WAIT) Message from the system CLASSIFICATION OF YOUR OUTPUT?	The system requires you to identify the security classification of output you will generate. The lesson material CDTS presents is considered output.
14	(TYPE) ZZZ (TRANSMIT)	
15	(WAIT) Message from the system CLASSIFICATION OF FILES YOU WILL CREATE?	The system requires you to identify the security classification of any files you will create. CDTS will be updating a Student File to mark your progress through a given lesson.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
16	(TYPE) UZZ (TRANSMIT)	1. UZZ stands for UNCLASSIFIED. 2. LOG ON is complete.

Accessing the Computer Directed Training System

STEP	ACTION	DESCRIPTION
17	(WAIT) Message from the system *	CDTS may now be activated.
18	(TYPE) CDTS (TRANSMIT)	This places the CDT Executive into execution.
19	(WAIT) Message from CDTS CDTS STARTED STUDENT FILE - CAT/FILE STRING *	1. This sequence informs you that CDTS has been activated. CDTS requires you to identify your Student File. 2. The asterisk (*) is the prompting cue for you to enter a response. 3. Refer to AFM 50-752, Attachment 5, for action to be taken if error conditions arise.
20	(TYPE) Catalog/file description of your Student File Example: STUDENT/SMITH (TRANSMIT)	1. Enter the catalog/file description of your Student File as designated by your installation. The Student File contains information as to your progress through a lesson. 2. Type the message exactly. An incorrect entry will return you to step 19 where you may continue.
21	(WAIT) Message from CDTS	1. Your SSAN is part of your student identification. This is a security feature to verify your Student File before CDTS begins updating your progress.

```
IAW THE PRIVACY ACT OF 1974, YOU ARE ADVISED THAT
DISCLOSURE OF THIS INFORMATION IS VOLUNTARY
SEE AFM 50-752, ATTACHMENT 8, FOR MORE INFORMATION
STUDENT SSAN (9)
*
```

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
		2. Again, the asterisk (*) cues you to enter a response.
22	(TYPE) Your SSAN Example: 823456789 (TRANSMIT)	1. Your SSAN must be nine (9) numerics with no spaces or dashes. 2. The student SSAN is used to output student records to the training monitor to certify completion of a course. 3. If this is the first time you have accessed CDTS, whatever entry you make for an SSAN must then be used to continue updating that Student File. In the event CDTS does not accept your SSAN entry, CDTS will terminate. To rectify this condition, the training monitor will have to ERASE your Student File before you can continue. 4. If you have accessed CDTS previously and your Student File is current, you will be advanced to step 31.
23	(WAIT) Optional Message from CDTS GRADE OR TITLE (6) *	This message prompts you to identify your military grade or your civilian title.
24	(TYPE) Your Grade or Title Examples: 02 GS13 PL313 E4 (TRANSMIT)	1. This information is recorded in your Student File. 2. Your entry must not exceed six (6) characters.
25	(WAIT) Optional Message from CDTS LAST NAME (30) *	This message prompts you to enter your last name.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
26	(TYPE) Your LAST Name Example: SMITH (TRANSMIT)	1. This information is recorded into your Student File. 2. Your entry must not exceed thirty (30) characters.
27	(WAIT) Optional Message from CDTS FIRST NAME MI (30) *	This message prompts you to enter your first name and middle initial.
28	(TYPE) Your FIRST Name and MI Example: SAMUEL I. (TRANSMIT)	1. This information is recorded into your Student File 2. Your entry must not exceed thirty (30) characters.
29	(WAIT) Optional Message from CDTS ORGANIZATION (24) *	This message prompts you to identify your organization.
30	(TYPE) Your Organization Example: 3392 SCHS (TRANSMIT)	1. This information is recorded into your Student File. 2. Your entry must not exceed twenty-four (24) characters.
31	(WAIT) Message from CDTS LESSON FILE - CAT/FILE STRING *	CDTS is now ready to begin a Training Session. Before CDTS can present a lesson, it requires the location of the CDT lessons.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
32	(TYPE) Catalog/file description of the CDT lesson file Example: CDTSCDTS/TUC/LESSONS (TRANSMIT)	1. Enter the catalog/file description of the Lesson File as designated by your installation. Type the message exactly. 2. The Lesson File contains the CDT lessons. 3. An incorrect entry will return you to step 31 where you may continue.
	Getting a Lesson	
33	(WAIT) Message from CDTS ENTER COMMAND *	CDTS has selected a Lesson File and is ready to begin a lesson.
34	(TYPE) ?GET LLLLLL NEW or ?GET LLLLLL OLD or ?GET LLLLLL Example: ?GET TUC011 NEW (TRANSMIT)	1. LLLLLL is the six (6) position lesson name of a desired lesson. To retrieve lesson TUC011, LLLLLL would be replaced with TUC011. 2. NEW is to be specified if you are entering a lesson for the first time. 3. OLD is the default option and is appropriate if you are returning to a lesson that was begun and terminated before the lesson was completed.
35	(WAIT) Message from CDTS LLLLLL READY ENTER COMMAND *	1. CDTS informs you that it has selected the requested lesson. 2. LLLLLL is replaced by the six (6) position lesson name.
36	(TYPE) ?GOTO * (TRANSMIT)	Referring to step 34, ?GOTO * will start a "NEW" lesson at the beginning and restart an "OLD" lesson at the point at which the lesson was concluded if the lesson was incomplete.

EXHIBIT TUC011-4 (Cont)

STEP **ACTION** **DESCRIPTION**

37 (WAIT) (READ)

Course material will begin printing on your device.

1. Follow the course material closely. Read all information presented.

2. CDT lessons require a great deal of student interaction and participation. Various types of questions will be asked and you will be required to respond to the questions.

3. When CDTS presents a question, it will allow you time to transmit your answer. The nonblinking asterisk (*) cue will prompt you when CDTS is awaiting a response. When the asterisk (*) appears, go on to step 38.

4. As CDTS presents material and as you respond to the course material, you will advance through the lesson. At some point you will reach the end of a lesson. When you reach the end of a lesson, CDTS will display the message "LESSON ENDED." When this occurs, go on to step 39.

38 (TYPE)

Appropriate response to course material

(TRANSMIT after each response)

1. When the asterisk (*) cue appears, transmit your response to the course material.

2. Continue with steps 37 and 38 until you are ready to discontinue a lesson or until the lesson has concluded.

3. To break a lesson, when the lesson has not been completed, go on to step 40.

39 (LESSON ENDED)

Message from CDTS

LESSON ENDED

ENTER COMMAND
*

1. This sequence occurs when a lesson has been concluded in its entirety. You should have reached this level from step 37, paragraph 4.

2. At this level you may return to step 34 and select another lesson, or you may go on to step 43 and conclude the training session.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
	Terminating a Lesson	
40	(WAIT) Message from CDTS *	1. The nonblinking asterisk (*) is your cue to enter a response. You should have reached this point from step 38. 2. At this point CDTS is awaiting a response to course material; however, CDTS will respond to certain commands. To terminate a lesson when the lesson is incomplete, go on to step 41.
41	(TYPE) ?DONE (TRANSMIT)	By entering ?DONE following the CDTS cue in response to a question within a lesson, CDTS will record your progress in your Student File and mark your location in your current lesson. CDTS will then restart the respective lesson at that location when you access the respective lesson at a later date.
	Terminating/Continuing a Training Session	
42	(WAIT) Message from CDTS LLLLLL READY ENTER COMMAND *	1. LLLLLL will be replaced by the lesson name. 2. At this level, you have several options. First, you may return to step 36 and continue with your current lesson. Second, you may return to step 34 and select another lesson. Third, you may go on to step 43 and terminate the training session.
43	(TYPE) ?CLOSE (TRANSMIT)	The command ?CLOSE will terminate your training session.

EXHIBIT TUC011-4 (Cont)

STEP	ACTION	DESCRIPTION
44	(WAIT) Message from CDTS RECORDS SPAWNED 9999T CDTS ENDED	1. CDTS informs you it has terminated. 2. The message "RECORDS SPAWNED" is printed to inform you that your Student Records are being processed. The number 9999T is called a SNUMB and is used by the System to identify that activity. 3. Your Student Records are processed and sent to your training monitor. Your training monitor will use these records to identify problem areas or to certify completion of a course.
45	(WAIT) Message from the system *	CDTS has returned you to the Time-sharing System.
46	(TYPE) BYE TRANSMIT	BYE is the command to terminate your terminal activity.
47	(WAIT) Message from the system RESOURCES USED........... TIME-SHARING OFF AT......	
48	(PRESS) KDT switch OFF (FLIP DOWN) DCU switch OFF	Secure your device according to local convention.

EXHIBIT TUC011-4 (Cont)

THE OPERATION OF THE HONEYWELL SERIES 7700 VISUAL INFORMATION PROJECTION (VIP)

Lesson TUC012 is designed to develop the skills necessary to interact with the Computer Directed Training System (CDTS) when using a Honeywell series 7700 Visual Information Projection (VIP) device. This lesson is aimeu at those individuals who have little or no knowledge of the operation of a VIP and have limited knowledge of the operation of CDTS. This lesson will develop your skills in the following areas:

1. Logging on to a VIP.

2. Accessing the Computer Directed Training System.

3. Getting a Computer Directed Training (CDT) lesson.

4. Correcting typographical errors.

5. Terminating a CDT lesson.

6. Interpreting error conditions.

Lesson TUC012 follows lessons TUC010 and TUC011; however, TUC010 and TUC011 are not prerequisites to TUC012. Individuals who have cumpleted TUC010 and/or TUC011 will not receive redundant information in the lesson presentation of TUC012. If the individual has completed TUC010 and/or TUC011, TUC012 will present only the information necessary to develop proficiency in the operation of the series 7700 VIP.

Before advancing to EXHIBIT TUC012-3, look closely at EXHIBIT TUC012-2 and locate some of the more important keys on the keyboard--specifically the TRANSMIT, LINE RETURN, HOME, CLEAR, SHIFT, CONTROL (CTRL), NEW LINE, LOCK, PRINT, LINE CLEAR (LINE CLR), CHARACTER INSERT and DELETE (CHAR I/D), LINE INSERT and DELETE (LINE I/D) and the four directional arrows.

EXHIBIT TUC012-1

Series 7700 Visual Information Projection (VIP) keyboard

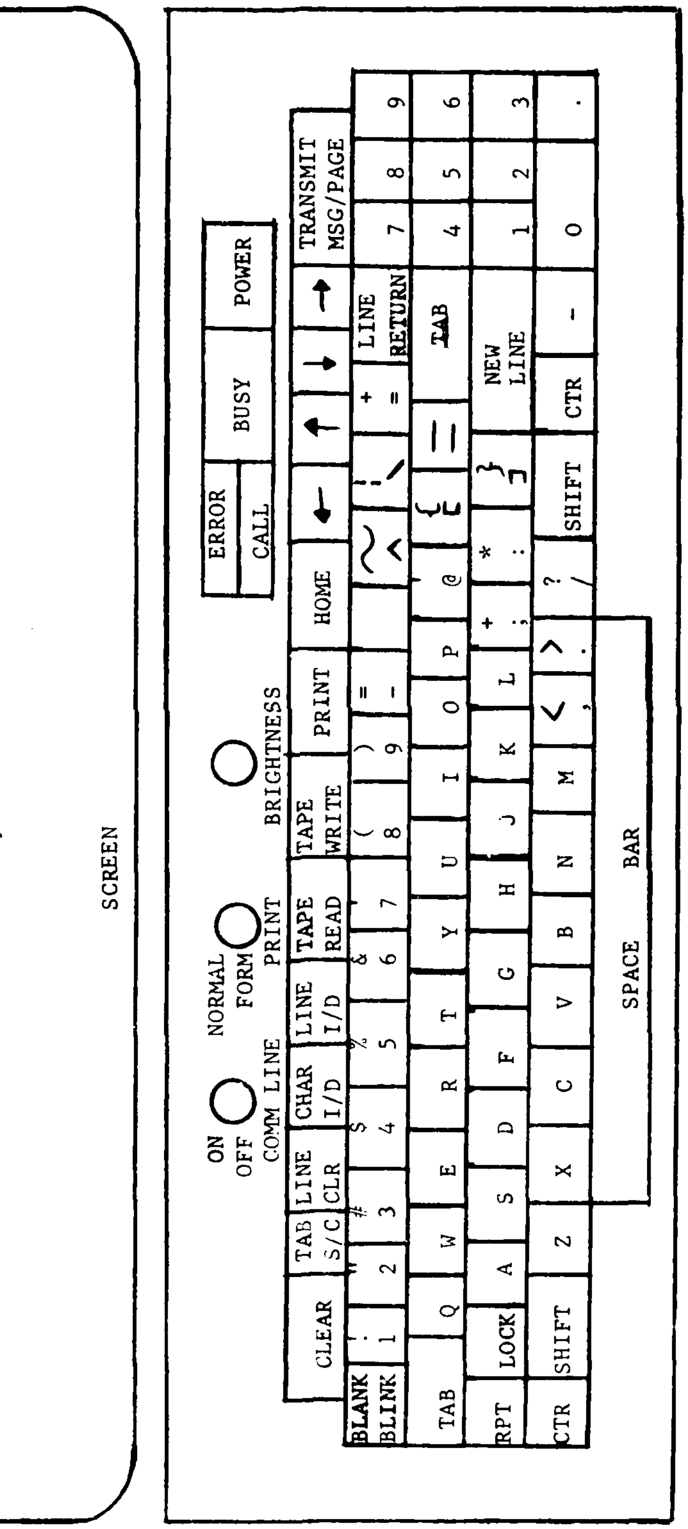


EXHIBIT TUC012-2

INSTRUCTIONS FOR BEGINNING A TRAINING SESSION

The next exhibit, TUC012-4, will show you step-by-step how to:

1. Power up a series 7700 VIP Keyboard Display Station (KDS).

2. Log-on to a series 7700 VIP.

3. Access the Computer Directed Training System.

4. Get a CDT lesson.

5. Terminate a training session.

Read all steps contained in TUC012-4. DO NOT attempt to activate the Computer Directed Training System until you have read all steps completely. If you have any questions, see your Training Systems Monitor before beginning the sequence.

During the LOG-ON sequence, the computer will allow you one (1) minute between required responses. If you cannot enter the correct response within the time limit, the computer will disconnect your terminal. Once you have successfully completed your LOG-ON and accessed CDTS, you will be required to respond to course material. Your response to course material is time limited also, and this time limit is set by your local facility (generally, 10 minutes).

In the event you shou'd make a typographical error and notice the error prior to transmission, y' ust do the following:

1. PRESS the CLEAR key.

2. RETYPE the correct message.

3. PRESS the TRANSMIT key.

If the computer system goes down unexpectedly while you are under CDTS make note of where you are in the lesson and see your Training Systems Monitor. If you are aware that the computer system is going down at a given time, wait for the asterisk (*) cue from CDTS in response to a question and enter ?CLOSE to terminate the lesson and CDTS.

Make sure you have the following information from your Training Systems Monitor before attempting to activate CDTS:

1. USERID and PASSWORD
2. IDENT
3. In response to system prompt (*), enter CDTS.
4. Catalog/file description of your Student File.
5. Catalog/file description of the Lesson File.

EXHIBIT TUC012-3

NOTE TO TRAINING MONITOR: The Student File must be at least 2 random blocks with write permission.

If you have any questions, see your training monitor.

Should you become completely confused or lost while activating CDTS do the following:

1. Wait until the computer is waiting for you to input a response (nonblinking asterisk cue).

2. Type $*$DIS

3. PRESS the TRANSMIT key.

4. Secure your device.

5. See your Training System Monitor. While using a VIP you will not have a printout to document the events leading to your confusion, so recall the events accurately.

IMPORTANT NOTE TO TRAINEE: When the following symbols appear *|| (a blinking asterisk immediately followed by a blinking cursor, do the following:

1. TYPE a blank (one space).

2. PRESS the TRANSMIT key.

Now read EXHIBIT TUC012-4.

NOTE TO TRAINING MONITOR: The Student File must be at least 2 random blocks with write permission.

If you have any questions, see your training monitor.

Should you become completely confused or lost while activating CDTS do the following:

1. Wait until the computer is waiting for you to input a response (nonblinking asterisk cue).

2. Type $*$DIS

3. PRESS the TRANSMIT key.

4. Secure your device.

5. See your Training System Monitor. While using a VIP you will not have a printout to document the events leading to your confusion, so recall the events accurately.

IMPORTANT NOTE TO TRAINEE: When the following symbols appear *|| (a blinking asterisk immediately followed by a blinking cursor, do the following:

1. TYPE a blank (one space).

2. PRESS the TRANSMIT key.

Now read EXHIBIT TUC012-4.

TO GET ON/OFF THE SYSTEM

STEP	ACTION	DESCRIPTION
	Starting a Training Session	
	Powering Up the 7700 Series VIP Keyboard Display Station (KDS)	
1	(PUSH) POWER button to ON position on the VIP. (Some model VIPs have a power switch located on the upper left rear side of the terminal.)	1. The power button is labeled POWER and located just beneath the screen on the right side of the VIP. 2. The VIP is powered up. 3. Seven zeros will appear down the center of the screen as the device performs automatic diagnostics. If any of the seven positions appear as an X, contact the Field Engineer. Do NOT continue.
2	(FLIP UP) COMM LINE switch to the ON position.	The device is now linked to the computer and communications may begin. (NOTE: The COMM line switch will normally be in ON position.)
3	(PRESS) CLEAR key.	The screen is cleared and the blinking cursor appears at the top of the screen.
4	(TYPE) $*$DIS	The keyboard on a VIP operates similarly to that of a typewriter. Messages may be transmitted in either upper or lowercase letters. Uppercase letters are typed while depressing the SHIFT key.
5	(PRESS) TRANSMIT key	1. The TRANSMIT key will send your message to the computer. 2. Notice the BUSY light located beneath the screen. This light will flash as communications take place between the computer and your terminal.

NOTE: Each time you TYPE a message you must PRESS the TRANSMIT KEY to transmit your response to the computer.

EXHIBIT TUC012-4

STEP	ACTION	DESCRIPTION
6	(WAIT) Message from the system LINE TERMINATED -- INC	The VIP is now cleared and ready to begin the LOG ON sequence.
7	(PRESS) CLEAR key	The screen will be cleared and the cursor repositioned at the upper lefthand corner of the screen.
8	(TYPE) $*$LOG22L,TS1 (TRANSMIT)	This message will call up the Time-sharing System. CDTS operates under TS1.
9	(WAIT) Message from the system 312000 TERMINAL ID	Terminal information is presented.
10	(WAIT) Message from system USERID$PASSWORD	1. USERID and PASSWORD are asked for to check and see if you are a valid system user. 2. This message may be presented more than once depending upon local procedures.
11A	(TYPE) Your USER ID and PASSWORD Example: STUDENT$CDTS (TRANSMIT)	Be sure to type your USERID and PASSWORD exactly, as an incorrect entry may cause a security breach and a lockout from the system. If you do not have one, push the POWER button to OFF and see your training monitor.
11B	(WAIT) Message from the system IDENT?	

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
11C	(TYPE) Your Accounting information Example: FP7ØØ ØJ/31,3-7/CLARK, (TRANSMIT)	The information you enter is the variable field of the $ IDENT control card. This enables an accounting of your terminal activity. The format of this information varies at every base. Check with your training monitor to get the correct information.
12	(WAIT) Message from the system CLASSIFICATION OF YOUR OUTPUT?	The system requires you to identify the security classification of the output you will generate. The lesson material CDTS presents is considered output.
13	(TYPE) ZZZ (TRANSMIT)	1. ZZZ stands for DO NOT MARK and is appropriate for material with no security classification. 2. Some installations will require you to use UZZ so as to specifically identify the output as UNCLASSIFIED. 3. Unless REQUIRED by your facility to use UZZ, use ZZZ. 4. Any acceptable entry other than ZZZ may cause problems in CDTS lesson presentation.
14	(WAIT) Message from the system CLASSIFICATION OF FILES YOU WILL CREATE?	The system requires you to identify the security classification of any files you will create. CDTS will be updating a Student File to mark your progress through a given lesson.
15	(TYPE) UZZ	1. UZZ stands for UNCLASSIFIED. 2. LOG ON is complete.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
	Accessing the Computer Directed Training System	
16	(WAIT) Message from the system *	CDTS may now be activated.
17	(TYPE) CDTS (TRANSMIT)	This places the CDT Executive into execution.
18	(WAIT) Message from CDTS CDTS STARTED STUDENT FILE - CAT/FILE STRING *	1. This sequence informs you that CDTS has been activated. CDTS requires you to identify your Student File. 2. The nonblinking asterisk (*) is the prompting cue for you to enter a response. 3. Refer to AFM 50-752, Attachment 5, for action to be taken if error conditions arise.
19	(TYPE) Catalog/file description of your Student File Example: STUDENT/CLARK (TRANSMIT)	1. Enter the catalog/file description of your Student File as designated by your installation. The Student File contains information as to your progress through a lesson. 2. Type the message exactly. An incorrect entry will return you to step 18 where you may continue.
20	(WAIT) Message from CDTS IAW THE PRIVACY ACT OF 1974, YOU ARE ADVISED THAT DISCLOSURE OF THIS INFORMATION IS VOLUNTARY SEE AFM 50-752, ATTACHMENT 8, FOR MORE INFORMATION STUDENT SSAN (9) *	1. Your SSAN is part of your student identification. This is a security feature to verify your Student File before CDTS begins updating your progress. 2. Again, the asterisk (*) cues you to enter a response.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
21	(TYPE) Your SSAN Example: 823456789 (TRANSMIT)	1. Your SSAN must be nine (9) numerics with no spaces or dashes. 2. The student SSAN is used to output student records to the training monitor to certify completion of a course. 3. If this is the first time you have accessed CDTS, whatever entry you make for an SSAN must then be used to continue updating that Student File. In the event CDTS does not accept your SSAN entry, CDTS will terminate. To rectify this condition, the training monitor will have to ERASE your Student File before you can continue. 4. If you have accessed CDTS previously and your Student File is current, you will be advanced to step 30.
22	(WAIT) Optional Message from CDTS GRADE OR TITLE (6) *	This message prompts you to identify your military grade or your civilian title.
23	(TYPE) Your Grade or Title Examples: 02 GS13 PL313 E4 (TRANSMIT)	1. This information is recorded in your Student File. 2. Your entry must not exceed six (6) characters.
24	(WAIT). Optional Message from CDTS LAST NAME (30) *	This message prompts you to enter your last name.

STEP	ACTION	DESCRIPTION
25	(TYPE) Your LAST Name Example: CLARK (TRANSMIT)	1. This information is recorded into your Student File. 2. Your entry must not exceed thirty (30) characters.
26	(WAIT) Optional Message from CDTS FIRST NAME MI (30) *	This message prompts you to enter your first name and middle initial.
27	(TYPE) Your FIRST Name and MI Example: RICHARD T (TRANSMIT)	1. This information is recorded into your Student File. 2. Your entry must not exceed thirty (30) characters.
28	(WAIT) Optional Message from CDTS ORGANIZATION (24) *	This message prompts you to identify your organization.
29	(TYPE) Your Organization Example: 3390 HQ Sq Sec (TRANSMIT)	1. This information is recorded into your Student File. 2. Your entry must not exceed twenty-four (24) characters.
30	(WAIT) Message from CDTS LESSON FILE - CAT/FILE STRING *	CDTS is now ready to begin a training session. Before CDTS can present a lesson, it requires the location of the CDT lessons.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
31	(TYPE) Catalog/file description of the CDT lesson file Example: CDTSCDTS/TUC/LESSONS (TRANSMIT)	1. Enter the catalog/file description of the Lesson File as designated by your installation. Type the message exactly. 2. The Lesson File contains the CDT lessons.
	Getting a Lesson	
32	(WAIT) Message from CDTS ENTER COMMAND *	CDTS has selected a Lesson File and is ready to begin a lesson.
33	(TYPE) ?GET LLLLLL NEW or ?GET LLLLLL OLD Example: ?GET TUC012 NEW (TRANSMIT)	1. LLLLLL is the six (6) position lesson name of a desired lesson. To retrieve lesson TUC012, LLLLLL would be replaced with TUC012. 2. NEW is to be specified if you are entering a lesson for the first time. 3. OLD is the default option and is appropriate if you are returning to a lesson that was begun and terminated before the lesson was completed.
34	(WAIT) Message from CDTS LLLLLL READY ENTER COMMAND *	1. CDTS informs you that it has selected the requested lesson. 2. LLLLLL is replaced by the six (6) position lesson name.
35	(TYPE) ?GOTO * (TRANSMIT)	Referring to step 33, ?GOTO * will start a "NEW" lesson at the beginning and restart an "OLD" lesson at the point at which the lesson was concluded if the lesson was incomplete.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
36	(WAIT) (READ) Course material will begin printing on your device.	1. Follow the course material closely. Read all information presented. 2. CDT lessons require a great deal of student interaction and participation. Various types of questions will be asked and you will be required to respond to the questions. 3. When CDTS presents a question, it will allow you time to transmit your answer. The nonblinking asterisk (*) cue will prompt you when CDTS is awaiting a response. When the asterisk (*) appears, go on to step 37. 4. As CDTS presents material, and as you respond to the course material, you will advance through the lesson. At some point you will reach the end of a lesson. When you reach the end of a lesson, CDTS will display the message "LESSON ENDED." When this occurs, go on to step 38.
37	(TYPE) Appropriate response to course material (TRANSMIT after each response)	1. When the asterisk (*) cue appears, transmit your response to the course material. 2. Continue with steps 36 and 37 until you are ready to discontinue a lesson or until the lesson has concluded. 3. To break a lesson, when the lesson has not been completed, go on to step 39.
38	(LESSON ENDED) Message from CDTS LESSON ENDED ENTER COMMAND *	1. This sequence occurs when a lesson has been concluded in its entirety. You should have reached this level from step 36, paragraph 4. 2. At this level you may return to step 33 and select another lesson, or you may go on to step 42 and conclude the training session.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
	Terminating a Lesson	
39	(WAIT) Message from CDTS *	1. The asterisk (*) is your cue to enter a response. You should have reached this point from step 37. 2. At this point CDTS is awaiting a response to course material; however, CDTS will respond to certain commands. To terminate a lesson when the lesson is incomplete, go on to step 40.
40	(TYPE) ?DONE (TRANSMIT)	By entering ?DONE following the CDTS cue in response to a question within a lesson, CDTS will record your progress in your Student File and mark your location in your current lesson. CDTS will then restart the respective lesson at that location when you access that lesson at a later date.
	Terminating/Continuing a Training Session	
41	(WAIT) Message from CDTS LLLLLL READY ENTER COMMAND *	1. LLLLLL will be replaced by the lesson name. 2. At this level, you have several options. First, you may return to step 35 and continue with your current lesson. Second, you may return to step 33 and select another lesson. Third, you may go on to step 42 and terminate the training session.
42	(TYPE) ?CLOSE (TRANSMIT)	The command ?CLOSE will terminate your training session.

EXHIBIT TUC012-4 (Cont)

STEP	ACTION	DESCRIPTION
43	(WAIT) Message from CDTS RECORDS SPAWNED 9999T CDTS ENDED	1. CDTS informs you it has terminated. 2. The message "RECORDS SPAWNED" is printed to inform you that your Student Records are being processed. The number 9999T is called a SNUMB and is used by the System to identify that activity. 3. Your Student Records are processed and sent to your training monitor. Your training monitor will use these records to identify problem areas or to certify completion of a course.
44	(WAIT) Message from the System *	CDTS has returned you to the Time-sharing System.
45	(TYPE) BYE (TRANSMIT)	BYE is the command to terminate your terminal activity.
46	(WAIT) Message from the system RESOURCES USED........... TIME-SHARING OFF AT......	
47	(PRESS) POWER switch OFF	Secure your device according to local convention.

EXHIBIT TUC012-4 (Cont)

Given the catalog structure SUPPLY/FUEL$JP4, create a subcatalog JET with:

1.	Password	ALCOHOL
2.	General Permission	READ
3.	Specific Permissions	(None)

EXHIBIT TUC030-1

Given a UMC "SUPPLY", create the quick access file "MYFILE" with the following specifications:

1.	Minimum Size	1 Block
2.	Maximum Size	10 Blocks
3.	Type	Sequential
4.	Password	ME
5.	General Permission	WRITE
6.	Specific Permissions	READ (for User BX)
7.	Open the file	

EXHIBIT TUC030-2

Given the catalog/file string "SUPPLY/BX/SHOES$ME" having a maximum size of 10 blocks, modify the file so that:

1. There is no password.

2. The maximum size is 20 blocks.

EXHIBIT TUC030-3

Given a user master catalog "SUPPLY" (i.e., you have logged on under the UMC SUPPLY), create the subcatalog "CAT1" with the following specifications:

1.	Password	ME
2.	General Permission	READ
3.	Specific Permissions	(None)

EXHIBIT TUC030-4

You have created the catalog string "SUPPLY/CAT1$ME". Now under "CAT1" create the file "FILE1" with the following specifications:

1. Minimum Size	1 Block
2. Maximum Size	10 Blocks
3. Type	Linked
4. Password	YOU
5. General Permissions	READ and WRITE
6. Specific Permissions	EXCLUDE (for user "USER1")
7. Open the file	

EXHIBIT TUC030-5

Given the catalog/file string "SUPPLY/CAT1$ME/FILE1$YOU," remove the password from "CAT1".

EXHIBIT TUC030-6

You have modified the catalog so that your catalog/file string is now "SUPPLY/CAT1/FILE1$YOU". Modify this linked file as follows:

1. Maximum Size	50 Blocks
2. General Permission	READ

(Hint: The general permissions are presently READ and WRITE. You must, in one step, remove the WRITE permission while keeping the READ permission.)

EXHIBIT TUC030-7

FILE SYSTEM

SECTION

1. File System Purpose

2. User Communication

3. File Structure

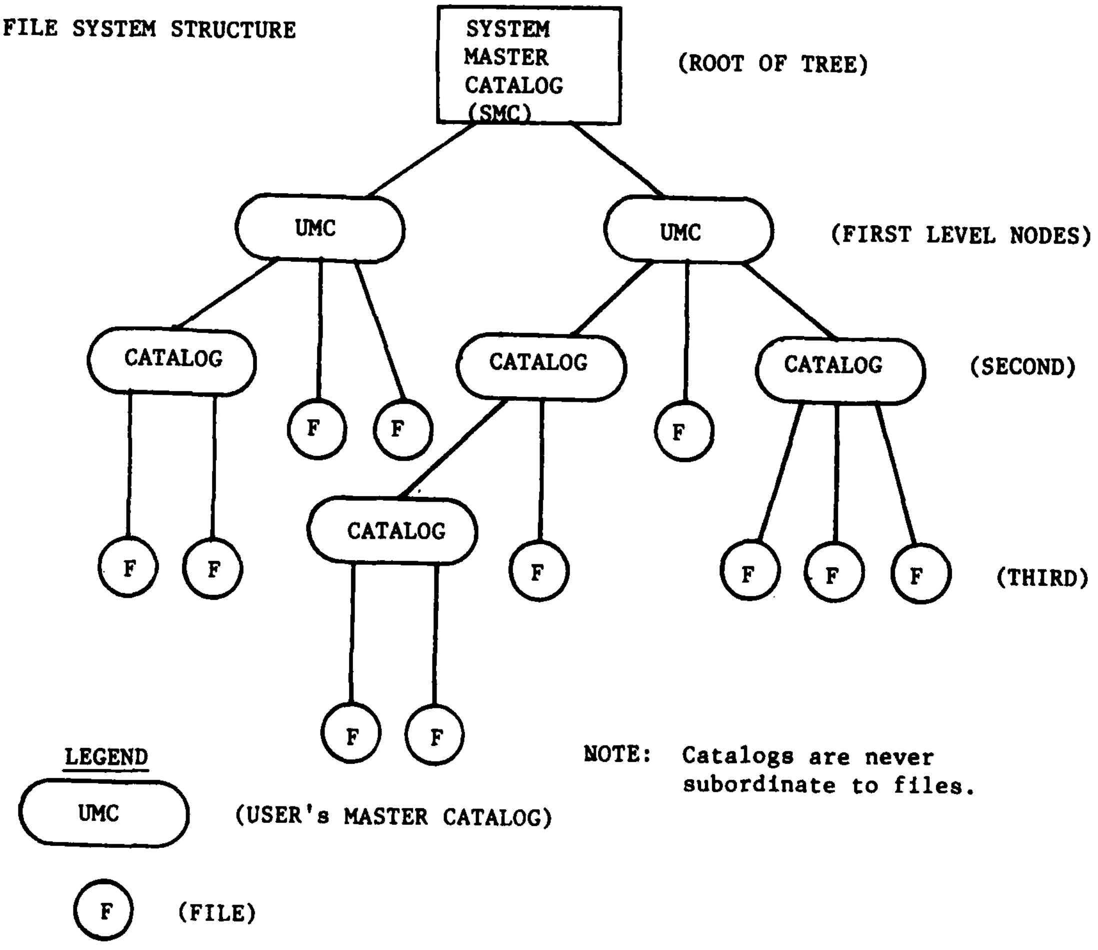
4. File System Security

   a. Passwords

   b. Permissions

EXHIBIT TUC035-1

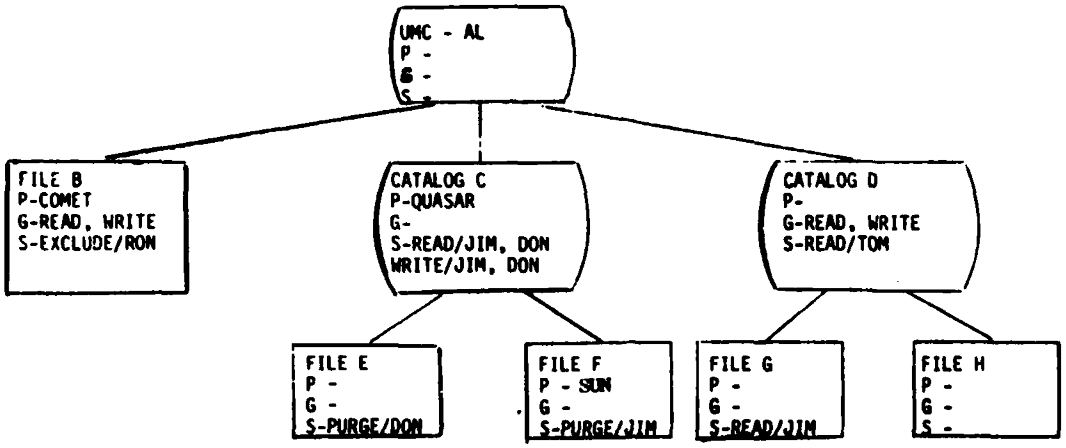
FILE SYSTEM STRUCTURE



NOTE: Catalogs are never subordinate to files.

EXHIBIT TUC035-2

SAMPLE FILE STRUCTURE *USERID$LOG-ON PASSWORD = AL$STAR



LEGEND
UMC - USER MASTER CATALOG
P - PASSWORD
G - GENERAL PERMISSIONS

*Anybody signing onto the computer system with AL$STAR is treated the same as the creator of the catalog/file string. The creator can access any catalog or file.

FILE B - To access use cat/file string AL/B$COMET.

FILE E - To access use cat/file string AL/C$QUASAR/E.
Any stranger including JIM and DON can read the file.
Only JIM and DON can write to the file.
Only DON can purge the file.

FILE F - To access use cat/file string AL/C$QUASAR/F$SUN.
Only JIM and DON can read or write the file.
Only JIM can purge the file.

FILE G - To access use cat/file string AL/D/G.
Any stranger including TOM and JIM can read the file.
Any stranger except TOM and JIM can write to the file.

FILE H - To access use cat/file strong AL/D/H.
Any stranger except TOM can write to the file.
Any stranger including TOM can read the file.

EXHIBIT TUC035-3

COMMAND SYNTAX

The command syntax for all CONVERT subsystem commands is:

```
command [input file(s)] [=output file] [:options]
```

EXHIBIT TUC040-1

MEDIA CODE OPTIONS

Option Abbreviation	File Format	Media Code (Octal)
BCD	Variable-length BCD	0
COMDK	BCD Compressed Deck Card Image	1
CARD	BCD 14-Word Card Image	2
PRINT	BCD Variable-Length Print Line Image	3
OLDASC	Obsolete TSS ASCII	4
ASCII	Standard System Format ASCII	5
APRINT	ASCII Print Line Image	6
ACARD	ASCII Card Image	7
SAME	No Change in Format	10

EXHIBIT TUC040-2

LINE NUMBER OPTIONS

Option	Function
ASIS or A	Leave line numbers as they are.
INSERT (i,j) or I (i,j)	Insert line numbers.
MOVE or M	Move line numbers from one place to another.
NORM (ch) or N (ch)	Move option with tab character expansion.
RESE (i,j) or R (i,j)	Resequence line numbers.
STRIP or S	Strip line numbers.
LABEL (abcde (i-j)) or L (abcde (i-j))	CARD and ACARD options.

EXHIBIT TUC040-3

KEYWORD: PROBLEM

SIMULATION EXERCISE 1

Your CURRENT FILE appears as follows:

```
                         1         2         3         4                       8
IMAGE SCALE -   1234567890123456789012345678901234567890-----------------------0

                010 :100:1CTR=0
                020 :200:1CTR=1CTR+1
                030 ::IF (1CTR .LT. 1000000) GO TO 200
                040 :300:STOP
```

The desired output format is as follows:

```
                                                                               1
                         1         2         3         4                       3
IMAGE SCALE -   1234567890123456789012345678901234567890-----------------------2

                010 :100:1CTR=0
                020 :200:1CTR=1CTR+1
                030 ::IF (1CTR .LT. 1000000) GO TO 200
                040 :300:STOP
```

NOTE: You have been assigned the following IDENT for this simulation:

```
$       IDENT    9AZ Q777 13A,R123WASH,987654323
```

Catalog/File Description - CDT/STUDENT/YOURFILE

EXHIBIT TUC041-1

KEYWORD: EXERCISE

SIMULATION EXERCISE 2

Your CURRENT FILE appears as follows:

```
                        1         2         3        4                        8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123457890------------------------Ø

               Ø1Ø :1ØØ:1CTR=Ø
               Ø2Ø :2ØØ:1CTR=1CTR+1
               Ø3Ø::IF (1CTR .LT. 1ØØØØØØ) GO TO 2ØØ
               Ø4Ø :3ØØ:STOP
```

The desired output is as follows:

```
                        1         2         3         4                7      8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø----------------34567890

               1ØØ 1CTR=Ø                                              ØØØØØØ1Ø
               2ØØ 1CTR=1CTR+1                                         ØØØØØØ2Ø
                   IF (1CTR .LT. 1ØØØØØØ) GO TO 2ØØ                    ØØØØØØ3Ø
               3ØØ STOP                                                ØØØØØØ4Ø
```

NOTE: You will use the following IDENT for this simulated FORTRAN example:

```
$       IDENT   9AZ Q777 13A,R123WASH,987654323
```

Catalog/File Description - YOURCAT/MYFILE

EXHIBIT TUC041-2

KEYWORD: TUCØ41

SIMULATION EXERCISE 3

Your CURRENT FILE appears as follows:

```
                        1         2         3         4                                       8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø---------------------------------------Ø

               1ØØ &LOOP
               2ØØ &&MOVE Ø TO COUNTER.
               3ØØ &BACK.
               4ØØ &&ADD 1 TO COUNTER.
               5ØØ &&IF COUNTER LESS THAN LIMIT  GO TO BACK.
               6ØØ &&STOP RUN.
```

Your BCD file should appear as follows:

```
                        1         2         3         4                                       8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø---------------------------------------Ø


                     LOOP.
                         MOVE Ø TO COUNTER.
                     BACK.
                         ADD 1 TO COUNTER.
                         IF COUNTER LESS THAN LIMIT GO TO BACK.
                         STOP RUN.
```

NOTE: ASCII Catalog/File Description - ACAT/FILE/ASCFILE

BCD Catalog/File Description - ACAT/FILE/BCDFILE

EXHIBIT TUC041-3

KEYWORD: NANCY

SIMULATION EXERCISE 4

Your BCD file appears as follows:

```
                        1         2         3         4                        8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø----------------34567890

                      LOOP.                                             ØØØØØØ1Ø
                           MOVE Ø TO COUNTER.                           ØØØØØØ2Ø
                      BACK.                                             ØØØØØØ3Ø
                          ADD 1 TO COUNTER.                             ØØØØØØ4Ø
                          IF COUNTER LESS THAN LIMIT GO TO BACK.        ØØØØØØ5Ø
                          STOP RUN.                                     ØØØØØØ6Ø
```

Your CURRENT FILE should appear as follows:

```
                        1         2         3         4                        8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø-----------------------Ø

               ØØØØØØ1Ø&LOOP.
               ØØØØØØ2Ø&&MOVE Ø TO COUNTER.
               ØØØØØØ3Ø&BACK.
               ØØØØØØ4Ø&&ADD 1 TO COUNTER.
               ØØØØØØ5Ø&&IF COUNTER LESS THAN LIMIT GO TO BACK.
               ØØØØØØ6Ø&&STOP RUN.
```

NOTE: BCD Catalog/File Description - MYCAT/AFILE/BCDFILE (Simulated COBOL example)

ASCII File - Use your current file.

EXHIBIT TUC041-4

KEYWORD: BARBARA

SIMULATION EXERCISE 5

Your CURRENT FILE appears as:

```
                        1         2         3         4                                       8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø---------------------------------------Ø

               1ØØ$:IDENT:9AZ Q777 13A,R123WASH,987654323
               2ØØ$:FILSYS
               3ØØ:USERID:ABC$XYZ
               4ØØ:RESTORE:CATA/DESCRIP/MYFILE
               5ØØ$:TAPE9:AB,SID,,00075
               6ØØ$:ENDJOB
```

The desired output should appear as follows:

```
                        1         2         3         4                                       8
IMAGE SCALE -  123456789Ø123456789Ø123456789Ø123456789Ø---------------------------------------Ø

               $      IDENT  9AZ 1777 13A,R123WASH,987654323
               $      FILSYS
                      USERID  ABC$XYZ
                      RESTORE CATA/DESCRIP/MYFILE
               $      TAPE9   AB,SID,,ØØØ75
               $      ENDJOB
```

NOTE: These statements are in your current file.

EXHIBIT TUC041-5

TEXT EDITOR PROJECT

Using the TEXT EDITOR subsystem in the build mode, create the following file of textual information. The file is then to be corrected to read as is reproduced below. Any additional errors entered as you create this file will afford you additional opportunity to utilize the capabilities of TEXT EDITOR.

```
IF THE PROGRAM CONTAIN A MISTAKE
THE COMPUTER INFORMS THE USER

TIME ARING HERMITS A DIALOGUE BETWEEN THE
COMPUTER AND THE USER PERMITING THE DIALODUE
TO START IMMEDIATELY, WITHOUT WAITING FOR
DATA IS FED FROM THE TERMINAL DIRECTLY TO
THE COMPUTER TO COMPLETE PREVIOUS PROGRAMS.
THE COMPUTER AND ANSWERS ARE RECEIVED
QUICKLY AT THE SAME TERMINAL.

THE   PROGRAM CANNOT BE CORRECTED OR ALTERED BY
USER ASIF HE WERE TALKING BY PHONE, ACCEPT IN THIS
CASE THE CONVERSATION IS
TYPED OR DISPLAYED, DEPENDANT UPON THE TYPE
OF TERMAL IN USE.
```

TEXT EDITOR EXERCISE

FINAL RESULTS SHOULD BE AS FOLLOWS:

```
TIME-SHARING PERMITS A DIALOGUE BETWEEN THE
COMPUTER AND USER PERMITTING THE DIALOGUE
TO BEGIN IMMEDIATELY, WITHOUT WAITING FOR
THE COMPUTER TO COMPLETE PREVIOUS PROGRAMS.
DATA IS FED FROM THE TERMINAL DIRECTLY TO
THE COMPUTER AND ANSWERS ARE RECEIVED
QUICKLY AT THE SAME TERMINAL.

IF THE PROGRAM CONTAINS A MISTAKE, THE
COMPUTER INFORMS THE USER.

THE PROGRAM CAN BE CORRECTED OR CHANGED BY
THE USER AS IF HE WERE CONVERSING BY PHONE,
EXCEPT IN THIS CASE, THE CONVERSATION IS
TYPED OR DISPLAYED, DEPENDENT UPON THE TYPE
OF TERMINAL IN USE.
```

EXHIBIT TUC050-1

EDITOR VERB FORMATS

1. VERB
2. VERBDM:/ST/
3. VERBDM;N
4. VERBDM:/ST/;N
5. VERBDM:/ST/;/ST/
6. VERBDM:/ST/;N:/ST/
7. VERBM:/ST/,/ST/

WHERE: VERB - is a selected editor verb.

D - is the position for a "V" for verify

M - is the mode indicator. "S" for string

/ - is a delimiter (any character may be used)

ST - is a string of characters

N - is an integer repeat value

NOTE: Format 7 applies only to the verbs CUT, COPY, or PASTE.

EDITOR EXAMPLES

B	Returns pointer to beginning of text.
B;5	Backs up pointer 5 lines.
P	Prints the line where the pointer is positioned.
P;3	Prints the line where the pointer is positioned along with the next 2 lines of text.
D	Deletes the line where the pointer is positioned.
D:/02/;*	Deletes all lines that begin with "02".
CUT:/HE/,/SHE/	Deletes the line that begins "HE" through the line that begins "SHE" and places them in a collector file.
COPY:/ME/,/WE/	Copies the line that begins "ME" through the line that begins "WE" and places them in a collector file.

EXHIBIT TUC050-2

PASTE:/281/ Takes whatever is currently in the collector file and copies it after the line that begins with "281".

F:/15/ Finds the line that begins "15".

I:/21/:/HELLO/ Insert after the line that begins "21", the line "HELLO".

RS:/16/;*:/17/ Finds every occurrence of the characters "16" in the text and replaces them with the characters "17".

EXHIBIT TUC050-2 (Cont)

RUNOFF FINAL PROJECT

Use the following information, along with the required RUNOFF commands, to produce a student list.

```
     Student Name+Class No.+Graduation Date

     BIMBO, FRANKLIN T.+06B+07DEC81
     BRIGHT, MARY L.+06A+07DEC81
     RAMJET, ROGER A.+10B+5JAN81
     SMITH, JOHN J.+10A+5JAN81
     ZULU, TOM T.+05A+22NOV81
```

This information, along with the required RUNOFF commands, should be entered into an editor file. The formatting will be as follows:

1. Three columns starting at positions 12, 40, 55.

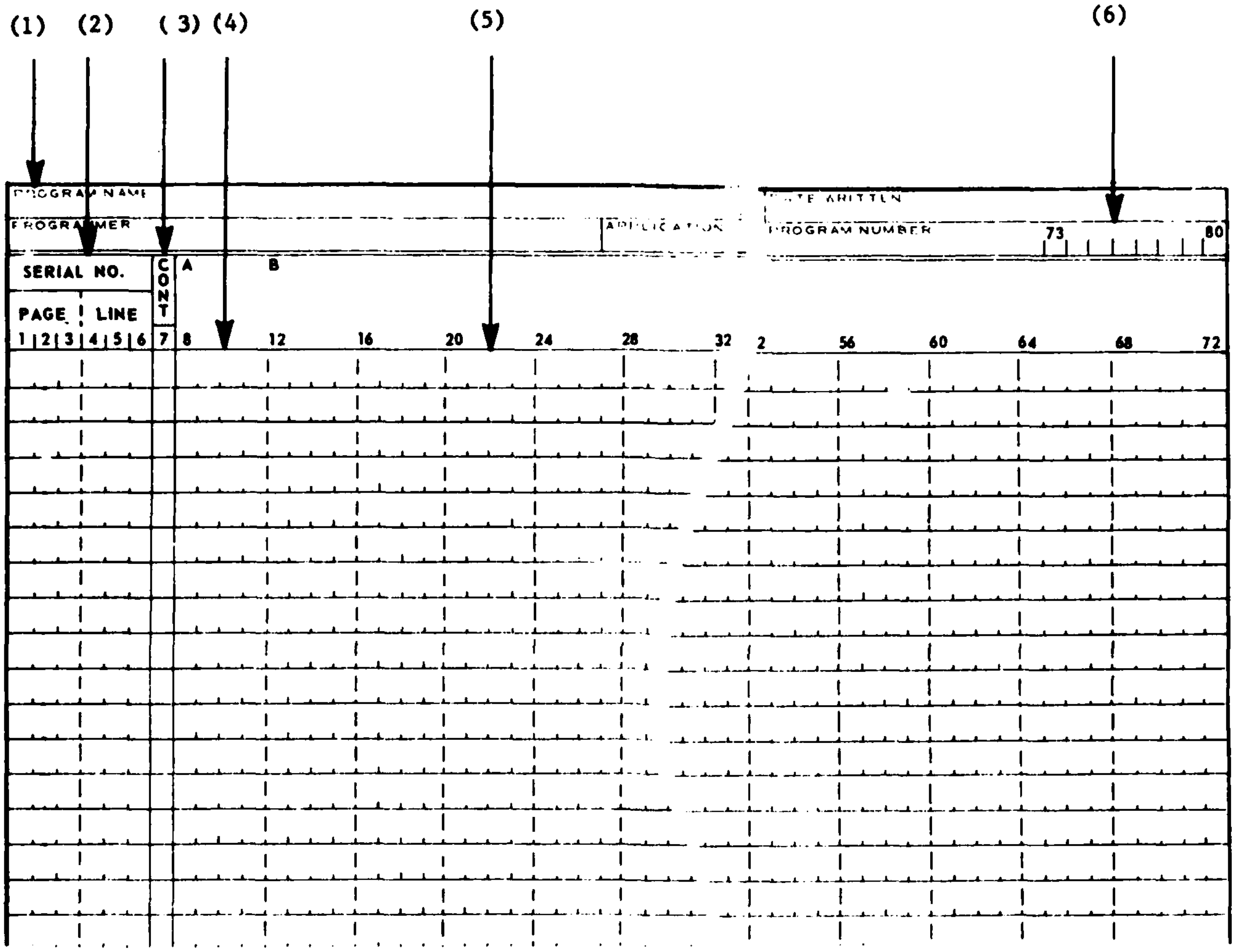
2. Use a two-inch margin for top of page.

3. Skip a line between the heading and the first student.

After entering the proper formatting commands, obtain a copy of this file using a print command.

EXHIBIT TUC060-1

COBOL PROGRAMMING (H6000-CDT) COURSE EXHIBITS



(1) PROGRAM NAME OR HEADING AREA

(2) SEQUENCE AREA (Card Columns 1-6)

(3) CONTINUATION AND COMMENT AREA (7)

(4) "A" AREA (8-11)

(5) "B" AREA (12-72)

(6) IDENTITY AREA(73-80)

KEYWORD CMT

EXHIBIT CBL010-1

TITLE: Sample COBOL-68 Program for Lesson CBL010.

```
$      IDENT   FE776AOJ/31,3-7/CDT-S,
$      USERID ABCD$EFG/DNM
$      COBOL  NDECK
010100 IDENTIFICATION DIVISION.
010200 PROGRAM-ID.      PAYROL.
010300 REMARKS.  THIS PROGRAM IS EXHIBIT CBL010-2.  THE PROGRAM SERVES
010400     AS A SAMPLE COBOL-68 PROGRAM.  BASICALLY THE PROGRAM READS
010500     TIMECARDS, CALCULATES WAGE AND PRODUCES A REPORT.
010600 ENVIRONMENT DIVISION.
010700 CONFIGURATION SECTION.
010800 SOURCE-COMPUTER.  6000-EIS.
010900 OBJECT-COMPUTER.  6000-EIS.
010950 INPUT-OUTPUT SECTION.
011000 FILE-CONTROL.
011100     SELECT TIME-CARDS ASSIGN TO TC-CARDS.
011200     SELECT PAY-REPORT ASSIGN TO PR-LISTING.
011300 I-O-CONTROL.
011400     APPLY SYSTEM STANDARD FORMAT ON TIME-CARDS, PAY-REPORT.
011500 DATA DIVISION.
011600 FILE-SECTION.
011700 FD  TIME-CARDS  LABEL RECORD IS STANDARD.
011800 01  IN-REC.
011900     03  NAME                     PICTURE X(20).
012000     03  SSAN                     PICTURE 9(9).
012100     03  FILLER                   PICTURE X(11).
012200     03  HOURS                    PICTURE 999V9.
012300     03  PAY-RATE                 PICTURE 99V99.
012400     03  FILLER                   PICTURE X(32).
012500 FD  PAY-REPORT  LABEL RECORD IS STANDARD.
020100 01  OUT-REC.
020200     03  FILLER                   PICTURE X(10).
020300     03  PNAME                    PICTURE X(20).
020400     03  FILLER                   PICTURE X(11).
020500     03  PSSAN                    PICTURE 9(9).
020600     03  FILLER                   PICTURE X(10).
020700     03  PHOURS                   PICTURE ZZ9.9.
020800     03  FILLER                   PICTURE X(10).
020900     03  PRATE                    PICTURE $$9.99.
021000     03  FILLER                   PICTURE X(9).
021100     03  PGROSS-PAY               PICTURE $$$,$$$.99.
021200     03  FILLER                   PICTURE X(32).
021300 WORKING-STORAGE SECTION.
021400 77  EMPLOYEE-CTR                 PICTURE 999.
021500 77  GROSS-PAY      VALUE ZEROES  PICTURE 9(5)V99.
021600 77  TOT-PAY        VALUE ZEROES  PICTURE 9(5)V99.
021700 01  FIRST-LINE   VALUE "          EMPLOYEE NAME
021800-    "SSAN                HOURS          RATE             GROSS PA
021900-    "Y              "                 PIC X(132).
022000 01  LAST-LINE.
022100     03  FILLER     VALUE "TOTAL NUMBER OF EMPLOYEES = "
```

EXHIBIT CBL010-2

```
022200                                          PICTURE X(28).
022300     03  TOT-EMP                          PICTURE ZZ9.
022400     03  FILLER          VALUE SPACES     PICTURE X(29).
022500     03  FILLER          VALUE "TOTAL PAYROLL = "  PICTURE X(16).
030100     03  TOT-PAYROLL                      PICTURE $$$,$$$.99.
030200     03  FILLER          VALUE SPACES     PICTURE X(46).
030300 PROCEDURE DIVISION.
030400 005-BEGIN SECTION.
030500 010-OPEN.
030600     OPEN INPUT TIME-CARDS.
030700     OPEN OUTPUT PAY-REPORT.
030800     MOVE SPACES TO OUT-REC.
030900     WRITE OUT-REC BEFORE ADVANCING TO TOP.
031000     WRITE OUT-REC FROM FIRST-LINE BEFORE ADVANCING 3 LINES.
031100     MOVE SPACES TO OUT-REC.
031200 015-READ.
031300     READ TIME-CARDS  AT END TO GO 020-END.
031400     MOVE NAME TO PNAME
031500     MOVE SSAN TO PSSAN
031600     MOVE HOURS TO PHOURS
031700     MOVE PAY-RATE TO PRATE.
031800     MULTIPLY HOURS BY PAY-RATE GIVING GROSS-PAY ROUNDED.
031900     MOVE GROSS-PAY TO PGROSS-PAY.
032000     ADD GROSS-PAY TO TOT-PAY.
032100     ADD 1 TO EMPLOYEE-CTR.
032200     WRITE OUT-REC BEFORE ADVANCING 2 LINES.
032300     GO TO 015-READ.
032400 020-END SECTION.
032500 025-TRAILER.
040100     MOVE EMPLOYEE-CTR TO TOT-EMP
040200     MOVE TOT-PAY TO TOT-PAYROLL.
040300     WRITE OUT-REC FROM LAST-LINE BEFORE ADVANCING TO TOP.
040400 030-TERMINATE.
040500     CLOSE TIME-CARDS, PAY-REPORT.
040600     STOP RUN.
$      EXECUTE
$      DATA    TC
BILL CATO           123456789          04000750
TONY DE STEFANO     234567890          05500625
TERRELL LANDRY      345678901          06000700
DON MEEK            456789012          06500675
TIK TOKAZ           567890123          07000650
$      SYSOUT  PR
$      ENDJOB
***EOF
```

EXHIBIT CBL010-2 (Cont)

```
EMPLOYEE NAME                  SSAN       HOURS        RATE       GROSS PAY

BILL CATO                   123456789     40.0        $ 7.50      $300.00

TONY DE STEFANO             234567890     55.0        $ 6.25      $343.75

TERRELL LANDRY              345678901     60.0        $ 7.00      $420.00

DON MEEK                    456789012     65.0        $ 6.75      $438.75

TIK TOKAZ                   567890123     70.0        $ 6.50      $455.00

TOTAL NUMBER OF EMPLOYEES = 5          TOTAL PAYROLL = $1,957.50
```

EXHIBIT CBL010-2 (Cont)

MOVE:

$$\underline{\text{MOVE}} \quad \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal} \end{array} \right\} \quad \underline{\text{TO}} \quad \text{identifier-2} \qquad [, \text{ identifier-3}] \ldots$$

Figure 1

WRITE:

$$\underline{\text{WRITE}} \qquad \text{record-name} \qquad [\underline{\text{FROM}} \quad \text{identifier-1}]$$

$$\left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \quad \text{ADVANCING} \quad \left\{ \begin{array}{l} \text{identifier-2 LINES} \\ \text{integer LINES} \\ \text{TO } \underline{\text{TOP}} \text{ OF PAGE} \\ \text{mnemonic-name} \end{array} \right\} \right]$$

Figure 2

EXHIBIT CBL030-1

DATA CARD:

Card Column	Data
1-25	NAME
26-27	GRADE (01-09,31-39)
28-33	BASE PAY (Dollars and Cents)
	ADDRESS
34-36	APT NUMBER
37-40	STREET NUMBER
41-50	STREET NAME
51-70	CITY
71-72	STATE ABBREVIATION
73-77	ZIP CODE

Figure 1

OUTPUT LISTING:

Print Position	Data
5-29	NAME
31-32	GRADE
38-45	PAY (Edit with a dollar sign and a decimal)
51-70	CITY
72-73	STATE

Figure 2

EXHIBIT CBL050-1

VEHICLE REGISTRATION FILE:

Card Column	Data
1-6	DECAL NUMBER
13-18	LICENSE NUMBER
19-20	STATE
21-45	OWNER'S NAME
46-50	MAKE OF AUTOMOBILE
51-55	MODEL
56-57	YEAR
58-61	LAST INSPECTION DATE (MMYY)
	NUMBER OF SEAT BELTS
62	FRONT
63	REAR
64-68	PRIMARY COLOR
	FIRST VIOLATION
69-70	POINTS
71-74	DATE (YYMM)
	SECOND VIOLATION
75-76	POINTS
77-80	DATE (YYMM)

Figure 1

VEHICLE REGISTRATION LISTING:

Print Position	Date
2-7	DECAL NUMBER
12-36	OWNER'S NAME
40-44	MAKE
49-53	MODEL
59-60	YEAR
69-74	LICENSE NUMBER
82-83	STATE

Figure 2

EXHIBIT CBL050-2

```
000750 FD  CARD-FILE.
000760 01  CARD-REC.
000770     03  SR-NUM1     PIC 9(6).
000780     03  CD-CODE     PIC 9.
000790     03  FLDA1       PIC9(8).
000800     03  FLDB1       PIC XX.
000810     03  FLDC1       PIC X(20).
000820     03  FILLER      PIC X(43).
000830 01  CARD-REC2.
000840     03  SRNUM2      PIC 9(6).
000850     03  FILLER      PIC 9.
000860     03  FLDA2       PIC X.
000870     03  FLDB2       PIC X(4).
000880     03  FLDC2       PIC 9(8).
000890     03  FLDD2       PIC 9(20).
```

Figure 1. Implicit Redefinitions

```
000100 WORKING-STORAGE SECTION.
000110 01  REK-1.
000120     03  FLDA        PIC X(10)  VALUE "1234567890".
000130     03  FLDB.
000140         05  FLDC    PIC 999V99 VALUE 100.00.
000150         05  FLDD    PIC 999V99 VALUE ALL 9.
000160     03  FLDE        PIC X(5)   VALUE "ABCDE".
000170     03  FLDF        PIC 9(5)   VALUE 54321.
000180 01  REK-2  REDEFINES REK-1.
000190     03  FLDG.
000200         05  FLDH    PIC 9(5).
000210         05  FLDI    PIC 999V99.
000220         05  FLDJ    PIC X(5).
000230     03  FLDK        PIC XXX.
000240     03  FLDL        PIC X(7).
000250     03 FLDM.
000260         05  FLDN    PIC XX.
000270         05  FLDO    PIC XXX.
```

Figure 2. Explicit Redefinition

EXHIBIT CBL060-1

HEADING INFORMATION

Report Heading:

Print Position	Data
18-45	"VEHICLE REGISTRATION LISTING"
51-55	"DATE"
57-64	(Print the current date) NOTE: The "ACCEPT" Verb is used to obtain the date from the computer and place it into a WORKING-STORAGE entry that has been provided by you. See the Honeywell COBOL-68 Reference Manual for an example on how to set up the entries.

Page Heading:

Print Position	Data
1-8	"DECAL NO"
19-23	"OWNER"
40-43	"MAKE"
49-53	"MODEL"
58-61	"YEAR"
69-74	"LIC NO"
80-84	"STATE"

Report Data:

Print Position	Data
2-7	Decal number
12-36	Owner's name
40-44	Make
49-53	Model
58-61	Year
69-74	License number
82-83	State

EXHIBIT CBL060-2

PROBLEM SOLUTION CBL060

```
000530  WORKING-STORAGE SECTION
000540  01   SPACE-HDR  PIC X VALUE SPACES.
000560  01  HDR-2 PIC X(84)   VALUE "DECAL NO          OWNER"
000570-     " MAKE  MODEL  YEAR  LIC NO  STATE".
000580  01   HDR-1.
000590      03   HDR-TEXT  PIC X(56)  VALUE "      VEHICLE"
000600-     "REGISTRATION LISTING    DATE".
000610      03   MON PIC 99.
000620      03   D-1 PIC X VALUE "-".
000630      03   DA PIC 99.
000640      03   D-2 PIC X VALUE "-".
000650      03   YR PIC 99.
000660* THE ENTRIES FOR CURRENT DATE FOLLOW
000670  01  DAYS-DATE.
000680      03   MONTH PIC 99.
000690      03   DAY-OF-MONTH PIC 99.
000700      03   YEARS  PIC 99.
000710      03   TYME PIC 9(10) USAGE COMP-3.
```

EXHIBIT CBL060-2 (Cont)

TITLE: COBOL IF Statement

The following and additional information may be referenced in the Honeywell COBOL-o8 Reference Manual.

General Format:

```
                       statement-1      ELSE statement-2
   IF condition
                       NEXT SENTENCE    ELSE NEXT SENTENCE
```

SYNTAX RULES:

1. Statement-1 and statement-2 represent either a conditional statement or an imperative statement, and either may be followed by a conditional statement.

2. The "ELSE NEXT SENTENCE" phrase may be omitted only if it immediately precedes the final period of the sentence.

3. "Conditions" are described in your Honeywell COBOL-68 Reference Manual.

GENERAL RULES:

1. When an IF statement is executed, the following action takes place:

   a. If the condition is true, the statements immediately following the ccndition (represented by statement-1) are executed and control then passes implicitly to the next sentence.

   b. If the condition is false, either the statements following ELSE are executed or, if the ELSE phrase is omitted, the next sentence is executed.

2. When an IF statement is executed and the NEXT SENTENCE phrase is present, control passes explicitly to the next sentence depending on the truth value of the condition and the placement of the NEXT SENTENCE phrase in the statement.

3. Statement-1 and statement-2 may contain an IF statement. In this case, the IF statement is said to be nested.

   IF statements within IF statements may be considered as paired IF and ELSE combinations, proceeding from left to right. Thus, any ELSE encountered is considered to apply to the immediately preceding IF that has not already been paired with an ELSE.

4. When control is transferred to the next sentence, either implicitly or explicitly, control passes to the next sentence as written or to a return mechanism of a PERFORM or a USE statement.

EXHIBIT CBL090-1

TITLE: Variations of IF Statements

This and additional information may be referenced in your Honeywell COBOL-68 Reference Manual.

RELATIONAL TEST

$$\underline{\text{IF}} \left\{\begin{array}{l}\text{literal-1}\\ \text{data-name-1}\\ \text{arithmetic expression-1}\end{array}\right\} \quad \text{IS} \;\; \underline{\text{NOT}} \left\{\begin{array}{l} = \\ < \\ > \\ \underline{\text{EQUAL}} \text{ TO}\\ \underline{\text{LESS}} \text{ THAN}\\ \underline{\text{GREATER}} \; \underline{\text{THAN}}\end{array}\right\} \left\{\begin{array}{l}\text{literal-2}\\ \text{data-name-2}\\ \text{arithmetic expression-2}\end{array}\right\}$$

SIGN CONDITIONS

$$\underline{\text{IF}} \left\{\begin{array}{l}\text{data-name}\\ \\ \text{arithmetic expression}\end{array}\right\} \quad \text{IS} \;\; \underline{\text{NOT}} \left\{\begin{array}{l}\underline{\text{ZERO}}\\ \underline{\text{POSITIVE}}\\ \underline{\text{NEGATIVE}}\end{array}\right\}$$

CLASS TESTS

$$\underline{\text{IF}} \;\; \text{data-name} \;\; \text{IS} \;\; \underline{\text{NOT}} \left\{\begin{array}{l}\underline{\text{NUMERIC}}\\ \\ \underline{\text{ALPHABETIC}}\end{array}\right\}$$

COMPOUND CONDITIONS

$$\underline{\text{IF}} \;\; \text{condition-1} \left\{\begin{array}{l}\underline{\text{AND}}\\ \\ \underline{\text{OR}}\end{array}\right\} \text{condition-2} \left[\left\{\begin{array}{l}\underline{\text{AND}}\\ \\ \underline{\text{OR}}\end{array}\right\} \text{condition-n} \right] \;\; \ldots$$

CONDITIONAL VARIABLE TESTS

$$\underline{\text{IF}} \;\; [\underline{\text{NOT}}] \;\; \text{condition-name}$$

EXHIBIT CBL090-2

TITLE: Sample program segments needed for Final Quiz in CBL090.

1. Use the following COBOL program segment to respond to questions 9 thru 12 of the quiz in lesson CBL090.

```
03  KODE           PIC X.
    88  MONTHLY    VALUE "1".
    88  QTRLY      VALUE "2".
    88  VALID      VALUE "1" "2".
```

2. Use the following COBOL program segment to respond to questions 13 thru 15 of the quiz in lesson CBL090.

```
03  KODE           PIC S9.
    88  ISSU       VALUE 1, 2, 3.
```

Both of the above program segments are also located within lesson CBL090 and will be displayed.

EXHIBIT CBL090-3

TITLE: I-O-CONTROL Paragraph

The following and additional information may be referenced in your Honeywell COBOL-68 Reference Manual.

Format 1:

```
    I-O-CONTROL.  COPY   library-name
     [                     { word-2       }
     [ REPLACING  word-1  BY { literal-1    }
     [                     { identifier-1 }

      [                  { word-4       } ]       ]
      [ ,   word-3  BY  { literal-2    } ]  ...  ] .
      [                  { identifier-2 } ]       ]
```

Format 2:

```
    I-O-CONTROL.

       [APPLY PROCESS AREA ON file-name-1    [, file-name-2...]]

       [APPLY BLOCK SERIAL NUMBER ON file-name-3   [, file-name-4...]]

              { SYSTEM STANDARD }
       [APPLY {                 }  FORMAT ON file-name-5
              { VLR             }

                [, file-name-6...]]

         RERUN [ON file-name-7]

         EVERY integer-1 RECORDS OF file-name-8

  [       [{      RECORD }]
  [ SAME  [{ SORT-MERGE  }]   AREA FOR file-name-9, file-name-10
  [       [{ SORT        }]
                                                        ]
          [, file-name-11, file-name-12 ...]            ]   ...
 [                                                      ]
 [ MULTIPLE FILE TAPE CONTAINS file-name-13  [POSITION integer-2]

        [, file-name-14  [POSITION integer-4] ...]  ]  ...
```

Syntax and General Rules are found in your Honeywell COBOL-68 Reference Manual.

EXHIBIT CBL100-1

TITLE: List of matching responses for Quiz 2 and Quiz 3 in lesson CBL100.

1. Use the following list of clauses in Quiz 2 of lesson CBL100. You will enter the letter below corresponding to the type of clause you are being quizzed on by the matching type questions.

   A. RERUN
   B. SAME AREA
   C. LABEL RECORDS
   D. BLOCK CONTAINS
   E. SAME RECORD AREA
   F. MULTIPLE FILE TAPE

2. Use the following list of verbs in Quiz 3 of lesson CBL100. You will enter either the letter below corresponding to the verb or the verb itself in a matching type quiz.

   A. CLOSE
   B. COPY
   C. OPEN
   D. None of the above verbs apply.

EXHIBIT CBL100-2

TITLE: Exercise Problem for Lesson CBL100

You or your training monitor should store several records on a tape according to the following format:

Record Description

SSAN	1 - 9	
NAME	10 - 27	
TAX-RATE	28 - 29	(percent)
HOURLY-RATE	30 - 32	(dollars and cents)
HOURS-WORKED	33 - 35	(integer)
ALMTS	36 - 40	(dollars and cents)

You are to process the input tape file and create a printed listing according to the following format:

NAME	1 - 18	
GROSS-PAY	21 - 30	Float dollar signs to the decimal; print comma.
TAX	34 - 42	Fixed dollar sign, zero suppress all positions.
SURCHARGE	45 - 53	Fixed dollar sign, zero suppress all positions.
ALMTS	56 - 64	Fixed dollar sign, zero suppress all positions.
PAY-DUE	67 - 75	Dollar sign, check protecting asterisks for all positions.
OVERPAYMENT	78 - 87	Fixed dollar sign, zero suppress all positions.

Print the above left to right.

Print an indicative heading above each field. Bypass the record if HOURS-WORKED and ALMTS are both zero or spaces. Compute each man's pay status as follows:

GROSS = HOURLY-RATE * HOURS-WORKED
TAX = TAX-RATE * GROSS
SURCHARGE = 10 per cent of TAX
NET = GROSS - TAX - SURCHARGE - ALMTS

If a man has pay due, print it in 67-75. If he is in debt, print this in 78-87. Maintain a running total for END-OF-FILE balance report and print as follows:

Identification	Amount	
1-10	21 - 30	Float dollar signs to the decimal; print comma.
TOT-GROSS	AMOUNT	
TOT-TAX	"	
TOT-SURCHG	"	
TOT-ALMTS	"	
TOT-NET	"	
BALANCE	"	(Should be zero, but print in 31 if negative.)

EXHIBIT CBL100-3

TITLE: File Description (FD) of the Data Division

The following and additional information may be reference in your Honeywell COBOL-68 Reference Manual.

Format 1:

```
    FD  file-name   COPY   library-name

   [                       { word-2       }
   [ REPLACING   word-1  BY { literal-1    }
   [                       { identifier-1 }

          [          { word-4       }  ]      ]
          [, word-3 BY { literal-2    }  ]  ...]
          [          { identifier-2 }  ]      ]
```

Format 2:

```
    FD  file-name

      [                      { BINARY } [ { HIGH }          ] ]
      [ RECORDING MODE IS   {        } [ {      }  DENSITY ] ]
      [                      { BCD    } [ { LOW  }          ] ]

      [ BLOCK CONTAINS [integer-1 TO]  integer-2 { RECORDS    } ]
      [                                          { CHARACTERS } ]

      [ RECORD CONTAINS  [integer-3 TO]  integer-4   CHARACTERS ]

             { RECORDS ARE }  { STANDARD                                 }
        LABEL {             }  { OMITTED                                  }
             { RECORD IS   }  { data-name-1    [, data-name-2] ...      }
                              { label-name-1   [, label-name-2] ...     }

     [                             { data-name-4 }
     [ VALUE OF data-name-3 IS     {             }
     [                             { literal-1   }

          [                    { data-name-6 } ]      ]
          [ , data-name-5 IS   {             } ]  ... ]
          [                    { literal-2   } ]      ]

     [      { RECORD IS   }                                         ]
     [ DATA {             }  data-name-7  [, data-name-8]   ...     ]
     [      { RECORDS ARE }                                         ]

   [ { REPORT IS   }                                              ]
   [ {             }  report-name-1   [, report-name-2]   ...     ]
   [ { REPORTS ARE }                                              ]
```

Syntax and General Rules may be located in your Honeywell COBOL-68 Reference Manual.

EXHIBIT CBL100-4

TITLE: CLOSE Statement

The following and additional information may be referenced in your Honeywell COBOL-68 Reference Manual.

General Format:

$$\underline{\text{CLOSE}}\quad \text{file-name-1}\quad \left[\begin{array}{l}\underline{\text{REEL}}\\ \underline{\text{UNIT}}\end{array}\right] \left[\text{WITH}\left\{\begin{array}{l}\underline{\text{NO REWIND}}\\ \underline{\text{LOCK}}\end{array}\right\}\right]$$

$$\left[\text{, file-name-2}\quad \left[\begin{array}{l}\underline{\text{REEL}}\\ \underline{\text{UNIT}}\end{array}\right] \left[\text{WITH}\left\{\begin{array}{l}\underline{\text{NO REWIND}}\\ \underline{\text{LOCK}}\end{array}\right\}\right]\quad \ldots\right]$$

Syntax Rules:

1. Each file-name is the name of a file upon which the CLOSE statement is to operate; it must not be the name of a sort file or merge file.

2. The REEL and WITH NO REWIND options apply only to files stored on tape devices and other devices to which these terms are applicable. The UNIT option is applicable only to mass storage files in the sequential access mode.

General Rules:

See your Honeywell COBOL-68 Reference Manual.

EXHIBIT CBL100-5

TITLE: Format for the Perform Verb

Format 1:

$$\underline{\text{PERFORM}}\quad \text{procedure-name-1}\quad \left[\left\{\begin{matrix}\underline{\text{THRU}}\\ \underline{\text{THROUGH}}\end{matrix}\right\}\quad \text{procedure-name-2}\right]$$

Format 2:

$$\underline{\text{PERFORM}}\quad \text{procedure-name-1}\quad \left[\left\{\begin{matrix}\underline{\text{THRU}}\\ \underline{\text{THROUGH}}\end{matrix}\right\}\quad \text{procedure-name-2}\right]$$

$$\left\{\begin{matrix}\text{identifier-1}\\ \text{integer-1}\end{matrix}\right\}\quad \underline{\text{TIMES}}$$

Format 3:

$$\text{PERFORM}\quad \text{procedure-name-1}\quad \left[\left\{\begin{matrix}\underline{\text{THRU}}\\ \underline{\text{THROUGH}}\end{matrix}\right\}\quad \text{procedure-name-2}\right]$$

$$\underline{\text{UNTIL}}\quad \text{condition-1}$$

EXHIBIT CBL110-1

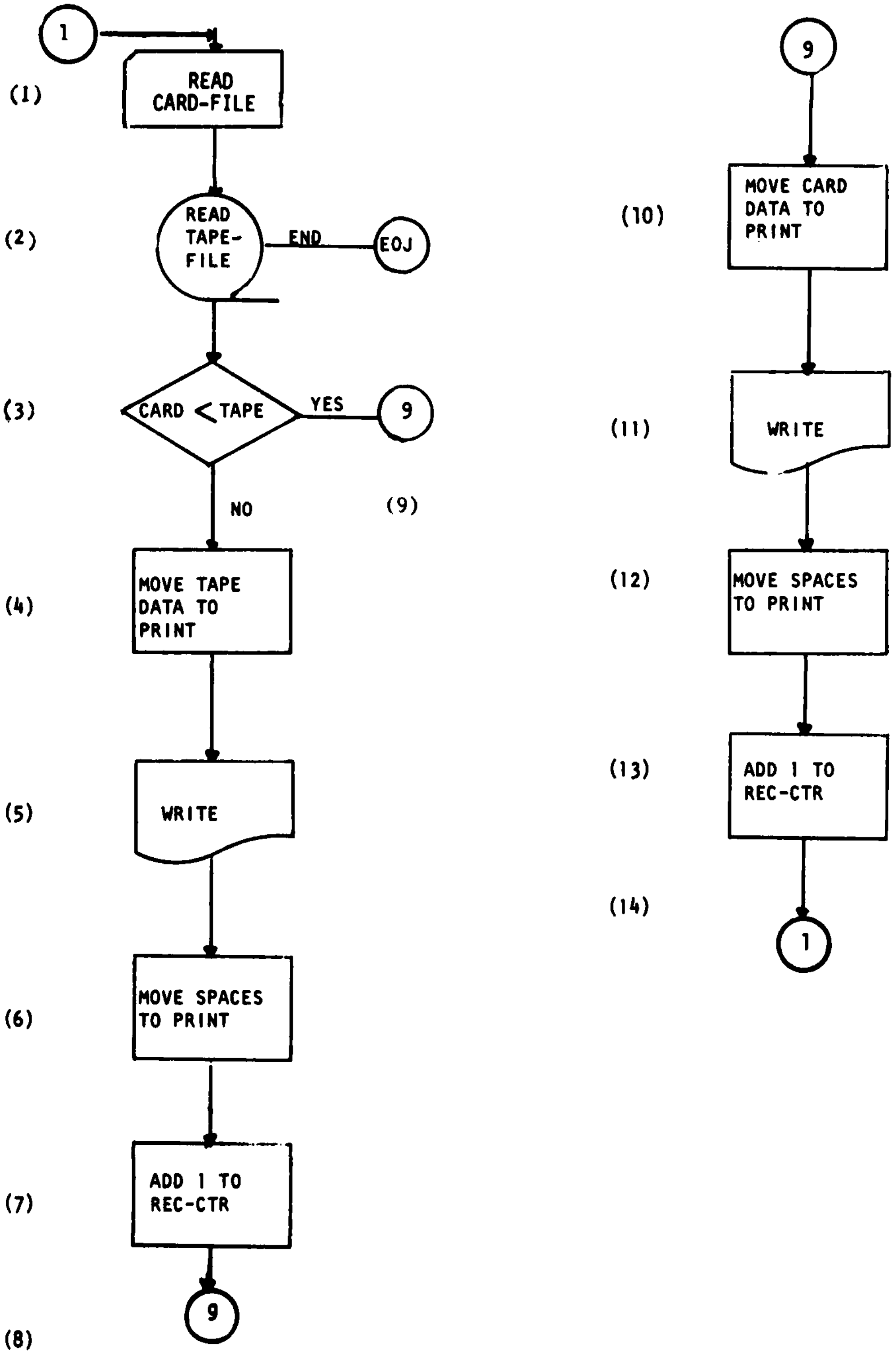


EXHIBIT CBL110-2

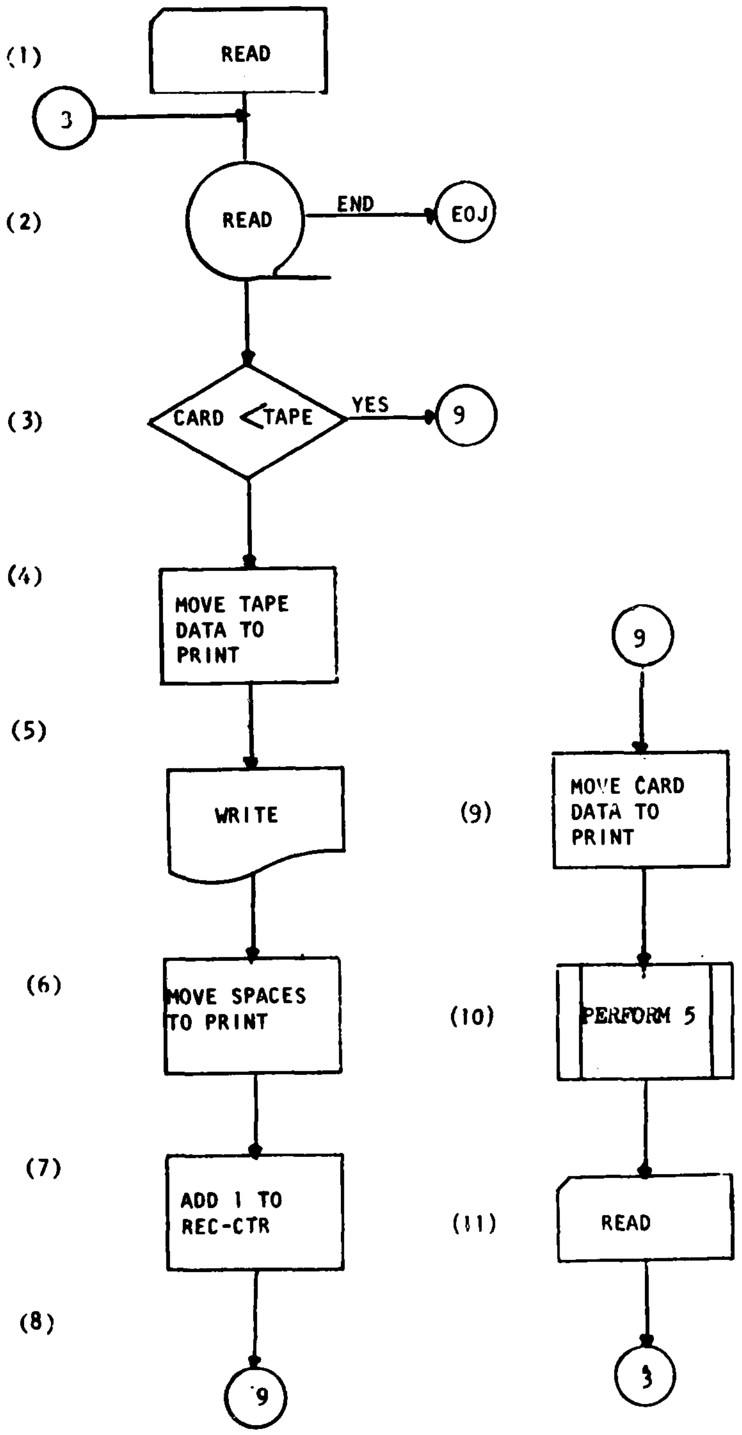


EXHIBIT CBL110-3

TITLE: Partial Coding for Exhibit CBL110-3

```
010-CARD-READ.
    READ CARD-FILE.
020-TAPE-READ.
    READ TAPE-FILE  AT END GO TO 990-EOJ.
030-COMPARE.
    IF C-SSAN LESS THAN T-SSAN  GO TO 090-CARD.
    MOVE T-NAME TO P-NAME.
    .
    .  (as many moves as necessary)
    .
050-WRITE.
    WRITE PRT-REC.
    MOVE SPACES TO PRT-REC.
    ADD 1 TO REC-CTR.
080-OUT.
    GO TO 020-TAPE-READ.
090-CARD.
    MOVE C-NAME TO P-NAME.
    .
    .  (as many moves as necessary)
    .
```

EXHIBIT CBL110-4

TITLE: Sample Routines to Print Heading Line

```
110-WRITE.
    WRITE PRT-REC.
    MOVE SPACES TO PRT-REC.
    ADD 1 TO LYN-CTR.
    IF LYN-CTR LESS THAN 30  GO TO 120-EXIT.  (Note the GO TO)
    WRITE PRT-REC BEFORE TOP OF PAGE.
    WRITE PRT-REC FROM HEAD-LINE BEFORE 2 LINES
    MOVE SPACES TO PRT-REC.
    MOVE 0 TO LYN-CTR.
120-EXIT.
    EXIT.
990-EOJ.
    (any instructions).
```

Figure 1

```
110-WRITE.
    WRITE PRT-REC.
    MOVE SPACES TO PRT-REC.
    ADD 1 TO LYN-CTR.
    IF LYN-CTR GREATER THAN 30 PERFORM 120-OFLO.
120-OFLO.
    WRITE PRT-REC BEFORE TOP.
    WRITE PRT-REC FROM HEAD-LINE BEFORE 2 LINES.
    MOVE SPACES TO PRT-REC.
    MOVE 0 TO LYN-CTR.
990-EOJ.
    (any instructions).
```

Figure 2

EXHIBIT CBL110-5

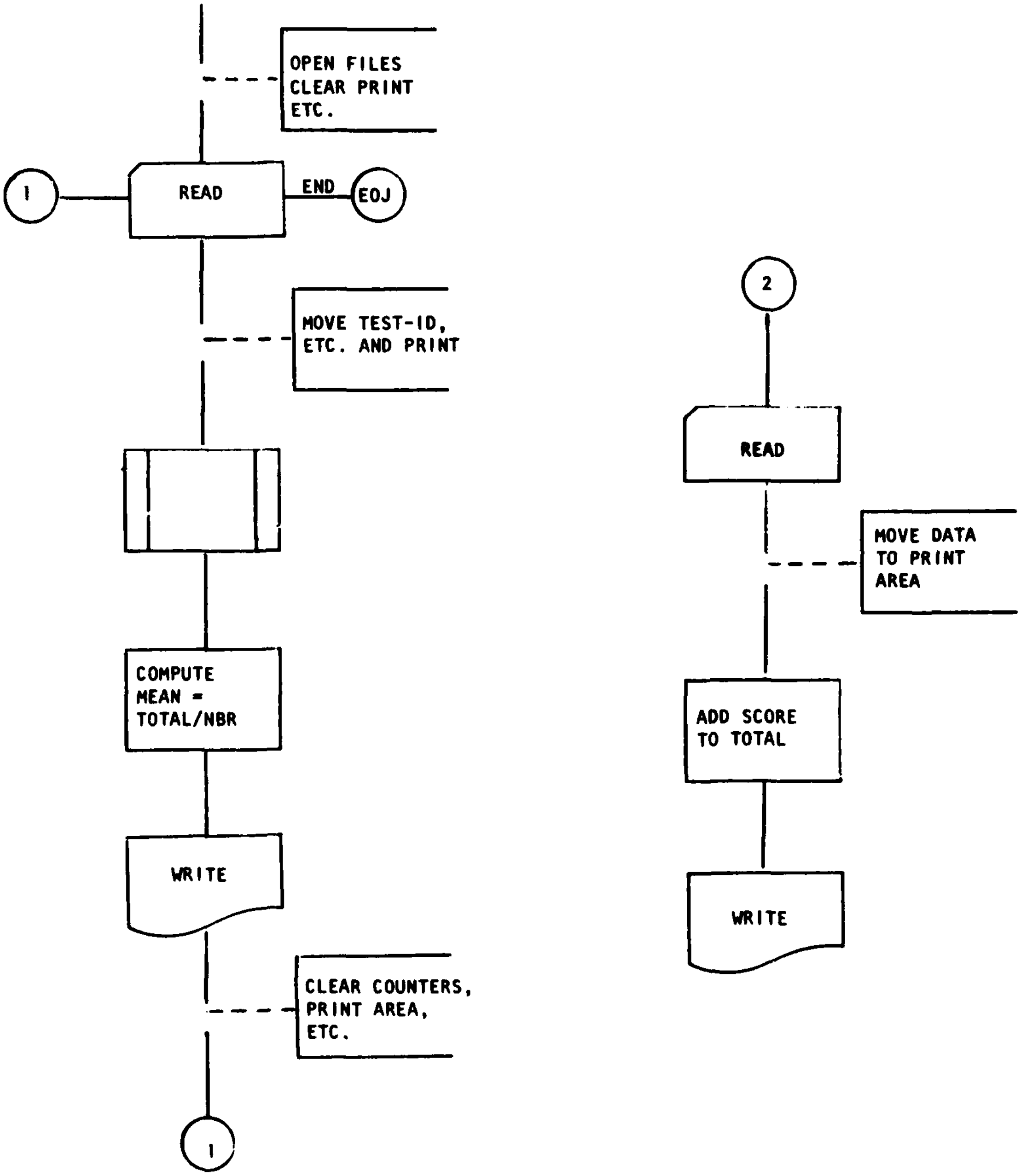


EXHIBIT CBL110-6

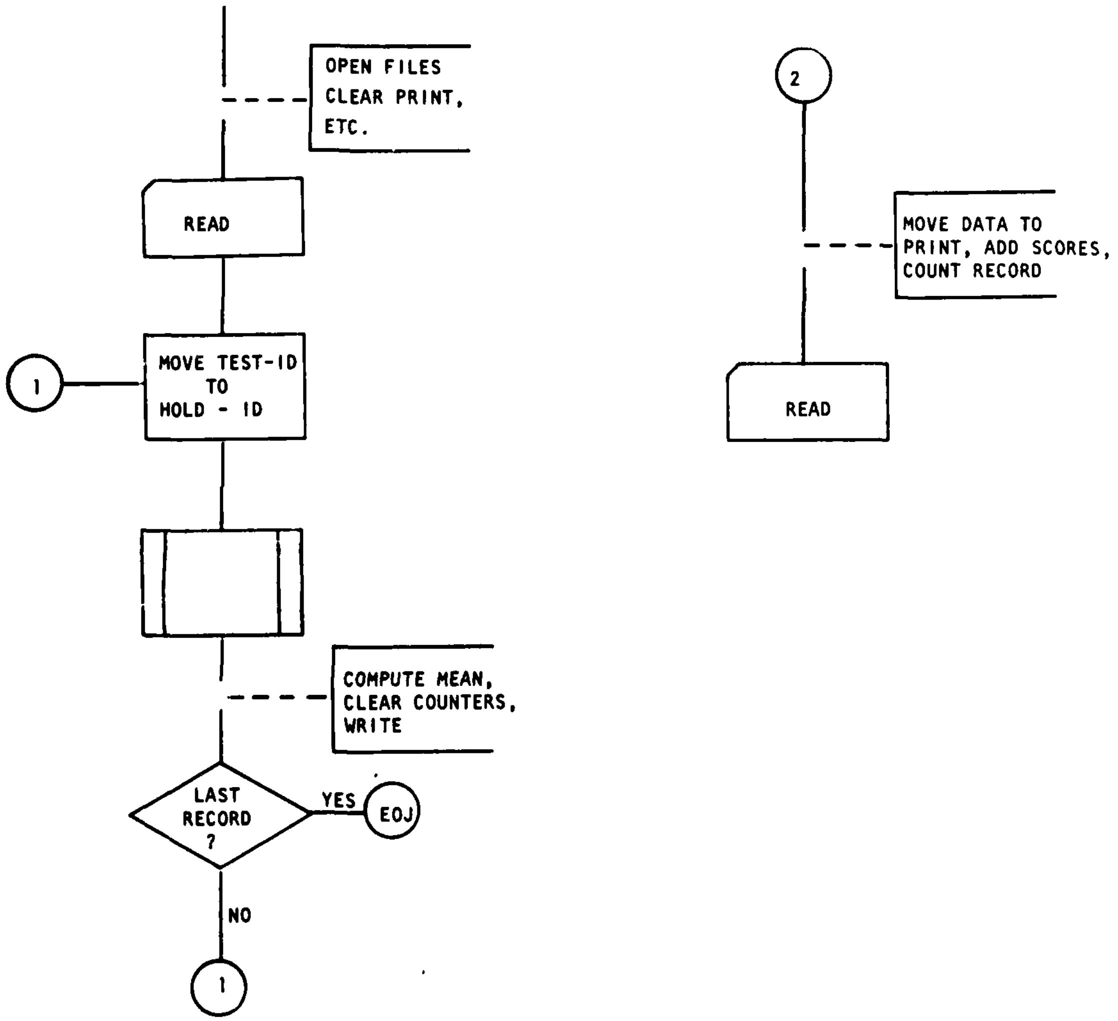


EXHIBIT CBL110-7

TITLE: SOURCE-COMPUTER Paragraph

The following and additional information may be referenced in your Honeywell COBOL-68 Reference Manual.

Format 1

```
[ SOURCE-COMPUTER.   COPY   library-name
                                        { word-2       }
    [ REPLACING   word-1   BY           { literal-1    }
                                        { identifier-1 }

                            { word-4       }
        [,   word-3   BY    { literal-2    } ]   ... ]   . ]
                            { identifier-2 }
```

Format 2

```
[                       { 6000     }       ]
[ SOURCE-COMPUTER.      {          }   .   ]
[                       { 6000-EIS }       ]
```

Syntax Rules:

1. The SOURCE-COMPUTER Paragraph must begin with the paragraph-name SOURCE-COMPUTER which must be followed by a period and a space.

2. When Format 1 is used, the COPY library-name phrase is required. The library-name must be identical to the name associated with the desired text on the library.

3. In Format 1, a word is any COBOL word.

4. When Format 2 is used, the computer specified must be a 6000 or 6000-EIS. Series 60 users should specify 6000-EIS.

General Rules:

1. For a description of the COPY function, see your Honeywell COBOL-68 Reference Manual.

2. This paragraph provides program documentation only and has no effect on compilation.

EXHIBIT CBL120-1

TITLE: FILE-CONTROL Paragraph

The following and additional information may be referenced in your Honeywell COBOL-68 Reference Manual.

Format 1:

```
    FILE-CONTROL.   COPY   library-name
     [                          { word-2       }
     [ REPLACING   word-1   BY  { literal-1    }
     [                          { identifier-1 }
     [
     [ [                  { word-4       } ]       ]
     [ [ ,  word-3   BY   { literal-2    } ]  ...  ]
     [ [                  { identifier-2 } ]       ]
```

Format 2:

```
    FILE-CONTROL.

     { SELECT   [OPTIONAL]  [OVERLAY]  file-name-1  [RENAMING  file-name-2]
     {
                                   { device-type-1  , device-type-2 }
       ASSIGN  TO  [integer-1]     { file-code-1    , file-code-2   } ...
                                   { code-device-1  , code-device-2 }

     [                  REEL ]
     [ FOR MULTIPLE          ]
     [                  UNIT ]

        [            { integer-2 }            [ AREA  ] ]
        [ , RESERVE  {           } ALTERNATE  [       ] ]  [FOR BLANK COMMON]
        [            { NO        }            [ AREAS ] ]

       [  { FILE-LIMIT IS   }  { data-name-1 }  { THRU    }  { data-name-2 } 
       [ ,{                 }  {             }  {         }  {             }
       [  { FILE-LIMITS ARE }  { literal-1   }  { THROUGH }  { literal-2   }

           [   [  { data-name-3 }  { THRU    }  { data-name-4 } ]     ]
           [   [ ,{             }  {         }  {             } ] ... ]
           [   [  { literal-3   }  { THROUGH }  { literal-4   } ]     ]

          [                    { SEQUENTIAL } ]
          [ , ACCESS MODE IS   {            } ]
          [                    { RANDOM     } ]

          [, PROCESSING MODE IS SEQUENTIAL]
          [, ACTUAL KEY IS data-name-5]   . }
```

EXHIBIT CBL120-2

TITLE: FILE-CONTROL Paragraph (continued)

Format 3:

```
    FILE-CONTROL.
     { SELECT file-name-1
                                  { device-type-1   [, device-type-2] }
         ASSIGN TO [integer-1]    { file-code-1     [, file-code-2  ] } ... } ...
                                  { code-device-1   [, code-device-2] }
```

Syntax Rules:

1. The FILE-CONTROL paragraph is required when the INPUT-OUTPUT SECTION leader is present.

2. The FILE-CONTROL paragraph must begin with the paragraph-name FILE-CONTROL which must be followed by a period and a space.

3. When Format 1 is used, the COPY library-name phrase is required. The library-name must be identical to the name associated with the desired text on the library.

4. In Format 1, a word is any COBOL word.

5. Format 3 must be used whenever sort files or merge files are selected to define a set of collation files for the sort or merge operation.

6. Integer-1 is treated as documentation only since the assignment of multiple devices is handled by the system using control cards.

7. Integer-1 may not be specified when file-code-2, file-code-3, etc., is also specified in the ASSIGN phrase.

8. Multiple file-codes in the ASSIGN phrase are treated as documentation only.

9. The file-code must be a two-character word consisting either of two letters (A through R, T through Z) or of one letter and one digit (0 through 9). File-codes beginning with the letter S should not be used in programs that utilize the sort or merge process. These file-codes, which include S1, S2, ..., SA, ..., SZ, have a special meaning in the sort or merge operation. The code specified must not be a COBOL reserved word. The first file-code for each file named in a SELECT sentence must be unique within the program. When the object program is submitted for execution, it is accompanied by peripheral assignment cards which are used to specify the peripheral device for each file. The file-code in the peripheral assignment card must be the same as that assigned in the source program. Each of the object program's files is associated with the proper peripheral device when the operating system matches the file-codes.

EXHIBIT CBL120-2 (Cont)

10. The MULTIPLE UNIT option is treated as documentation only.

General Rules: See the Honeywell COBOL-68 Reference Manual.

EXHIBIT CBL120-2 (Cont)

PROBLEM FOR CBL120

PROCESSING INSTRUCTIONS

1. This is a master file update program.

2. The records of the master file were placed on disk using the decal number as the relative address.

3. As each transaction record is read, the contents of decal number must be transferred to the relative key field for the master file.

4. Then, the proper master record must be read, updated, and written back to the file.

5. NOTE. Reductions must not be processed by this program.

6. If the first violation field is blank, the transaction date and points must be placed in the first violation field.

7. If it already contains data, the next violation must be updated.

8. After reading the end of the transaction file, the report must be printed.

9. The report must reflect every master record which contains data in either or both violation fields.

PAGE HEADING

Positions	Data
1 - 6	Your program-ID
12 - 32	"SUMMARY OF VIOLATIONS"
34 - 38	"AS OF"
40 - 43	Current Date (YYMM Format)

COLUMN HEADING

Positions	Data
8 - 12	"DECAL"
15 - 30	"OWNER OF VEHICLE"
42 - 47	"POINTS"
49 - 57	"LAST DATE"

FORMS CONTROL

1. The heading lines must be printed on the top of each page.

2. Single space between the headings; double-space between the second heading line and the first detail line.

3. Print 25 single-spaced detail lines per page.

> NOTE: Do not process directly against the CAI002 Master File; instead, use the Utility System to copy the 6 block random file CAI002 to a separate disk file. In the event that a student modifies CAI002, use the procedures in Figure 3-2 of AFM 171-752 to restore the file.

FILES REQUIRED

Master File:
Medium = Random Disk File
File Name: CAI002
The record description is:

COLUMNS	DATA
1-6	Decal Number
13-18	License Number
19-20	State
21-44	Owner's Name
45-50	Make of Automobile
51-55	Model
56-57	Year
58-61	Last Inspection Date (MMYY)
	Number of Seat belts
62	Front
63	Rear
64-68	Primary Color
	First Violation
69-70	Points
71-74	Date (YYMM)
	Next Violation
75-76	Points
77-80	Date (YYMM)

EXHIBIT CBL120-3 (Cont)

Transaction File
Medium = Sequential Disk File
File Name: CAI001
Record Description

POSITIONS	DATA	

1	Transaction Type	B = Violation
		D = Reduction

2	Violation Type	1 = Parking 2 = Moving 3 = Speeding
3-8	License Number	
15-16	State	
17-18	Points	
19-24	Date of Violation (YYMMDD)	
30-35	Decal Number	
36-80	Description of Vehicle	

Report:
Record Description

POSITIONS	DATA
7-12	Decal Number
15-38	Owner's Name
44-45	Total Points (zero suppress first digit)
51-54	Date of Last Violation (YYMM)

When your program has successfully executed, the first page of your report should look like that shown on the next page with your program ID in the firs six positions of the heading.

EXHIBIT CBL120-3 (Cont)

SUMMARY OF VIOLATIONS AS OF 7205

DECAL	OWNER OF VEHICLE	POINTS	LAST DATE
000001	JAMES JONES	2	7205
000002	RON SMITH	1	7205
000003	BILL CALLIFLOWER	4	7108
000006	STAN BURNS	1	7205
000007	TED ROLLING	2	7206
000010	JIM PARRISH	6	7206
000011	SAMUEL GUENTHER	1	7201
000014	TODD BAILEY	2	7206
000015	TED RALEIGH	4	7206
000017	STAN BIRNY	3	7206
000018	ROBERT ROBERTSON	6	7206
000019	WILLIAM CONQUERER	3	7207
000022	STEVEN OWENS	1	7207
000023	WAYNE TUCKETT	1	7203
000025	RALPH WORDEN	2	7204
000027	JOHN LEVINE	6	7208

EXHIBIT CBL120-3 (Cont)

TITLE: The OCCURS Clause

Basic Format: OCCURS integer-1 TIMES. (See the Honeywell COBOL-68 Reference Manual for full format.)

```
Example:  01 WEEK-ARRAY    VALUE "SUNMONTUEWEDTHUFRISAT".
             03  DAY-NAME  PIC XXX  OCCURS 7 TIMES.
```

Basic Rules:

1. The clause must not be used with 01, 66, 77, or 88 level data descriptions.

2. The integer-1 must be a positive (greater than zero) integer.

3. The clause may ·lescribe group or elementary items.

4. An item with an OCCURS clause or an item subordinate to an item with an OCCURS clause:

   a. Must be subscripted when referenced in the Procedure Division.

   b. Cannot have a VALUE clause.

   c. Cannot be explicitly redefined.

EXHIBIT CBL130-1

TITLE: Sample Coding for Lesson CBL130

```
.
.
.
SPECIAL-NAMES.  GTIME IS XGTIMEX.
.
.
.
WORKING-STORAGE SECTION.
.
    03 HDR-MO          PIC XXX.
.
.
.
01  MONTH-TABLE  VALUE "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC".
    03  M-T              PIC XXX  OCCURS 12 TIMES.
.
.
.
01  DATE-TIME.
    03  DAYT.
        05  MNTH   PIC 99.
        05  DAYY   PIC 99.
        05  YEAR   PIC 99.
    03  TYME       PIC 9(10) COMP-3.
PROCEDURE DIVISION.
.
.
.
ACCEPT DATE-TIME FROM XGTIMEX.
MOVE M-T (MNTH) TO HDR-MO.
.
.
.
```

EXHIBIT CBL130-2

TITLE: Various Portions of Coding for Part 2 of Lesson CBL130

```
01  GRADE-SEX.
    03  GRADE  OCCURS 4 TIMES.
        05  SEX  PIC 99  OCCURS 2 TIMES.


0000000000000000   -   View of Core at "GRADE-SEX".
 09  10  11  12    -   Grade.
FFMMFFMMFFMMFFMM   -   Sex.
```

Figure 1

```
FILE SECTION.
FD  PRINTOUT.
01  P-REC.
    03  HDR-TITLE        PIC X(23).
    03  FILLER           PIC X(7).
    03  HDR-MONTH        OCCURS 12.
        05  H-M          PIC XXX.
        05  FILLER       PIC XXX.
    03  FILLER           PIC X(30).
WORKING-STORAGE SECTION.
77  SUB-P                PIC 99  VALUE 13.
77  SUB-T                PIC 99  VALUE 0.
01  MONTH-TABLE   VALUE "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC".
    03  M-T              PIC XXX  OCCURS 12.
```

Figure 2

```
01  TABLE-X  VALUE "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789".
    03  FIELD-A          OCCURS 3.
        05  FIELD-B      OCCURS 3.
            07  FIELD-C    PIC XX.
            07  FIELD-D    PIC XX.


AB CD EF GH IJ KL MN OP QR ST UV WX YZ 01 23 45 67 89
-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
CC DD CC DD CC DD CC DD CC DD CC DD CC DD CC DD CC DD
----- ----- ----- ----- ----- ----- ----- ----- -----
  B     B     B     B     B     B     B     B     B

_________________ _________________ _________________
        A                 A                 A
```

Figure 3

EXHIBIT CBL130-3

TITLE: PERFORM Verb with the VARYING Option

```
                          [ { THRU    }                   ]
PERFORM procedure-name-1  [ {         }  procedure-name-2 ]
                          [ { THROUGH }                   ]

          { identifier-1 }        { identifier-2 }
VARYING   {              }  FROM  { index-name-2 }
          { index-name-1 }        { literal-1    }

     { identifier-3 }
BY   {              }      UNTIL  condition-1
     { literal-2    }
```

Figure 1

```
                          [ { THRU    }                   ]
PERFORM procedure-name-1  [ {         }  procedure-name-2 ]
                          [ { THROUGH }                   ]

          { identifier-1 }        { identifier-2 }
VARYING   {              }  FROM  { index-name-2 }
          { index-name-1 }        { literal-1    }

     { identifier-3 }
BY   {              }      UNTIL  condition-1
     { literal-2    }

[          { identifier-4 }        { identifier-5 }
[ AFTER    {              }  FROM  { index-name-4 }
[          { index-name-3 }        { literal-3    }

     { identifier-6 }                              ]
BY   {              }      UNTIL  condition-2      ]
     { literal-4    }                              ]
```

Figure 2

EXHIBIT CBL130-4

Title: Flow Charts of the Perform Statement Using the Varying Option with One and Two Conditions.

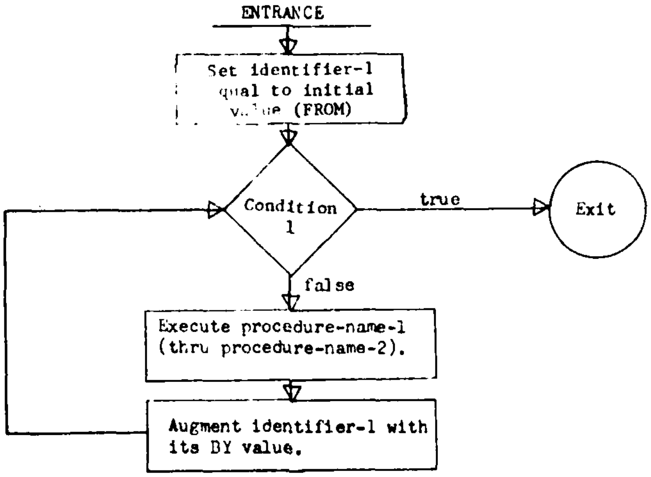

ENTRANCE
Set identifier-1 equal to initial value (FROM)
Condition 1
true
Exit
false
Execute procedure-name-1 (thru procedure-name-2).
Augment identifier-1 with its BY value.

Figure 1. One Condition

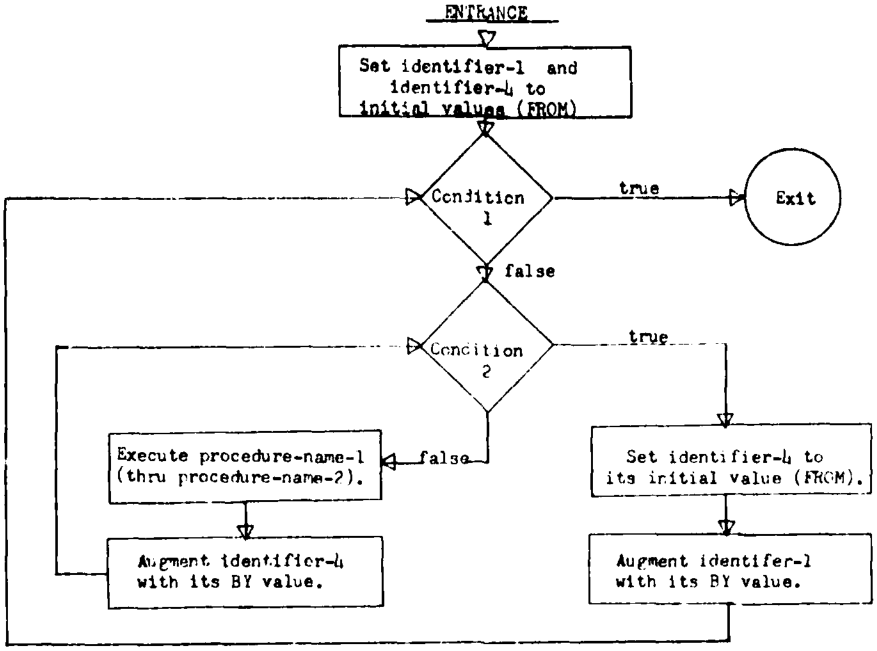

ENTRANCE
Set identifier-1 and identifier-4 to initial values (FROM)
Condition 1
true
Exit
false
Condition 2
true
false
Execute procedure-name-1 (thru procedure-name-2).
Augment identifier-4 with its BY value.
Set identifier-4 to its initial value (FROM).
Augment identifer-1 with its BY value.

Figure 2. Two Conditions

TITLE: Partial Program Using Perform Varying

```
.
.
.
FD  PRINTOUT   LABEL RECORDS ARE STANDARD.
01  PREC.
    03  FILLER         PIC X(10).
    03  COLUMNS.
        05  COL        PIC XXXXX  OCCURS 15.
03  FILLER             PIC X(47).
.
.
.
WORKING-STORAGE SECTION.
77  SUB-1              PIC 99.
77  SUB-2              PIC 99.
.
.
.
PROCEDURE DIVISION.
START.
    OPEN OUTPUT PRINTOUT.
    MOVE SPACES TO PREC.
A.
    PERFORM B VARYING SUB-1 FROM 1 BY 1 UNTIL SUB-1 EQUAL 16
    AFTER SUB-2 FROM 1 BY 1 UNTIL SUB-2 EQUAL 16.
    GO TO EJ.
B.
    MULTIPLY SUB-1 BY SUB-2 GIVING COL (SUB-2).
    IF SUB-2 EQUAL 15 WRITE PREC BEFORE 2.
EJ.
    CLOSE PRINTOUT.
    STOP RUN.
```

EXHIBIT CBL130-6

```
77  S-1                 PIC 9 COMP.
77  S-2                 PIC 9 COMP.
77  S-3                 PIC 99 COMP VALUE ZERO.

01  TAYBL               VALUE ZERO.
    03  T-A             OCCURS 5.
        05  T-B         PIC X OCCURS 5.

(START OF PROGRAM SEGMENT)
040-GO.
    PERFORM 050-GO THRU 060-GO VARYING S-1 FROM 1 BY 1
        UNTIL S-1 GREATER THAN 5
        AFTER S-2 FROM 5 BY -1 UNTIL S-2 EQUAL 0.
    GO TO 060-GO.
050-GO.
    IF S-2 = 3 GO TO 060-GO.
    IF S-1 = 3 GO TO 060-GO.
    MOVE S-2 TO T-B (S-1, S-2).
060-GO.
    ADD 1 TO S-3.
070-GO.
(END OF PROGRAM SEGMENT)
```

Partial Program Using Perform Varying

EXHIBIT CBL130-7

TITLE: Partial Coding for Final Project

```
    .
    .
    .
    FD   PRINTOUT  LABEL RECORDS STANDARD.
    01   PREC-1.
         03  P-SQ            PIC X(20).
         03  FILLER          PIC X(112).
    01   PREC-2.
         03  P-RANK          PIC XXX.
         03  FILLER          PIC X(7).
         03  P-GR            PIC ZZZ.
         03  FILLER          PIC X(119).
    WORKING-STORAGE SECTION.
    77   SQ-HOLD             PIC X(20).
    77   SW1                 PIC 9.
    77   SUB                 PIC 99.
    01   TAYBL-TEST.
         03  SQ-GR           OCCURS 4.
             05  T-SQ        PIC X(20).
             05  T-GR        PIC 999  OCCURS 9.
    01   TAYBL-RANK          VALUE "2LT1LTCPTMAJLTCCOL BG MGLTG".
         03  T-RANK          PIC XXX  OCCURS 9.
```

EXHIBIT CBL130-8

TITLE: Sample Sort Program

```
$      SNUMB   00999
$      IDENT   6ZU. . .
$      USERID  ABC$DEF/DNM
$      COBOL   NDECK
   010 IDENTIFICATION DIVISION.
   020 PROGRAM-ID.  SAMPLE.
   030 ENVIRONMENT DIVISION.
   040 CONFIGURATION SECTION.
   050 SOURCE-COMPUTER.  6000-EIS.
   060 OBJECT-COMPUTER.  6000-EIS.
   070 INPUT-OUTPUT SECTION.
   080 FILE-CONTROL
   090     SELECT INFILE ASSIGN AA-CARDS.
** 100     SELECT SFILE ASSIGN B1 B2 B3.
   110     SELECT OTFILE ASSIGN CC-LISTING.
   120 I-O-CONTROL.
** 130     APPLY STANDARD ON INFILE SFILE OTFILE.
   140 DATA DIVISION.
   150 FILE SECTION.
   160 FD  INFILE  LABEL RECORD STANDARD.
   170 01  INREC                PIC X(80).
   180 FD  OTFILE  LABEL RECORD STANDARD.
   190 01  OUTREC               PIC X(80).
** 200 SD  SFILE.
** 210 01  SREC.
** 220     03  FILLER           PIC X(10).
** 230     03  SSQDN            PIC X(8).
** 240     03  SNAME            PIC X(10).
** 250     03  SGRADE           PIC XX.
** 260     03  FILLER           PICX(50).
   270 PROCEDURE DIVISION.
   280 005-START SECTION.
   285 006-START-PARA.
** 290     SORT  SFILE  ASCENDING  SSQDN
** 291         DESCENDING  SGRADE  ASCENDING  SNAME
** 292         USING  INFILE  OUTPUT PROCEDURE  010-LIST.
   300     STOP RUN.
   310 010-LIST SECTION.
   320 015-OPEN.
   330     OPEN OUTPUT OTFILE.
   340 020-PROCESS.
** 350     RETURN SFILE INTO OUTREC  END GO TO 025-CLOSE.
   360     WRITE OUTREC BEFORE 2.
   370     GO TO 020-PROCESS.
   380 025-CLOSE.
   390     CLOSE OTFILE.
$      EXECUTE
$      DATA    AA
```

EXHIBIT CBL140-1

```
---detail cards fill this space---
$       FILE    B1,,3R        (or $ TAPE cards)
$       FILE    B2,,3R
$       FILE    B3,,3R
$       SYSOUT  CC
$       ENDJOB
***EOF
```

EXHIBIT CBL140-1 (Cont)

```
                        { DESCENDING }
SORT  file-name-1  ON  {            }  KEY data-name-1   [, data-name-2] ...
                        { ASCENDING  }

  [      { DESCENDING }                                          ]
  [ ON  {            }  KEY  data-name-3  [, data-name-4]  ...   ] ...
  [      { ASCENDING  }                                          ]

  { INPUT PROCEDURE  IS section-name-1 [ { THRU    }  section-name-2 ] }
  {                                    [ { THROUGH }                 ] }
  { USING file-name-2                                                  }

  { OUTPUT PROCEDURE IS section-name-3 [ { THRU    }  section-name-4 ] }
  {                                    [ { THROUGH }                 ] }
  { GIVING file-name-3                                                 }


                          { DESCENDING }
MERGE file-name-1  ON    {            }  KEY  data-name-1   [, data-name-2]
                          { ASCENDING  }

        [      { DESCENDING }                                            ]
        [ ON  {            }   KEY  data-name-3  [, data-name-4]   ...   ] ...
        [      { ASCENDING  }                                            ]

    USING  file-name-2,  file-name-3     [, file-name-4]    ...

  { OUTPUT PROCEDURE IS section-name-1    [ { THRU    } section-name-2 ] }
  {                                       [ { THROUGH }                ] }
  { GIVING  file-name-5                                                  }
```

EXHIBIT CBL140-2

TITLE: Sample Record Descriptions to be Used as Examples

	RECORD NAMES		
Column Numbers	Master Records	Detail-1	Detail-2
1 - 10	Name	Name	Name
11 - 20	Office	Office	Phone No.
21 - 30	Address	Week Pay	Office
31 - 40	Year Pay	Tax	Other Pay
41 - 42	Card ID	Card ID	Card ID
43 - 48		SOC SEC No.	Narrative
49 - 95			Narrative Continued
Record Length:	42 Characters	48 Characters	95 Characters
Words:	________	________	________

EXHIBIT CBL140-3

```
                                      Company Report (RH)
January 12, 1975 (PH)                                                  Page  001  (PH)

Dept 001  (CH)

                                      Sec 001  (CH)

Name (CH)                             Phone (CH)                       Yrs of Service (CH)

Adams, ...                            123-4567                         6
Jones, ...   (DE)                     890-1234  (DE)                   7  (DE)
Smith, ...                            567-8902                         8

                                      Total Yrs Sec  001 = 21  (CF)

                                      Sec  002  (CH)

                                      .
                                      .  (DE)
                                      .
                                      Total Yrs Sec  002 = ...  (CF)

                                      Total Yrs Dept  001 = ...  (CF)

Dept 002  (CH)

                                      Sec  001  (CH)

                                      .
                                      . (DE)
                                      .

Dept 003       )
                                      Sec 003  (CH)

                                      .
                                      .  (DE)
                                      .
                                      Total Yrs Sec  003 =  ...  (CF)

                                      Total Yrs Dept 003 =  ...  (CF)

                                      Total Yrs Company  =  ...  (CF Final)

                                      End Report (RF)
```

EXHIBIT CBL150-1

TITLE: Report Writer Abbreviations and their Definitions.

1. RH - Report Heading. The topmost report group that comes out only once, at the start of the report.

2. PH - Page Heading. Will be presented once at the top of every page. On the first page, it is second to the "RH" group.

3. CH - Control Heading. Presented at the top of a control when a control item changes. The control item may be any item in the File Section or Working-Storage Section. Control items are related to the report by a list of control data-names specified in the control(s) clause of the RD entry.

4. DE - Detailed Entry Group. The actual data that makes up a report.

5. CF - Control Footing. Printed at the end of a control. It is controlled by a data-name, same as the "CH".

6. PF - Page Footing. Presented at the end of every page in the report.

7. CF FINAL - Control Footing Final. Presented only once, upon execution of the Terminate Statement.

8. RF - Report Footing. Presented at the end of the report.

9. CH FINAL - Control Heading Final. Presented only once, upon the first execution of a Generate Statement.

10. OH - Overflow Heading. Presented when you are to repeat a heading that was presented on the previous page. You can never have both "PH" and "OH" on a given page. They are mutually exclusive.

11. OV - Overflow Footing. Presented when a footing needs to be repeated. A given report page can have either "PF" or "OV" but not both. If "OH" or "OV" are specified for a report, the RD entry must include the "Last Detail" phrase in the "Page Limits" clause.

EXHIBIT CBL150-2

TITLE: Sample Solution for Preparing Report in Exhibit CBL150-1

```
ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

.

.

.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
     SELECT INFILE ASSIGN FC ACCESS IS SEQUENTIAL.
     SELECT REPOUT ASSIGN OS-LISTING.
     SELECT SORDID ASSIGN A1.

I-O-CONTROL.

     APPLY SYSTEM STANDARD FORMAT ON INFILE SORDID.

DATA DIVISION.

FILE SECTION.

FD REPOUT LABEL RECORDS STANDARD REPORT IS COMREPORT.

FD INFILE LABEL RECORDS STANDARD.

01 INREC.
     02 DEPT PIC 999.
     02 SEC  PIC 999.
     02 NAM  PIC X(15).
     02 PHON PIC X(4).
     02 YSER PIC 99.

WORKING-STORAGE SECTION.

.

.

.

REPORT SECTION.

RD COMREPORT CONTROLS ARE FINAL DEPT SEC PAGE 62 LINES.
```

EXHIBIT CBL150-3

```
01 R-H TYPE RH
    02 LINE PLUS 1.
        03 COLUMN 60 PIC X(14) VALUE "COMPANY REPORT".
01 P-H TYPE PH.
    02 LINE PLUS 1.
        03 COLUMN 73 PIC X(4)  VALUE "PAGE".
        03 COLUMN 78 PIC Z(9)  SOURCE PAGE-COUNTER.
01 O-H TYPE OH NEXT GROUP PLUS 2.
    02 LINE PLUS 1.
        03 COLUMN 60 PIC XXX   VALUE "SEC".
        03 COLUMN 64 PIC 999   SOURCE SEC.
        03 COLUMN 68 PIC X(6)  VALUE "(CONT)".
        03 COLUMN 75 PIC X(4)  VALUE "PAGE".
        03 COLUMN 80 PIC Z9    SOURCE PAGE-COUNTER.
    02 LINE PLUS 2.
        03 COLUMN 40 PIC X(4)  VALUE "NAME".
        03 COLUMN 60 PIC X(5)  VALUE "PHONE".
        03 COLUMN 80 PIC X(11) VALUE "YRS SERVICE".
01 CH1 TYPE CH DEPT.
    02 LINE PLUS 5.
        03 COLUMN 30 PIC X(4)  VALUE "DEPT".
        03 COLUMN 35 PIC 999   SOURCE DEPT.
01 CH2 TYPE CH SEC NEXT GROUP PLUS 2.
    02 LINE PLUS 5.
        03 COLUMN 60 PIC XXX   VALUE "SEC".
        03 COLUMN 64 POIC 999  SOURCE SEC.
    02 LINE PLUS 2.
```

EXHIBIT CBL150-3 (Cont)

```
            03 COLUMN 40 PIC X(4)  VALUE "NAME".
            03 COLUMN 60 PIC X(5)  VALUE "PHONE".
            03 COLUMN 80 PIC X(11) VALUE "YRS SERVICE".
    01 D-LINE TYPE DE.
        02 LINE PLUS 1.
            03 COLUMN 35 PIC X(15) SOURCE NAM.
            03 COLUMN 60 PIC X(4)  SOURCE PHON.
            03 COLUMN 84 PIC 99    SOURCE YSER.
    01 CF2 TYPE CF SEC.
        02 LINE PLUS 2.
            03 COLUMN 64 PIC X(13) VALUE "TOTAL YRS SEC".
            03 COLUMN 78 PIC 999   SOURCE SEC.
            03 COLUMN 82 PIC X     VALUE "=".
            03 SUMSEC COLUMN 84 PIC ZZZ9 SUM YSER.
    01 CF1 TYPE OF DEPT.
        02 LINE PLUS 2.
            03 COLUMN 63 PIC X(14) VALUE "TOTAL YRS DEPT".
            03 COLUMN 78 PIC 999  SOURCE DEPT.
            03 COLUMN 82 PIC X     VALUE "=".
            03 SUM-DEPT COLUMN 84 PIC ZZZ9 SUM SUMSEC.
    01 CFF TYPE CF FINAL.
        02 LINE PLUS 2.
            03 COLUMN 64 PIC X(17) VALUE "TOTAL YRS COMPANY".
            03 COLUMN 82 PIC X     VALUE "=".
            03 COLUMN 84 PIC ZZZZ9 SUM SUM-DEPT.
    01 R-F TYPE RF.
        02 LINE PLUS 3.
```

EXHIBIT CBL150-3 (Cont)

```
          03 COLUMN 60 PIC X(10) VALUE "END REPORT".
    PROCEDURE DIVISION.
    START.
    SORT SORDID ASCENDING DEPT SEC USING INFILE
        OUTPUT PROCEDURE DOIT.
    STOP RUN.
    DOIT SECTION.
    BEGIN.  OPEN OUTPUT REFOUT.
    INITIATE COMREPORT.
    AA.
        RETURN SORDID INTO INREC AT END GO TO FINI.
        .
        .
        .
        .
        GENERATE D-LINE.
        GO TO AA.
    FINI.  TERMINATE COMREPORT.
        CLOSE REPOUT.
```

EXHIBIT CBL150-3 (Cont)

TITLE: Report Writer Problem

GENERAL INSTRUCTIONS: Sort the input records in ascending sequence with DEPT-NUMBER then EMPLOYEE-NUMBER as the sort fields. Using Report Writer, produce a report as follows: a. A maximum of 40 lines per page (including heading). b. Calculate grand totals for all monetary columns. c. Calculate subtotals for each Department.

SPECIFIC INSTRUCTIONS:

1. If WEEK equals 1, 14, 27, or 40, the beginning of a new quarter is indicated, so zero out QTRFICA field before starting new accumulation.

2. FICA = 4.8 percent of TAXABLE-GROSS. TAXABLE-GROSS is that part of the WKGROSS which does not force YTDGROSS to exceed $6600.00. (No earnings over $6600.00 are taxable.) If there is no FICA to be deducted. store zeros in that field.

3. FIT (Federal Income Tax) = (WKGROSS - (TAXCLASS * 13.00)) * 18%. Use the COMPUTE Verb and numeric literals where possible. If the deduction amount (TAXCLASS * 13.00) exceeds WKGROSS, no tax is paid, so store zeros in that field (FIT).

4. NET = WKGROSS - FIT - FICA. (Use SUBTRACT Verb.)

5. Update the cumulative totals: YTDGROSS, YTDFIT and QTRFICA.

6. Calculate average weekly gross pay. (Use the DIVIDE Verb.)

7. Edit information into output field for print and allow for overflow on report.

Data Card Format:

Columns	Field Name:
1 - 2	Week Number
3 - 8	Employee Number
9 - 10	Tax Class
11 - 17	Year to Date Gross
18 - 23	Year to Date FIT
24 - 28	Quarterly FICA
29 - 35	Week Gross
36	Dept Number
37 - 80	Blank

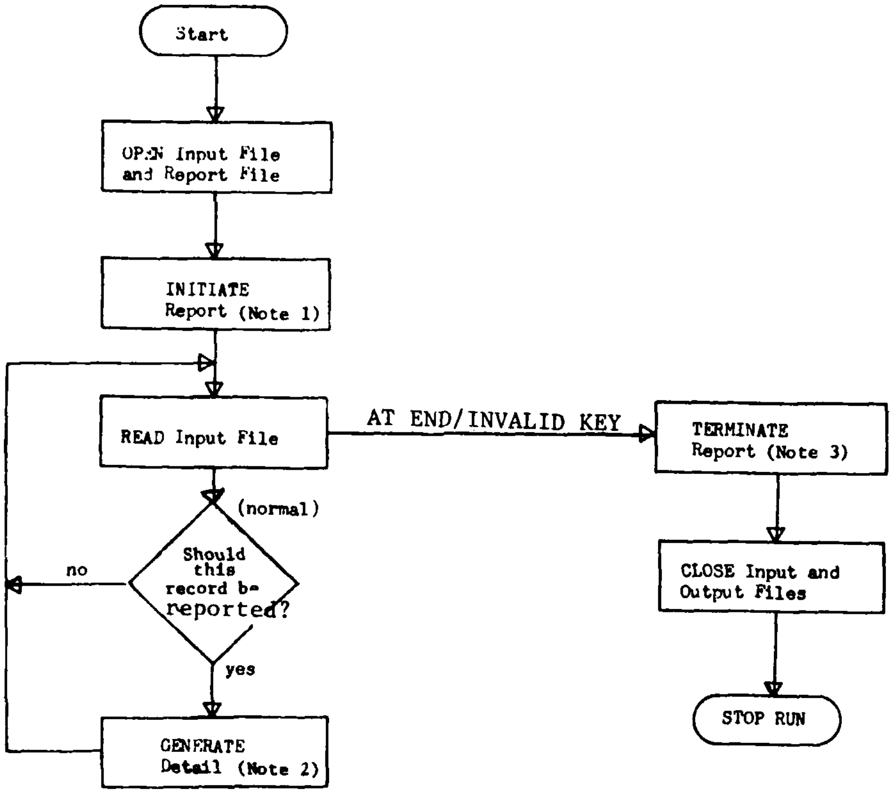
EXHIBIT CBL150-4

The printed report should look like this:

```
                              XYZ COMPANY
                            PAYROLL REGISTER

January 31, 1975                                               Page 001

DEPT ...

EMP# YTD-GROSS YTD-FIT QTRFICA GROSS-PAY AVG-GROSS FIT FICA NET

.
.
.
TOTAL DEPT  ...  $
.
.
.
TOTAL FOR COMPANY (Last Line)  ...  $
```

EXHIBIT CBL150-4 (Cont)

TITLE: Report Writer Relationships between INITIATE, GENERATE and TERMINATE.


Start
OPEN Input File and Report File
INITIATE Report (Note 1)
READ Input File
AT END/INVALID KEY
TERMINATE Report (Note 3)
(normal)
Should this record be reported?
no
yes
CLOSE Input and Output Files
STOP RUN
GENERATE Detail (Note 2)

NOTES:

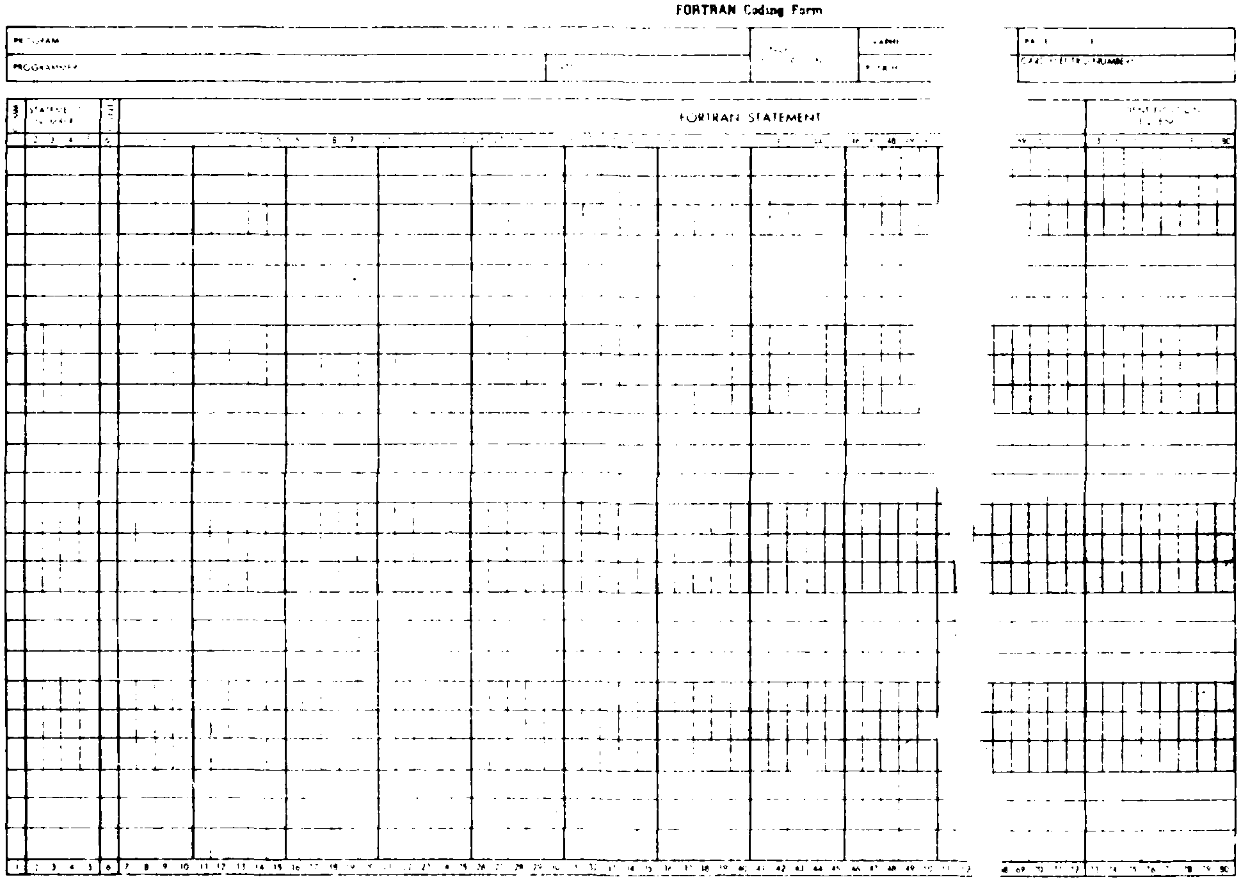
1. INITIATE establishes initial housekeeping values and prints report headings.

2. GENERATE causes tests for control breaks and page breaks, with appropriate actions, as well as detail line presentation and total accumulation.

3. TERMINATE causes final footings and totals to be printed.

EXHIBIT CBL150-5

FORTRAN PROGRAMMING (H6000-CDT) COURSE EXHIBITS



FORTRAN Coding Form

PROGRAM

PROGRAMMER

CARD ELECTRO NUMBER

FORTRAN STATEMENT

Comment (Col 1)
Sequence (Col 1-5)
Continuation (Col 6)

Programming Area (Col 7-72)
Identification Area (Col 73-80)

EXHIBIT FTN010-1

Keyword: FORTRAN

	SAMPLE		PRACTICE
1.	$a(b+c)$	1.	$(b+c)a$
2.	$\frac{ab}{c}$	2.	$\frac{ac}{b}$
3.	$\frac{a}{bc}$	3.	$\frac{c}{ab}$
4.	$a+bc$	4.	$ba+c$
5.	$(a+b)^2$	5.	$(a+c)^5$
6.	a^{n+1}	6.	c^{x-3}
7.	$\frac{a}{b+c}$	7.	$\frac{9}{a+x}$
8.	$a-(b-c)$	8.	$b-(c-a)$
9.	$a(-b)$	9.	$x(-y)$
10.	$a(b + \frac{1}{c+d})$	10.	$b\ (8 + \frac{1}{9+a})$

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

EXHIBIT FTNO 30-1

MODE	MODE	RESULT MODE
DP	DP	DP
DP	R	DP
DP	I	DP
R	DP	DP
R	R	R
R	I	R
I	I	I
I	R	R
I	DP	DP

EXHIBIT FTN041-1

".AND." TRUTH TABLE

IF A=	IF B=	THEN A.AND.B=
.TRUE.	.TRUE.	.TRUE.
.TRUE.	.FALSE.	.FALSE.
.FALSE.	.TRUE.	.FALSE.
.FALSE.	.FALSE.	.FALSE.

".OR." TRUTH TABLE

IF A=	IF B=	THEN A.OR.B=
.TRUE.	.TRUE.	.TRUE.
.TRUE.	.FALSE.	.TRUE.
.FALSE.	.TRUE.	.TRUE.
.FALSE.	.FALSE.	.FALSE.

".NOT." TRUTH TABLE

.NOT. .TRUE. = .FALSE.
.NOT. .FALSE. = .TRUE.

EXHIBIT FTN050-1

HIERARCHY OF ARITHMETIC OPERATORS

( )	PARENTHESES	1ST
**	EXPONENTIATION	2ND
* /	MULTIPLICATION/DIVISION	3RD
+ -	ADDITION/SUBTRACTION	4TH

HIERARCHY OF CONNECTIVE OPERATORS

( )	1ST
.NOT.	2ND
.AND.	3RD
.OR.	4TH

EXHIBIT FTN050-2

```
1     7
   30 READ(5,10)VALUE
       .
      IF(L1)GOTO 100
      X=0.0
      GO TO 30
  100 X=VALUE+10
```

EXHIBIT FTN050-3

RELATIONAL OPERATORS	MEANING
.LT.	LESS THAN
.EQ.	EQUAL
.GT.	GREATER THAN
.GE.	GREATER OR EQUAL
.LE.	LESS THAN OR EQUAL
.NE.	NOT EQUAL

Relational Operators Used in FORTRAN

EXHIBIT FTN050-4

```
INITIALIZE              I=0
                         .
                         .
INCREMENT            80 I=I+10
                         .
                         .
                         .       (LOOP)
TEST                    IF(I.LT.100)GO TO 80
                     90 STOP
                        END
```

Sample Conditions for Controlled Loops

EXHIBIT FTN050-5

```
             1     7
(A)                DATA ICOUNT/0/
(B)                DATA PERCNT/.75/
(C)             30 READ(5,10)VALUE
(D)             10 FORMAT(F5.2)
(E)                PRICE=VALUE*PERCNT+VALUE
(F)                WRITE(6,20)PRICE
(G)             20 FORMAT(1HO,F7.2)
(H)                ICOUNT=ICOUNT+1
(I)                IF(ICOUNT.EQ.50)GO TO 40
(J)                GO TO 30
(K)             40 STOP
(L)                END
```

Sample Program Using Data Statements

EXHIBIT FTN050-6

```
1     7

  100 READ(5,10,END=999)EMPNUM,YRSEXP,PAY
   10 FORMAT(F5.2,F5.2,F6.2)
      IF(YRSEXP-5.)100,200,300
  200 ADJPAY=PAY+5.00
      GO TO 400
  300 ADJPAY=PAY+10.00
  400 WRITE(6,20)EMPNUM,ADJPAY
   20 FORMAT(1HO,13HEMPLOYEE NO. ,F5.2,15H ADJUSTED PAY=,F6.2)
      GO TO 100
  999 STOP
      END
```

Sample Program

EXHIBIT FTN050-7

FIELD NAME	CARD COLUMNS
EMPID (EMPLOYEE ID)	1-6
SALARY	9-16
ICODE (CODE FOR YEARS EXPERIENCE)	19

CODE	MEANING
0	0 TO 3 YEARS EMPLOYMENT
1	4 TO 5 YEARS EMPLOYMENT
2	6 TO 10 YEARS EMPLOYMENT
3	OVER 10 YEARS EMPLOYMENT

YEARS EMPLOYMENT	ICODE	PERCENTAGE
0 TO 3	0	NO INCREASE
4 TO 5	1	.01
6 TO 10	2	.03
OVER 10	3	.05

Format for Data Card File

EXHIBIT FTN050-8

```
1     7
    1 READ(5,10,END=999)EMPID,SALARY,ICODE
   10 FORMAT(F6.2,2X,F8.2,2X,I1)
      IF(ICODE.EQ.0)ICODE=ICODE+4
      GO TO(100,200,300,50),ICODE
   50 PERCNT=0.0
      GO TO 400
  100 PERCNT=.01
      GO TO 400
  200 PERCNT=.03
      GO TO 400
  300 PERCNT=.05
  400 SALINC=SALARY*PERCNT+SALARY
  500 WRITE(6,20)EMPID,SALARY,SALINC,PERCNT
   20 FORMAT(1H0,13HEMPLOYEE NO. ,F6.2,T26,12HOLD SALARY= ,F8.2,T47,
     112HNEW SALARY= ,F8.2,T80,17HPERCENT INCREASE= ,F4.2)
      GO TO 1
  999 STOP
      END
```

Sample Program

EXHIBIT FTN050-9

```
1     7
5     ASSIGN 10 TO K
100   READ(5,4000,END=999)RECORD
      GO TO K,(10,20,30,40)
10    PAY=RECORD*.05+RECORD
      ASSIGN 20 TO K
      GO TO 100
20    DEDUK=RECORD*750.0
      ASSIGN 30 TO K
      GO TO 100
30    VPAY=RECORD*PAY/30.
      ASSIGN 40 TO K
      GO TO 100
40    EMPNUM=RECORD
      WRITE(6,5000)EMPNUM,PAY,VPAY,DEDUK
5000  FORMAT(1X,8HEMP. NO.,F8.0,T22,9HNEW PAY $,F8.2,T50,8HVACATION,
      *6H PAY $,F8.2,T40,3HTAX,12HDEDUCTIONS $,F8.2)
      GO TO 5
999   STOP
      END
```

Use of Assigned Go To Statement

EXHIBIT FTN050-10

Sample Program

```
1     7
      I=0
1     I=I+1
2     READ(5,10)INDEX
      WRITE(6,10)INDEX
10    FORMAT(I5/)
      IF(I.EQ.10)GO TO 20
      GO TO 1
20    STOP
      END
```

Sample Program with Single Dimension Array and Data Statement Added

```
1     7
      DIMENSION ISCORE(6)
      DATA ISCORE/70,85,90,76,82,85/
      K=1
5     WRITE(6,10)ISCORE(k),K
10    FORMAT(1HO,7HISCORE=,1X,I2,3X,2HK=,I1)
      K=K+1
      IF(K.GT.6)GO TO 20
      GO TO 5
20    STOP
      END
```

EXHIBIT FTN060-1

CONSTRUCTING FUNCTION SUBPROGRAMS

DO'S

1. Use the FUNCTION statement as the first statement of the function subprogram.

   WHY? So the compiler will recognize that the data which follows is a function subprogram unit with the name and type indicated in the FUNCTION statement and with the number and type of arguments required.

2. Use the function name as a resultant variable at least once in the function subprogram.

   WHY? To insure that the correct result is returned to the calling program.

3. Use at least one RETURN statement in the function subprogram.

   WHY? So the calling program can resume processing with the result of the function.

4. Use an END statement as the last statement of the function subprogram.

   WHY? So the compiler can recognize the physical end of the function subprogram unit.

DON'Ts

1. Don't use the name of the function subprogram in any nonexecutable statement other than the FUNCTION statement.

   WHY? It will cause numerous problems when compiling your subprogram. Also, it is not possible to "dimension" the function name.

2. Don't use names of the dummy arguments in the EQUIVALENCE, COMMON, or DATA statements.

   WHY? Since you are passing information to the function subprogram via the dummy arguments, it would be rather futile to try to initialize their values in the DATA statement since they would be changed at the first function call.

3. Don't use the SUBROUTINE, BLOCK DATA, or FUNCTION statements within a function subprogram.

   WHY? The compiler will certainly reject the whole subprogram since it would not know what to do. You will learn how to construct the SUBROUTINE and BLOCK DATA statements later in the lesson.

EXHIBIT FTN080-1

PROBLEM ON FUNCTION SUBPROGRAM

```
10  END
11  OBOY(A,B,C)=A+B+C
12  DATA OMEN/3.7/
13  INTEGER FUNCTION OMEN(A,B,C)
14  RETURN
15  OMEN=OBOY(A,B,C)+1
```

EXHIBIT FTN080-2

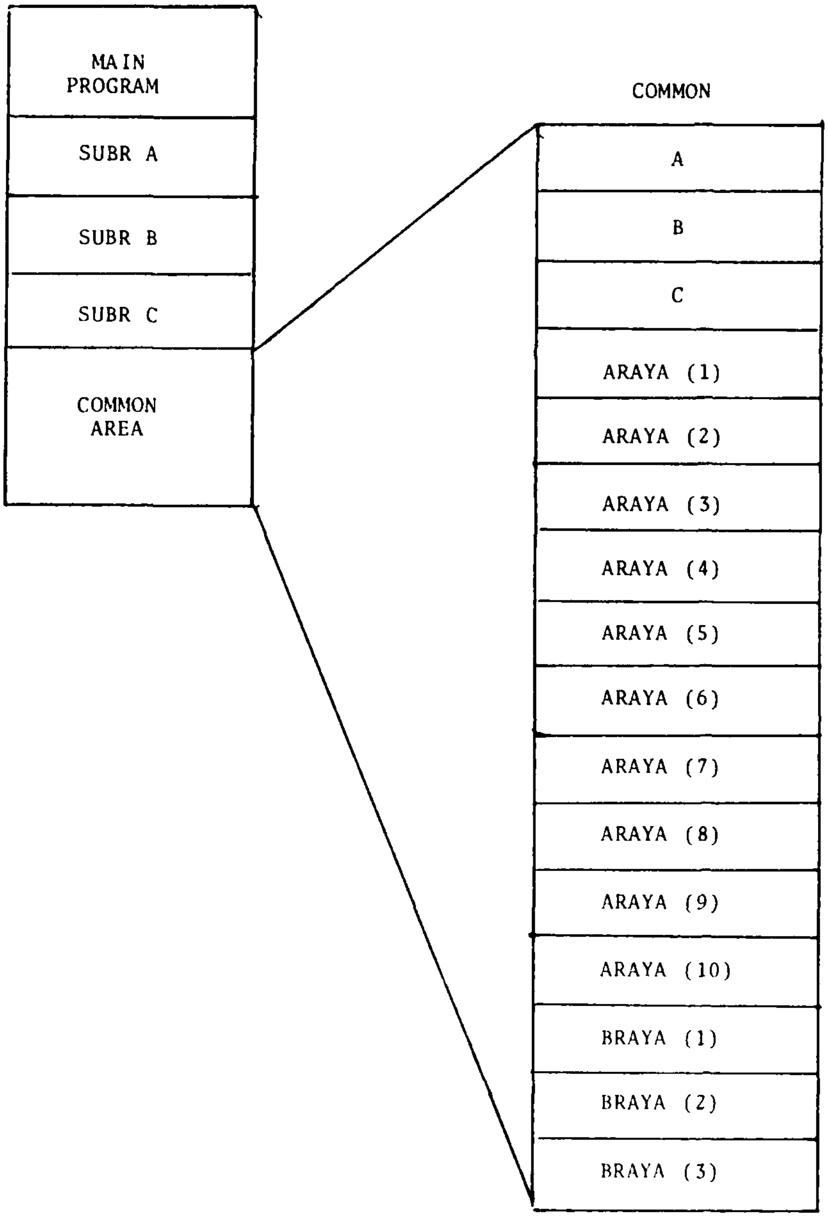
PROBLEM USING HOLLERITH CONSTANT

```
101 RESULT=EXAM(3HABC,B,C)
                   .
                   .
    END
                   .
                   .
    FUNCTION EXAM(A,B,C)
    CHARACTER A
    IF(A .EQ. 3HABC) GO TO 10
    RETURN
    END
```

EXHIBIT FTN080-3

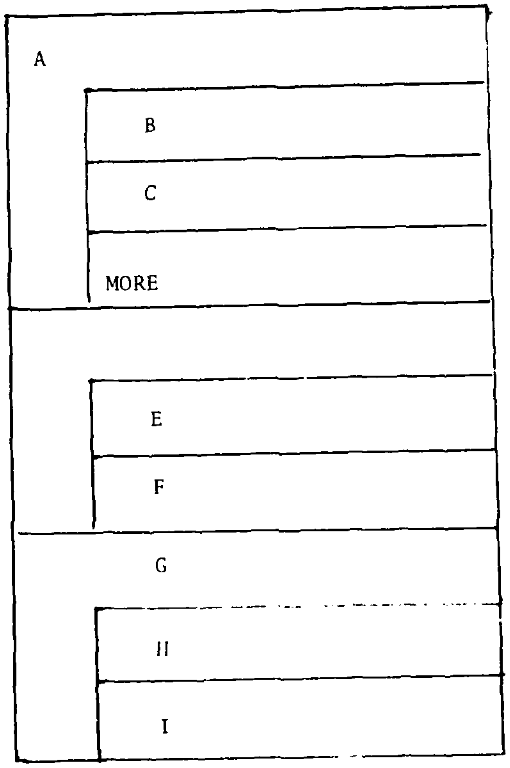
```
DIMENSION A(10)
          .
I=50
MLIN=ARAY(A,I)
          .
END
          .
FUNCTION ARAY(X,I)
DIMENSION X(100)
ARAY = X(I)
RETURN
END
```

EXHIBIT FTN080-4



Object Program with Three Subprograms in Common Area

EXHIBIT FTN080-5



Data in Common Area after Completion

EXHIBIT FTN080-6

RULES FOR CODING A "BLOCK DATA" PROGRAM

1. The first statement in the BLOCK DATA subprogram must be BLOCK DATA.

2. Since we are only intitializing whole blocks of values, there cannot be any executable instructions in the BLOCK DATA subprogram. This leaves us with the type EQUIVALENCE, DATA, COMMON, and DIMENSION statements.

3. You can initialize more than one common block in a BLOCK DATA subprogram.

4. You must initialize all elements in a common block, not a portion of the elements present. In addition, all elements in the common statement of the main program must be listed in a common statement within the BLOCK DATA subprogram.

5. Variables in BLOCK DATA statements must be in the same order as in the COMMON statement in the main program.

6. There may be as many BLOCK DATA subprograms as necessary in a program.

EXHIBIT FTN080-7

```
DIMENSION ARAY(3)
EQUIVALENCE(ARAY(1),TIME),(ARAY(2),DATA),(ARAY(3),BAL)
                      .
                      .
READ(5,10)ARAY
                      .
                      .
WRITE(6,12)TIME,DATE,BAL
```

EXHIBIT FTN080-8

Effects of Equivalencing in Common Area

PROPERLY
EXTENDED
COMMON

ARAY (1)	
ARAY (2)	
ARAY (3)	
ARAY (4)	BRAY (1)
ARAY (5)	BRAY (2)
DATA	BRAY (3)
	BRAY (4)
	BRAY (5)
	BRAY (6)

COMMON AREA

IMPROPERLY EQUIVALENCED
COMMON

	BRAY (1)
	BRAY (2)
ARAY (1)	BRAY (3)
ARAY (2)	BRAY (4)
ARAY (3)	BRAY (5)
ARAY (4)	BRAY (6)
ARAY (5)	
DATA	

COMMON AREA

EXHIBIT FTN080-9

RESTRICTIONS FOR A "SUBROUTINE" SUBPROGRAM

1. The subroutine subprogram may contain any statement except the FUNCTION statement, the BLOCK DATA statement or another SUBROUTINE statement. The compiler will check for this type of error and will flag it for you.

2. The subroutine subprogram may redefine or define one or more of its arguments to return results to the calling program unit.

   Example:

```
SUBROUTINE CHANGE(THIS,IS,AN,EXAM)
EXAM=THIS/FLOAT(IS)
IF(EXAM .GE. AN)EXAM=0
IF(EXAM .LT. AN)EXAM=1
RETURN
END
```

3. The argument names may not be used in EQUIVALENCE, COMMON, or DATA statements. This is to avoid redundancy. Since arguments are used for communicating between the main-line program and the subroutine, their use in any of the above statements is not consistent.

4. A RETURN statement must be in the "subroutine" subprogram. The RETURN statement is used to return control back to the main-line program.

5. The final statement in the "subroutine" subprogram must be an END statement.

EXHIBIT FTN080-10

SAMPLE SUBROUTINE PROGRAM

```
SUBROUTINE TEST(A1,A2,A3,INT1)
CHARACTER A2
IF(A2 .EQ. 3HECH) RETURN
A1=A3/FLOAT(INT1)
RETURN
END
```

EXHIBIT FTN080-11

EXAMPLES OF ENTRIES INTO A SUBPROGRAM

CALLING PROGRAM

```
          .
          .
10 CALL SUBA (V,W,X,Y,Z)
          .
          .
11 CALL SUBB (S,T,V,Z)
          .
          .
12 CALL SUBC (X,Y,W)
          .
          .
   END
```

SUBPROGRAM TO BE CALLED

```
SUBROUTINE SUBA(A,B,C,D,E)
       .
       .
F=A+B
       .
ENTRY SUBB(A,D,E,B)
       .
       .
GO TO 9
       .
       .
ENTRY SUBC(A,B,C)
       .
       .
RETURN
END
```

EXHIBIT FTN080-12

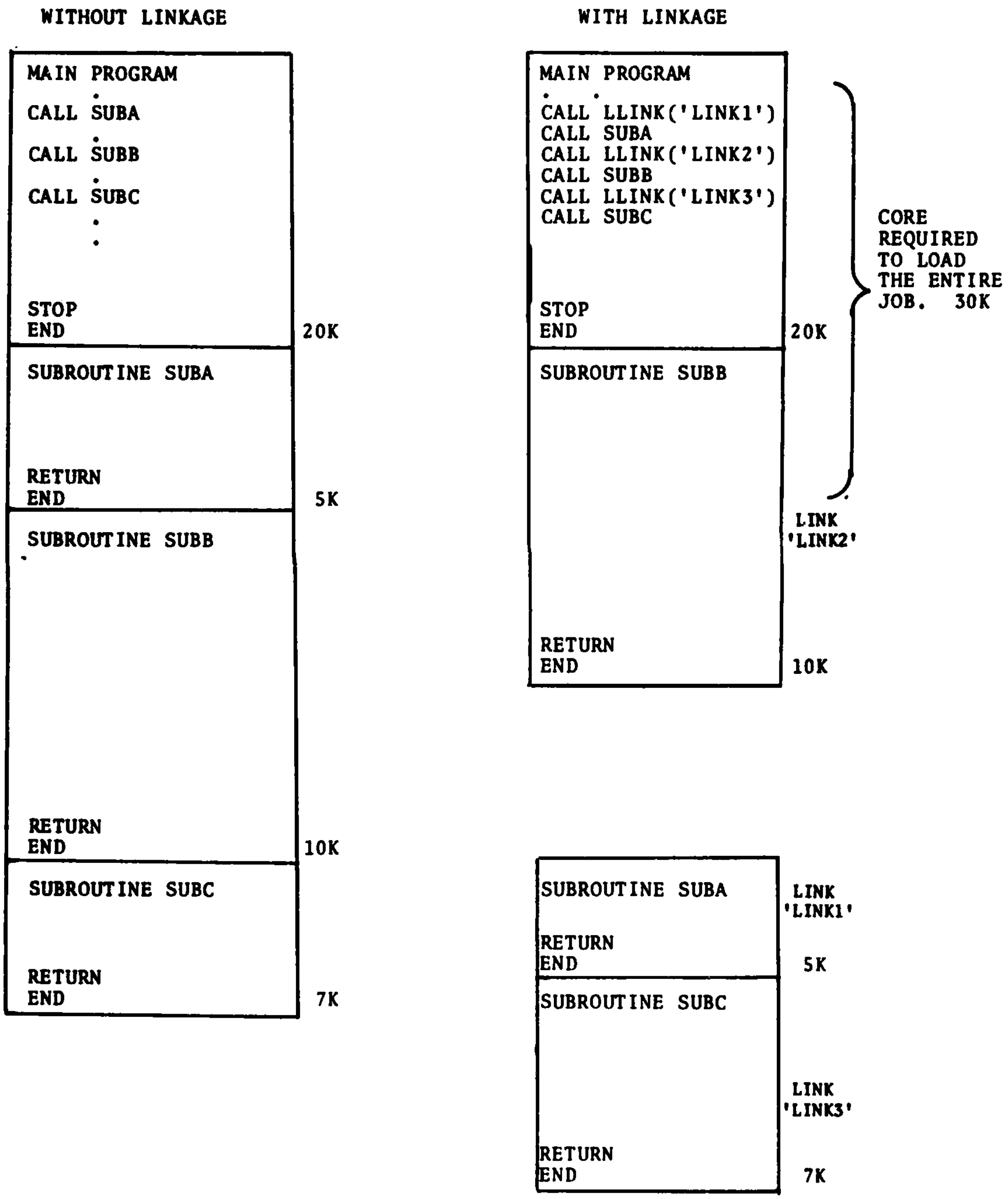
GENERAL FORM OF RETURNS IN SUBPROGRAMS

```
   CALLING PROGRAM                            CALLED SUBPROGRAM
          .                                   SUBROUTINE ABLE (X,Y,Z,*,*)
          .                                             .
          .                                             .
10 CALL ABLE(A,B,C,$20,$30)                             .
11 -----                                  100 IF (R) 200,300,400
          .                               200 RETURN
          .                               300 RETURN1
          .                               400 RETURN2
20 -----                                      END
          .
          .
30 -----
          .
          .
   END
```

EXHIBIT FTN080- 13

```
                        1
1         8             6

$         USERID
$         IDENT
$         FORTY    NDECK
$         OPTION   FORTRAN
             .
          (MAIN PROGRAM)
             .
$         LINK     LINK1
$         FORTY    NDECK
$         OPTION   FORTRAN
             .
          (SUBROUTINE SUBA)
             .
$         LINK     LINK3,LINK1
$         FORTY    NDECK
$         OPTION   FORTRAN
             .
          (SUBROUTINE SUBC)
             .
$         LINK     LINK2, LINK3
$         FORTY    NDECK
$         OPTION   FORTRAN
             .
          (SUBROUTINE SUBB)
             .
$         EXECUTE
$         SYSOUT   06
$         DATA     05
             .
          (CARD DATA IF ANY)
             .
$         ENDJOB
***EOF
```

EXHIBIT FTN090-1



```
MAIN PROGRAM
     .
CALL SUBA
     .
CALL SUBB
     .
CALL SUBC
     .
     .

STOP
END
```

```
SUBROUTINE SUBA

RETURN
END
```

```
SUBROUTINE SUBB

RETURN
END
```

```
SUBROUTINE SUBC

RETURN
END
```

```
MAIN PROGRAM
 .   .
CALL LLINK('LINK1')
CALL SUBA
CALL LLINK('LINK2')
CALL SUBB
CALL LLINK('LINK3')
CALL SUBC

STOP
END
```

```
SUBROUTINE SUBB

RETURN
END
```

```
SUBROUTINE SUBA

RETURN
END
```

```
SUBROUTINE SUBC

RETURN
END
```

EXHIBIT FTN100-1

JOINT OPERATION PLANNING SYSTEM (JOPS) III USER (H6000-CDT) COURSE EXHIBITS)

SECTION 1--DEFINITIONS OF COMMONLY USED TERMS

The following is a list of selected terms used in the Joint Planning Community along with a practical definition of each. Exhibits JOP005-1 through JOP015-1 are for use with the new JOPS course. (Section II, a table of acronyms, follows this list of definitions.)

ACCOMPANYING SUPPLIES

Supplies in addition to those contained in TUCHA that the supported commander or service Headquarters deploy with a unit.

ACTUAL PLANNING

Actual planning involves the concept of identifying actual units to fulfill plan requirements at the time the force list is initially established. This identification is accomplished by assigning Unit Identification Codes (UIC) to each requirement listed in the force list for an OPLAN.

ADEQUACY

Operation plan review criterion. The scope and concept of planned operations are sufficient to accomplish the task assigned.

ALERT ORDER

A formal directive issued by the JCS. It reflects an NCA decision that US military forces may be required, provides essential guidance for planning in the prevailing situation, and marks the onset of execution planning.

ANNEXES

Documents appended to the basic plan or order to make it clearer or to give further details.

ASSIGNED FORCES

Forces in being that have been placed under the operational command or control of a commander.

ATTACHMENT TO AN OPERATION PLAN OR ORDER

An attachment to an operational plan or order is a separately identifiable amplification of the basic plan or operation order. Attachments are annexes, appendices, tabs, and enclosures.

AUGMENTATION FORCES

Forces to be transferred to the operational command of a supported commander during the execution of an operation.

AVAILABLE TO LOAD DATE (ALD)

The date specified for each unit in an operation plan which indicates when that unit will be available for loading at the port of embarkation (POE).

BASIC PLAN

That part of an operation plan which forms the base structure for annexes and appendices. It consists of general statements related to the situation,

EXHIBIT JOP005-1

mission, execution, logistics, administration, and command and signal.

C-DAY

The unnamed day on which deployment operations commence.

CIVIL ENGINEERING SUPPORT PROGRAM (CESPG)

CESPG is a JOPS III subsystem. The purpose of the CESPG is to compute facility, construction force, and gross material requirements to support joint operation plans.

CIVIL ENGINEERING SUPPORT PLANNING

That part of operation plan development which concerns the improvement or expansion of resources and facilities in the area of operations.

CONPLAN

An operation plan in a abbreviated concept format which would require expansion into an OPLAN or OPORD prior to implementation.

CONCEPT OF OPERATIONS

A verbal or written statement, in broad outline, of a commander's assumptions or intent in regard to an operation or series of operations. The purpose of the concept is to give an overall view of the operation.

CRISIS ACTION SYSTEM (CAS)

CAS provides guidance and procedures for the conduct of joint planning for the use of military forces during emergency or time-sensitive situations. The procedures provide the Joint Chiefs of Staff with information to develop timely recommendations to the National Command Authorities (NCA) for decisions involving the use of US military forces.

DATA BASE

A data base is a collection of one or more files which represent all the data associated with or supporting the objective of an ADP system. For example, in JOPS, one data base supports each OPLAN. It is called the TPFDD/SRF. It consists of two files; the TPFDD (which contains force records, nonunit-related cargo records, and nonunit-related personnel records) and the Summary Reference File (which contains up to nine different records which house analysis results and data base management efforts for the OPLAN).

DATA ELEMENT

The smallest intelligible piece of information relating to an overall topic. For example, in a personnel system, NAME and SSAN could be data elements. Sometimes the contents of the data elements may be a code and not the complete data itself. For example, JOPS uses the code "A" as a one-character data element to represent the transportation mode of "AIR." To identify the

EXHIBIT JOP005-1 (Cont)

Military Airlift Command (MAC) as the source of transportation, another character data element is used in which the code "K" is stored.

D-DAY

The unnamed day on which an operation plan is executed.

DELIBERATE PLANNING

Operation planning as a result of JSCP or other tasking directive using JOPS I, II, and III procedures and concerns the relocation of forces to the desired area of operation.

DEPLOYABILITY POSTURE

The state or stage of a unit's preparedness for deployment to participate in a military operation.

DEPLOYMENT DATA BASE

The JDS data base containing the necessary information on forces, materiel, filler, and execution. The data base reflects information contained in the refined TPFDD or data developed during the various phases of the Crisis Action System and the movement schedules or tables developed by the TOAs to support the deployment of required forces, personnel, and materiel.

DEPLOYMENT PLAN

A plan that supports the deployment phase of a supported commander's OPLAN.

DEPLOYMENT PLANNING

Operation planning as a result of Joint Strategic Capabilities Plan (JSCP) or other tasking directive using JOPS I, II, and III procedures and concerns the relocation of forces to the desired area of operation.

DESTINATION

The terminal geographic location in the routing scheme. Nonunit-related transportation requirements will not use the destination as part of its routing. The destination identifies the station or location in the objective area at which the unit will be deployed.

DIRECT ACCESS COMMUNICATION (DAC)

A system on the H6000 computer which allows a user to communicate with a computer program through a remote terminal. The standard JOP III software has been written to use this method of access. Each user must initiate the JOPS programs and then connect to them using the DAC system. Under DAC control, each user on the JOP III subsystem requires 38-40K of memory. This severely restricts the number of users that can efficiently use the resources simultaneously.

DISK STORAGE

A method of storing information on a device patterned after phonograph records. They are stacked, however, to provide more surface for storage. Each disk plane is recorded by tracks which are partitioned into sectors. Each sector can be magnetized to represent characters of information. Disk storage has much faster access and transfer rate than tape storage. Moreover, the capability to access data in a random manner permits direct retrieval of data rather than a sequential search of each record until the desired one is located. In JOPS, all data files and data bases are on disk storage devices during processing. However, tapes are used for permanent storage of OPLAN unique files between processing sessions. Thus the advantages of both can be attained; tapes provide a cheaper means of permanently storing information, and during processing this data is restored to disk thus giving the faster processing capability.

EARLIEST ARRIVAL DATE (EAD)
The data specified at the port of debarkation that establishes the earliest date that a unit, resupply requirement, or replacement personnel requirement can be accepted during a deployment. Used with the latest arrival date (LAD) to define a time frame window for transportation purposes.

ELEMENTS OF AN OPLAN/OPORD
An element of an OPLAN or OPORD is an item which is listed in the table of contents including attachments.

EMPLOYMENT PLANNING
That part of operation planning which concerns the strategic or tactical use of forces and material within the area of operations.

ESTIMATED DEPARTURE DATE (EDD)
A date computed by the TFE simulation module for each transportation requirement which indicates the date of departure from the POE used during the simulation. This date will reflect the earliest date, after the available to load (ALD) date at the POE, which the simulation module could load and depart with the complete requirement.

EXECUTION PLANNING
The phase of the Crisis Action System (JOPS IV) planning in which an approved operation plan or other NCA-designated course of action is adjusted and refined, as required by the prevailing situation, and converted into an OPORD that can be executed at a designated time. Execution planning can proceed on the basis of prior deliberate planning, or it can take place under a NOPLAN situation. It includes adapting the plan or concept to the prevailing circumstances, the designation of units to satisfy force requirements, the establishment of appropriate deployability posture, the scheduling of necessary transportation

resources, and the dissemination of movement tables to regulate the deployment of forces requiring common user transportation.

FEASIBLE ARRIVAL DATE (FAD)

A date computed by the TFE simulation module which indicates for each transportation requirement the earliest date after the designated EAD that the shipment could be closed. The FAD is used to determine transportation shortfalls. When the FAD is later than the LAD, a transportation shortfall exists.

FEASIBILITY

Operation plan review criterion.

1. The assigned tasks could be accomplished by implementing the plan.

2. The forces, transportation, and logistic resources made available for planning can be provided for execution.

3. Support required for departments and agencies other than the Department of Defense can be provided.

FILE

A file is a collection of one or more types of records which are related for some overall topic. In JOPS, for example, the TPFDD file is made up of force records, nonunit-related cargo records, and nonunit-related personnel records. Together this information represents the overall time-phased transportation required.

FILLER

One of a number of individuals, officer or enlisted, required to bring a unit, organization, or approved allotment to authorized strength.

FORCE LIST

A list of forces expressed as type units, consisting of the total forces required by an operation plan with assigned forces, augmentation forces, and other forces to be employed in support of the plan.

FORCE PACKAGE

A grouping of combat, combat support, and combat service support forces linked together or uniquely identified so that they may be extracted from or adjusted as an entity in the TPFDD to enhance flexibility and usefulness of the operation plan during a crisis.

FORCE REQUIREMENT GENERATOR (FRG)

The JOPS III subsystem which supports the force and deployment planning steps in the Plan Development and Execution planning phases. The FRG provides the capability for a planner to add, delete, and modify force data and produce several different reports to aid analysis.

FORCE REQUIREMENT NUMBER

The alphanumeric code used to uniquely identify each force requirement in a given OPLAN TPFDD.

FORCE SHORTFALL

A deficiency in the number of type units available for planning within the time required for the performance of an assigned task.

FRAGMENTATION CODE (FRAG)

A one-character code which is the second part of the unit-line-number, used to uniquely identify units to satisfy a larger force requirement. In those cases where 2 to 35 units are required to satisfy a request for a single force requirement, FRAG will be unique for each unit and must not be zero. Codes 1-9 and A-Z are allowable. The zero must be used when only one unit satisfies the requested force requirement.

INSERT

A one-character code which is the third part of the unit-line-number used to uniquely identify units to satisfy a larger force requirement. It is used to provide unique identification for 2 to 35 units that replace a previously fragmented unit. Another function of the insert code is to provide the capability to include more than 35 units to satisfy any one force requirement. Using FRAG alone, the maximum number allowable is 35, given the values 1 through 9 and A through Z. Insert uses the same values, except for I and O. Using FRAG and INSERT combined, in excess of 1000 units may be associated with a single force requirement.

INTENSIVE MANAGEMENT

The continuous process by which the JDA, the supported and supporting commanders, the Services, TOAs, and appropriate Defense Agencies insure that movement data in the JDA deployment data base for the initial days (approximately the first 15 days) of deployment/mobilization are current to support immediate execution.

INTERMEDIATE LOCATION

An intermediate point in the deployment routing of forces used to delay the force for a specified time, normally longer than one day. It is often used to unite personnel and cargo portions of split shipments in the objective area. This point may occur between the origin and POE, the POE and POD, or the POD and destination; however, only one intermediate location may be used during the deployment of a unit.

JOINT DEPLOYMENT AGENCY

The JCS coordinating authority responsible for centralized coordination of mobilization deployment planning and the refinement of the TPFDD of the supported commander. Uses the TPFDD to create an operation plan data base that will support deployment.

JOINT DEPLOYMENT SYSTEM

The JDS is composed of a common set of planning and execution procedures, including a deployment data base. The JDS procedures and data base allow for the rapid exchange of information in order to request forces, identify units and cargo, and provide a movement monitoring capability. Within JDS, the deployment data base is the focus for intensively managing and maintaining previously developed deployment data in an executable status and for developing and manipulating crisis operation planning data during time-sensitive situations.

JOPSREP

Acronym for JOPS Reporting System - a formatted and coded means of providing standard data element descriptions, edit criteria, and procedures for reporting a time-phased force and deployment data (TPFDD) among commands and agencies involved in joint operational planning.

LATEST ARRIVAL DATE (LAD)

The date specified at the port of debarkation that indicates the latest date relative to D-Day that a unit, resupply or replacement personnel shipment can be closed. The LAD is used to time-phase shipments.

LIMITING FACTOR

A factor or condition that, either temporarily or permanently, impedes mission accomplishment. Illustrative examples are transportation network deficiencies, lack of in-place facilities, malpositioned forces or materiel, extreme climatic conditions, distance, transit/overflight rights, political conditions, etc.

MEDICAL EVACUEES

Personnel who are wounded, injured, or ill and must be moved to or between medical facilities.

MEDICAL PLANNING PROGRAM (MED or MPP)

The JOP III subsystem which allows the planner to utilize the basic plan data contained on the TPFDD to quantify the impact of a proposed operation on the medical system. The medical planner applies a combination of problem, force, or environment-related planning factors against planned force levels to determine the impact of personnel losses, hospital admissions, and the resultant levels of support such as bed, evacuation, and blood/fluid requirements.

EXHIBIT JOP005-1 (Cont)

MOBILITY ECHELON

A subordinate element of a type which is scheduled for deployment separately from the parent unit.

MODULE

A collection of one or more programs aligned to accomplish major functions of a subsystem. For example, JOPS Module F10 contains F10, F10A, F10B, and F10C which accomplish a specific function and contribute to the overall function of the module; OPLAN identification.

MOVEMENT REQUIREMENT GENERATOR (MRG)

The JOP III subsystem which provides an automated capability to produce resupply and replacement personnel requirements for an OPLAN. Th[illegible] logistical support requirements are added to the TPFDD/SRF in the required format to comp[illegible] the time-phased transportation requirements.

MOVEMENT SCHEDULE

A schedule developed to monitor or track [illegible]rate identity whether it is a force requirement, cargo increment number, or a lift asset. The schedule reflects the assignment of specific lift resources (such as an aircraft or ship) that will be used to move the personnel and cargo included in a specific movement increment. Arrival and departure times at POE, etc., are detailed to show a flow and workload at each location. Movement schedules are detailed enough to support plan implementation.

MOVEMENT TABLE

A table prepared by the TOAs for each force requirement and each nonunit-related personnel or cargo increment of the TPFDD concerning the scheduled movement from the origin or POE, intermediate location, and POD or destination. It is based on the estimated or planned availability of life resources and, hence, is not an execution document.

NON-ORGANIC TRANSPORTATION REQUIREMENTS

Unit personnel and cargo for which the transportation source must be an outside agency; normally, the transportation operating agencies (TOAs).

NONSTANDARD UNIT

A force requirement identified in an OPLAN for which movement characteristics have not been described in the TUCHA file. The planner is required to submit detailed movement characteristics for these units.

NONUNIT-RELATED CARGO

All equipment and supplies (other than those of a specific unit) which require transportation to an area of operations (e.g., resupply, military support for allies, or support for nonmilitary programs, such as civil relief, etc.).

NONUNIT-RELATED PERSONNEL

Personnel requiring transportation to an operations area other than those assigned to a specific unit (e.g., fillers, replacements, TDY/TAD, or civilians).

NOTIONAL PLANNING

The concept of planning with type forces instead of actual forces. For example, several infantry divisions are available. In notional planning, the requirement for an infantry division is simply shown as "an infantry division," thus postponing the assignment of an existing infantry division until Execution Planning. During Execution Planning, the most available infantry division could be designated as the unit to satisfy the requirement. When an actual unit is specified, it is called "Actual Planning," and a Unit Identification Code is assigned. For notional planning, the Unit Type Code (UTC) identifies the type unit requirement with no reference to a specific unit to fulfill that requirement. When notional units are used, notional origins and POEs must also be identified to permit reasonable transportation analysis.

OPERATION ORDER

An order prepared by the supported commander to implement the NCA decision for the execution of an operation.

OPERATION PLAN (OPLAN)

The term "operation plan" applies to any employment or deployment plan for the conduct of joint operations in a hostile environment (except for the SIOP) prepared by the commanders of the Unified and Specified commands in response to a requirement established by the Joint Chiefs of Staff. Operation plans are prepared in either complete format (OPLAN) or concept format (CONPLAN).

1. Operation Plan in a Complete Format (OPLAN). An operation plan for the conduct of joint operations that can be used as a basis for development of an OPORD. Complete plans include deployment/employment phases, as appropriate.

2. Operation Plan in Concept Format (CONPLAN). An operation plan in an abbreviated format that would require considerable expansion or alteration to convert it into an OPLAN or OPORD.

ORGANIC TRANSPORTATION

Transportation resources that are assigned to one's own unit and hence can provide the lift capability for all or part of the transportation requirements for that specific unit. Thus it may require little or no transportation support from the TOAs.

ORIGIN

The beginning point of a deployment which identifies the point or station at which the unit or nonunit-related equipment is located. For notional requirements, the origin will reflect the most likely station at which the requirement will become available.

PARENT

A force entry used to show the hierarchy or development strategy when force structuring. E.g., an Army Division (parent) actually will deploy as four separate company (subordinate) movements.

PERMANENT FILE (PRMFL)

This term is used to identify disk storage that remains part of the computer resources at all times. Permanent files are labeled and accessed by name. For example, JOP/JOPSPKGA/DATA/TUCHA would be a logical label for permanent disk space allocated for the TUCHA file. Files in JOPS III which are stored on permanent files include: APORTS, PORTS, GEOFILE, TUCHA, ASSETS, CHSTR, PDD, etc.

PLAN SUMMARY

A required element of an operation plan which provides a brief recap of the mission, the general situation, the concept of operations, the major forces required, command arrangements, and the commanders′ appraisal of logistic feasibility.

PORT OF EMBARKATION (POE)

The geographic point in the routing scheme where a transportation requirement will depart the continental United States. This point may or may not be collocated with the origin.

PORT OF DEBARKATION (POD)

The geographic point in the routing scheme where a transportation requirement will enter the objective area. For nonunit-related requirements, the POD will be the designated port of support for the specified type of shipment, and for planning purposes is the terminal point of the nonunit-related requirement. For unit requirements, the POD may be collocated with the destination.

PORT OF SUPPORT (POS)

A geographic location of a port in the objective area which may be used to distribute resupply or replacement personnel where they are needed. JOPS requires use of four types of ports of support: air (for personnel), sea (for general cargo), ammo, and POL.

PROGRAM

A collection of one or more routines aligned to accomplish sequential steps of a predetermined data processing objective.

EXHIBIT JOP005-1 (Cont)

PROJECTED CLOSURE DATE

The date at which an entire unit is expected to arrive and complete unloading at the destination. (This date is provided for those units to be moved by organic transportation.

PUNCH CARD

A storage media for information to be processed by the computer. Each card contains up to 80 columns of information. Each column can represent one letter or number (called a character). Normally punched cards have a rigid format, hence each column or group of columns must be used to represent one data element.

READY TO LOAD DATE (RLD)

A date specified for each unit in an OPLAN indicating when that unit will be ready to move from the origin.

RECORD

A collection of data elements that pertain to one logical subject. In JOPS, for example, all the data elements used to describe a force requirement and routing are stored in the force record. For resupply and replacement personnel, all the data elements are stored in nonunit-related cargo records and nonunit-related personnel records.

REPLACEMENTS

Personnel required to take the place of others who depart a unit.

RETROGRADE

The movement of personnel and/or cargo from the area of deployment back to their points of origin.

RETROGRADE PERSONNEL

Personnel evacuated from a theater of operations (may include noncombatants and civilians).

REQUIRED DELIVERY DATE (RDD)

A date established for each deployable unit that reflects the date relative to D-DAY that a unit must be delivered at the destination to properly support the deployment scheme as required by the concept of operations.

ROUTINE/SUBROUTINE

The smallest logical segment of computer code, usually designed to perform one function; i.e., compute SQRT, print a line, read a remote, etc.

SHORTFALL

The absence of forces, equipment, personnel, materiel, or capability--identified as a plan requirement--that would adversely affect the command's ability to accomplish its mission.

STANDARD-UNIT

A type unit whose UTC and movement characteristics are contained in the TUCHA file. The planner needs only to identify the UTC in proper format to have the movement characteristics inserted in the plan data.

EXHIBIT JOP005-1 (Cont)

SUBORDINATE COMMANDER A commander under the operational command of either a supported or supporting commander, normally a Service component commander or the commander of a subordinate Unified command or subordinate Joint Task Force.

SUBROUTINE See ROUTINE.

SUBSYSTEM A collection of one or more modules which are required to support the functional objectives of a system.

SUITABILITY Operation plan review criteria:

1. Is adaptable to the range of circumstances that could require implementation of the plan.

2. Makes effective use of the forces and resources available for planning.

3. Provides for the employment of combat and combat support operationals appropriate to the type and scale of threat envisaged.

4. Conforms to domestic law and international law (including the Law of War), which is binding upon the United States.

SUPPLY BUILDUP Supplies designed to provide a source of supply should, for some reason, the resupply pipeline to the objective area be interrupted.

SUPPORTED COMMANDER A commander having primary responsibility for all aspects of a task assigned in the JSCP or otherwise assigned; the commander who originates operation plans in response to requirements of the Joint Chiefs of Staff.

SUPPORTING COMMANDER A commander who provides augmentation forces or other support to a supported commander or develops a supporting plan. Includes the TOAs, as appropriate.

SUPPORTING FORCES Forces stationed in, or to be deployed to, an area of operations to provide support for the execution of an operation plan approved by the Joint Chiefs of Staff. Operational command of supporting forces is not passed to the supported commander.

SUPPORTING PLAN An operation plan prepared by either a supporting commander, a subordinate commander, or the JDA to satisfy the requests or requirements of the supported commander's plan.

EXHIBIT JOP005-1 (Cont)

SUSTAINING SUPPLY

That material required to support a unit after arrival in theater from the time accompanying supply and PWRs are anticipated to run out; until regular resupply commences.

TAPE STORAGE

A storage media for information to be processed by the computer. Similar to magnetic tape used for audio recording, computer tapes use arrangements of magnetic bits, activated by heads on a tape drive, to represent characters of information in records and files. In JOPS, tape storage is used for the TPFDD/SRF data base, the Planning Factors File (PFF), the TFE control file, etc. Before working with these files, they must be restored to disk.

TEMPORARY FILE (TEMP FILE)

This is a term used to describe disk space that is allocated to a user only for the time that the remote terminal is actively interfacing with the JOPS system. Files that use this method of storage in JOPS are files that are permanently stored on tape plus temporary work files used internally by several of the modules.

TIMES

(C-, D-, M-days end at 2400Z)
1. C-Day. The unnamed day on which a deployment operation commences or is to commence.

2. D-Day. The unnamed day on which a particular operation (i.e., land assault, air strike, naval bombardment, parachute assault, or amphibious assault) commences or is to commence.

3. H-Hour. The specific hour on D-Day at which a particular operation commences.

4. M-Day. The term used to designate the day on which mobilization is to begin.

TIME-PHASED FORCE DEPLOYMENT DATA (TPFDD)

The computer-supported data base portion of an operation plan; it contains time-phased force data, nonunit-related cargo and personnel data, and transportation data for the operation plan, including:

1. Units to be employed.

2. Units to be deployed to support the OPLAN, with a priority indicating the desired sequence of their arrival at ports on a given day.

3. Routing of forces to be deployed.

4. Mobility data associated with deploying forces.

EXHIBIT JOP005-1 (Cont)

5. Nonunit-related cargo and personnel movements to be conducted concurrently with the deployment of forces.

6. Estimate of transportation requirements that must be fulfilled by common-user lift resources as well as those requirements that can be fulfilled by assigned or attached transportation resources.

TPFDD REFINEMENT

A two-phased process that identifies specific forces, incorporates accurate movement requirements for the first 90 days of a TPFDD, and insures that the deployment transportation requirements for the TPFDD are within the capabilities defined in JCS guidance. The process is administered by JDA in coordination with the supported commander, supporting commanders, Services, TOAs, and other agencies. TPFDD refinement is a JDS procedure that complements JOPS procedures for selected operation plans and occurs during the JOPS plan development phase.

TIME-PHASED FORCE DEPLOYMENT LIST

A listing that identifies type units to support a particular operation plan and provides data concerning their routing from origin to destination.

TIME-PHASED TRANSPORTATION REQUIREMENTS LIST (TPTRL)

A computer listing retrieval from the TPFDD with the movement requirements, including a time-phased list of type units/mobility echelons, fillers and replacement personnel, and nonunit-related cargo to be transported by air or sea to support an OPLAN; provides mobility data related to these deployments and movement requirements to be fulfilled by both common-user lift resources and assigned or attached transportation resources.

TIME-SHARING SUBSYSTEM (TS1)

A system on the H6000 computer which allows several users to share the resources of one subsystem. JOPS III modules have been recently reprogrammed to operate under the Time-sharing Supervisor. Therefore, when using the TS1 version of JOPS III, several users can share the same programs, thus reducing the amount of core required for JOPS III supports.

TRANSPORTATION FEASIBILITY ESTIMATOR (TFE)

The JOPS III subsystem which compares transportation requirements to transportation assets and capabilities to determine whether or not OPLAN deployment requirements can be satisfied by available lift resources.

TYPE UNIT

A type of organizational entity established within the Armed Forces and uniquely identified by a UTC.

EXHIBIT JOP005-1 (Cont)

TYPE UNIT DATA FILE (TUCHA) — A file that provides standard planning data and movement characteristics for personnel, cargo, and accompanying supplies associated with type units.

UNIT DESIGNATION LIST — A list of actual units by UIC designated to fulfill requirements of a force list.

UNIT IDENTIFICATION CODE (UIC) — A six-character, alphanumeric code that uniquely identifies each Active, Reserve, and National Guard unit of the Armed Forces.

UNIT TYPE CODE (UTC) — The five-character, alphanumeric code that uniquely identifies each type unit.

UNIT-RELATED EQUIPMENT AND SUPPLIES — All equipment and supplies requiring transportation to an area of operations that are assigned to a specific unit or that are designated as accompanying supplies.

VISUAL INFORMATION PROCESSOR (VIP) — A remote terminal which employs a keyboard and a cathode-ray tube to provide communication between the computer and the user.

WARNING ORDER — A communication device used by commanders to advise subordinates of impending action. A warning order is generally a prelude to the ALERT ORDER, OPORD, or FRAGORD. The JCS may use the warning order as a planning directive to initiate Phase III of the Crisis Action System.

EXHIBIT JOP005-1 (Cont)

SECTION II -- ACRONYMS AND ABBREVIATIONS

The following is a list of selected acronyms that are frequently used in planning. Generally acronyms and abbreviations should be avoided in the writing of joint plans and orders. If a long title must be used repeatedly, the acronym or abbreviation may be employed provided the first time it is used the long title is spelled out fully along with its related acronym or abbreviation.

ACE	Allied Command Europe
ACFT	aircraft
ACP	Allied Communications Publication
AD	armored division
ADCOM	Aerospace Defense Command
ADM	atomic demolition munition
ADN	ACE DGZ number
ADNAC	Aerospace Defense of the North American Continent
ADP	automatic data processing
AFCE	Air Force civil engineering unit
AFRTS	American Forces Radio and Television Service
AID	Agency for International Development
AIF	automated installation intelligence file
ALD	accounting line designator
APOD	aerial port of debarkation
APOE	aerial port of embarkation
ARRS	Aerospace Rescue and Recovery Service
ASAP	as soon as possible
ASCC	Air Standardization Coordinating Committee
ASD(PA)	Assistant Secretary of Defense (Public Affairs)
ASGRO	Armed Services Graves Registration Office
ASMRO	Armed Services Medical Regulating Office
ASW	antisubmarine warfare
AUTODIN	Automatic Digital Network
AUTOSEVOCOM	Automatic Secure Voice Communications
AUTOVON	Automatic Voice Network
AVGAS	aviation gasoline
AWSR	Air Weather Service Regulation

BBLS	barrels
BD	base development
BE	Basic Encyclopedia
BPS	Basic PSYOP Study

C3	command, control, and communications
C3CM	command, control, and communications countermeasures
CAO SOP	Standing Operating Procedures for Coordination of Atomic Operations
CAS	Crisis Action System
CCTC	Command and Control Technical Center
CED	communications-electronics directive
CEOI	communications-electronics operating instruction
CEP	circular error probable

EXHIBIT JOP005-2

CESP	Civil Engineering Support Plan
CESPG	Civil Engineering Support Plan Generator
CIA	Central Intelligence Agency
CIN	cargo increment number
CINCAD	Commander in Chief, Aerospace Defense Command
CINCLANT	Commander in Chief, Atlantic
CINCMAC	Commander in Chief, Military Airlift Command
CINCNORAD	Commander in Chief, North American Air Defense Command
CINUNC/CFC	Commander in Chief, United Nations Command/Combined Forces Command
CNTY	country
COMINT	communications intelligence
COMJTF	Commander, Joint Task Force
COMJUWTF	Commander, Joint Unconventional Warfare Task Force
COMRDJTF	Commander, Rapid Deployment Joint Task Force
COMSC	Commander, Military Sealift Command
COMSEC	communication security
COMUSJTF	Commander, US Joint Task Force
CONPLAN	operation plan in concept format
CONUS	continental United States
CPFL	contingency planning facilities list
CRA	command relationships agreement
CRAF	Civil Reserve Air Fleet
CRITIC	critical intelligence
CRITICOMM	Critical Intelligence Communications System
CSG	cryptologic support group
CSS	Central Security Service
CUSRPG	Canada-United States Regional Planning Group
CW	chemical warfare
DAME	defense against methods of entry
DASE	defense against sound equipment
DAT	deployment action team
DCA	Defense Communications Agency
DCID	Director of Central Intelligence Directive
DCS	Defense Communications System
DE	damage expectancy
DEFCON	Defense Readiness Condition
DEPDA	Deployment Data
DEPLAN	Deployment Plan
DFE	division force equivalent
DGZ	desired ground zero
DIA	Defense Intelligence Agency
DIAM	DIA Manual
DIAR	DIA Regulation
DIP	Defense Intelligence Plan
DIPP	Defense Intelligence Projections for Planning
DIRNSA/CHCSS	Director, National Security Agency/Chief, Central Security Service
DLA	Defense Logistics Agency
DLAR	Defense Logistics Agency Regulation
DMA	Defense Mapping Agency
DNA	Defense Nuclear Agency
DOD	Department of Defense

DSAR	Defense Supply Agency Regulation
DSSCS	Defense Special Security Communications System
DSTP	Director of Strategic Target Planning
DTF	direct to forces
DTG	date-time group
EAD	earliest arrival date
ECAC	Electromagnetic Compatibility Analysis Center
ECCM	electronic counter-countermeasures
ECM	electronic countermeasures
EDP	emergency defense plan
E&E	evasion and escape
EEFI	essential elements of friendly information
EEFIS	Evasion and Escape Fingerprint Identification System
EEI	essential elements of information
ELINT	electronic intelligence
EMCON	emission control
EPW	enemy prisoner of war
EPW/CI/Det	enemy prisoners of war/civilian internees/detainees
ERT	execution reference time
ESC	Electronic Security Command
ESM	electronic warfare support measures
EW	electronic warfare
FAA	Federal Aviation Administration
FATs	fatalities
FIDP	foreign internal defense plan
FMA	Foreign Media Analysis
FMFM	Fleet Marine Force Manual
FRD	Formerly Restricted Data
FRG	Force Requirement Generator
FRN	force requirement number
FYDP	Five-Year Defense Program
GDP	general defense plan
GEOFILE	Standard Specified Geolocation File
GWOA	guerrilla warfare operational area
HALO	high altitude-low opening
HF	high frequency
HOB	height of burst
HUMINT	human resources intelligence
IADB	Inter-American Defense Board
ICAO	International Civil Aeronautics Organization
ICOD	intelligence cutoff date
ICP	inventory control point
ICR	intelligence collection requirement
ICRC	International Committee of the Red Cross
IFF	identification, friend or foe
IFFN	identification, friend, foe, or neutral
IMINT	imagery intelligence
IPSP	Intelligence Priorities for Strategic Planning

JANAP	Joint Army-Navy-Air Force Publication
JCC	Joint Coordination Center
JCGRO	Joint Central Graves Registration Office
JDA	Joint Deployment Agency
JDS	Joint Deployment System
JIB	Joint Information Bureau
JIEP	Joint Intelligence Estimate for Planning
JMRC	Joint Mobile Relay Center
JMRO	Joint Medical Regulating Office
JMTSS	Joint Multichannel Trunking and Switching System
JOPS	Joint Operation Planning System
JOPS I	Volume I (Deliberate Planning Procedures)
JOPS II	Volume II (Supplementary Planning Guidance)
JOPS III	Volume III (ADP Support)
JOPS IV	Volume IV (Crisis Action System)
JPAM	Joint Program Assessment Memorandum
JPAO	Joint Public Affairs Office
JRC	Joint Reconnaissance Center
JRS	Joint Reporting Structure
JSCP	Joint Strategic Capabilities Plan
JSPD	Joint Strategic Planning Document
JSPDSA	Joint Strategic Planning Document Supporting Analysis
JSPS	Joint Strategic Planning System
JSTPS	Joint Strategic Target Planning Staff
JTB	Joint Transportation Board
JTD	joint table of distribution
JTF	joint task force
JTSS	JOPS Time-sharing Software System
JUWTF	Joint Unconventional Warfare Task Force
kt	kiloton
LAD	latest arrival date
LERTCON	alert condition
LFM	Landing Force Manual
LNO	limited nuclear option
LOC	line of communication
LOC ACC	location accuracy
LOGDA	Logistics Geolocation Data File
MAAG	military assistance advisory group
MAB	Marine amphibious brigade
MAC	Military Airlift Command
MACR	Military Airlift Command Regulation
MAF	Marine amphibious force
MAGTF	Marine air-ground task force
MAO	major attack option
MAP	Military Assistance Program
MASINT	measurement and signature intelligence
MAU	Marine amphibious unit
MBPO	Military Blood Program Office
MC	NATO Military Committee Document
MCC	Military Cooperation Committee
MCD	manipulative communication deception

EXHIBIT JOP005-2 (Cont)

MC&G	mapping, charting, and geodesy
MCO	Marine Corps Order
METCON	meteorological control
MHE	materials handling equipment
MIJI	meaconing, interference, jamming, and intrusion
MILCON	military construction
MILGP	military group
MMPNC	medical materiel program for nuclear casualties
MOP	memorandum of policy
MOS	military occupational specialty
MRF	mobile relay facility
MRG	Movement Requirements Generator
MSC	Military Sealift Command
MSR	main supply routes
M/T	measurement tons
MTMC	Military Traffic Management Command
MU	marry-up
NATO	North Atlantic Treaty Organization
NAVMAT	Navy Material Command
NAVSECGRUCOM	Naval Security Group Command
NBC	nuclear, biological, and chemical
NCA	National Command Authorities
NCEUR	NSA/CSS Europe
NCPS	Nuclear Contingency Planning System
NCREP	National Security Agency/Central Security Service Representative
NDBS	National Data Buoy System
NDL	National DGZ list
NEO	noncombatant evacuation operation
NFIB	National Foreign Intelligence Board
NFIBONLY	NFIB Departments Only
NIPS	National Military Command System Information Processing System
NISP	NUWEP Intelligence Support Plan
NMCS	National Military Command System
NOAA	National Oceanic and Atmospheric Administration
NOCONTRACT	Not Releasable to Contractors or Contractor/Consultants
NOFORN	Not Releasable to Foreign Nationals
NOMS	Nuclear Operations Monitoring System
NOMSS	National Operational Meteorological Satellite System
NOP	nuclear operations
NOPLAN	no plan
NORAD	North American Air Defense Command
NRL	NUWEP reconnaissance list
NSA/CSS	National Security Agency/Central Security Service
NSC	National Security Council
NSCID	National Security Council Intelligence Directive
NSDAB	non-self-deployable aircraft and boats
NSN	national stock number
NSRL	National SIGINT Requirements List
NSTL	National Strategic Target List
NSWTG	Naval Special Warfare Task Group
NSWTU	Naval Special Warfare Task Unit

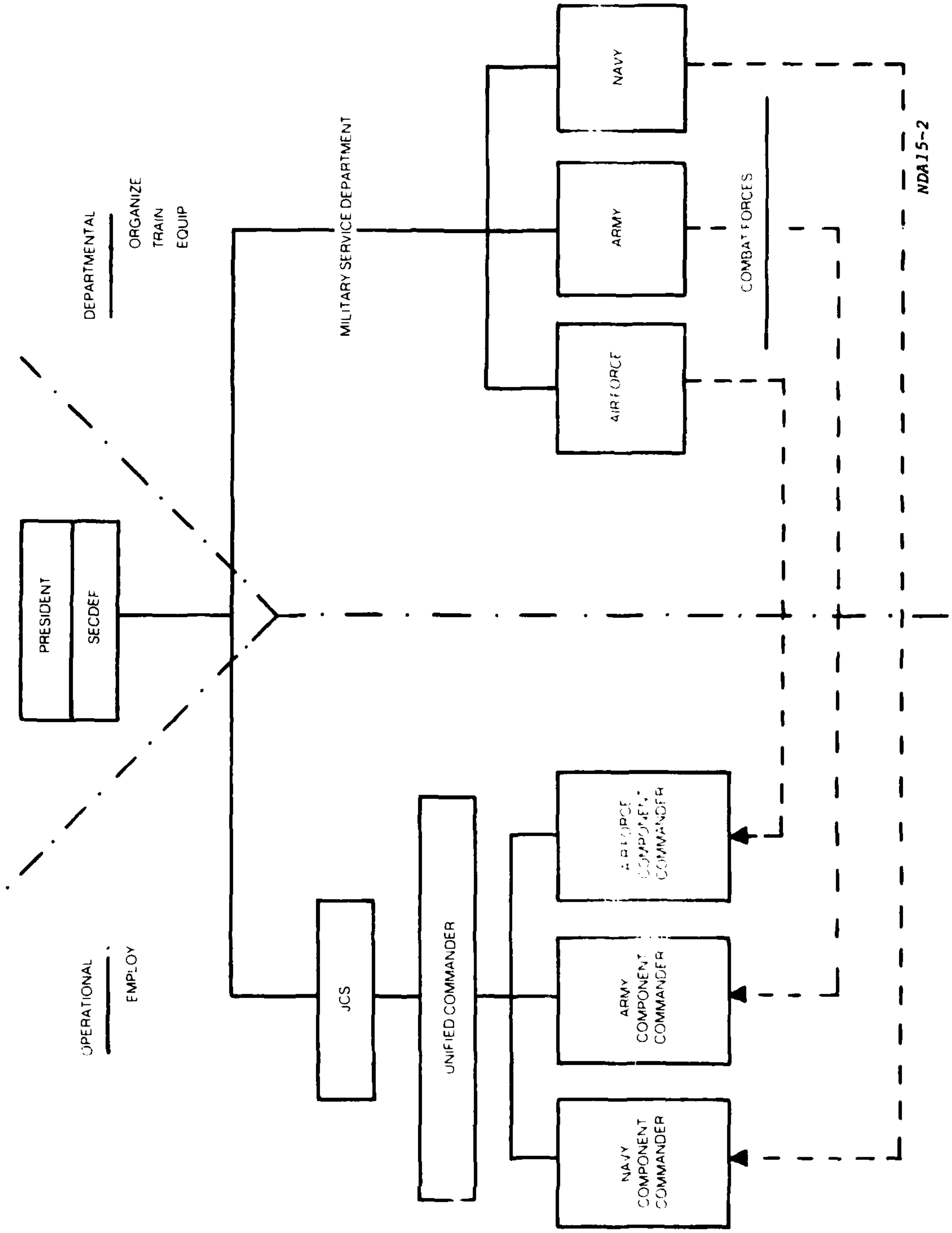
EXHIBIT JOP005-2 (Cont)

NUWEP	Policy Guidance for the Employment of Nuclear Weapons
NWP	Naval Warfare Publication
NWS	National Weather Service
OASD(PA)	Office of the Assistant Secretary of Defense (Public Affairs)
ODC	Office of Defense Cooperation
OER	operational ELINT requirements
OIC	officer in charge
OJCS	Organization of the Joint Chiefs of Staff
OMC	Office of Military Cooperation
OPCOM	operational command
OPCON	operation control
OPLAN	operational plan in complete format
OPORD	operation order
OPPA	operation plan package appraisal
OPR	office of primary responsibility
OPREP	commanders operational report
OPSEC	operations security
ORCON	Dissemination and Extraction of Information Ccutrolled by Originator
OSD	Office of the Secretary of Defense
PA	public affairs probability of arrival
PAMIS	Psychological Operations Automated Management Information System
PAR	population at risk
PD	probability of damage
PEAD	Presidential Emergency Action Document
PEAS	PSYOP Effects Analysis Subsystem
PIN	plan identification number
PNIO	Priority National Intelligence Objectives
POD	port of debarkation
POE	port of embarkation
POL	petroleum, oils, and lubricants
POM	program objective memorandum
POMCUS	prepositioning of materiel configured to unit sets
PRI	priority
PROPIN	Caution--Proprietary Information Involved
PSA	psychological operations support activity
PSB	poststrike base
PSYOP	psychological operations
PV	physical vulnerability
PW	prisoner of war
PWRMR	prepositioned war reserve materiel requirement
PWRMS	prepositioned war reserve materiel stock
R&D	research and development
RCA	riot control agent
RD	Restricted Data
RDD	required delivery date
RDF	rapid deployment force
RDJTF	Rapid Deployment Joint Task Force

RFI	radio frequency interference
RLT	regimental landing team
RNO	regional nuclear option
ROE	rules of engagement
RO/RO	roll-on/roll-off
RQMT	requirement
SACEUR	Supreme Allied Commander Europe
SACR	Strategic Air Command Regulation
SADM	special atomic demolition munition
SAFE	selected area for evasion
SAO	selected attack option
SAR	search and rescue
SAS	special ammunition storage
SCE	Service Cryptologic Element
SEAL	Sea/Air/Land
SEL REP	Selected Release Procedures
SERE	survival, evasion, resistance, and escape
SI	special intelligence
SIF	special interdiction force
SIGINT	signals intelligence
SIGSEC	signals security
SIOP	Single Integrated Operational Plan
SNL	standard nomenclature list
SOFADS	Special Operations Foreign Area Data Subsystem
SOP	standing operating procedure
SOTA	SIGINT operational tasking authority
SPS	Special PSYOP Study
SRF	Secure Reserve Force
SRP	SIOP Reconnaissance Plan
SSBN	fleet ballistic missile submarine (nuclear propulsion)
SSO	Special Security Office/Officer
SSP	SIGINT Support Plan
SSR	SACEUR Strategic Reserve
S/T	short tons
STANAG	Standardization Agreement
TAC	Tactical Air Command
TACM	Tactical Air Command Manual
TAD	temporary additional duty
TDD	target DGZ designator
TDI	Target Data Inventory
TDY	temporary duty
TFE	Transportation Feasibility Estimator
TGT	target
TI	target island
TI/DGZ	target island/desired ground zero
TO	table or organization
TOA	transportation operating agency
TOE	table of organization and equipment
TOP	time-sensitive operation planning procedures to support the CAS
TOT	time over target
TPFDD	Time-Phased Force Deployment Data

TPFDL	Time-Phased Force Deployment List
TPTRL	Time-Phased Transportation Requirements List
TUCHA	Type Unit Data File
UCP	Unified Command Plan
UDL	Unit Designation List
UDT	underwater demolition team
UIC	unit identification code
ULC	unit level code
ULN	unit line number
UNAAF	Unified Action Armed Forces
UNITREP	Unit Status and Identity Report
USAF	US Air Force
USAINSCOM	US Army Intelligence and Security Command
USARRED	US Army Forces, US Readiness Command
USBRO	US Base Requirements Overseas
USCINCRED	Commander in Chief, US Readiness Command
USEUCOM	US European Command
USG	US Government
USICA	US International Communication Agency
USSS	US SIGINT System
UTC	unit type code
UW	unconventional warfare
VNTK	vulnerability number for a degree of hardness
WIN	WWMCCS Intercomputer Network
WISP	Wartime Information Security Program
WNINTEL	WARNING NOTICE--Intelligence Sources and Methods Involved
WRM	war reserve materiel
WWMCCS	Worldwide Military Command and Control System
YLD	yield

EXHIBIT JOP005-2 (Cont)



TYPICAL UNIFIED COMMAND

Cover
Letter of Transmittal
Security Instructions
Record of Changes
Plan Summary
Table of Contents
Basic Plan

1. Situation
2. Mission
3. Execution (includes concept of operation)
4. Administration and Logistics
5. Command and Signal

Annexes

A. Task Organization (includes time-phased force and deployment list)

B. Intelligence

C. Operations

D. Logistics (includes medical, mobility/transportation, and civil engineering support)

E. Personnel

F. Public Affairs

G. Civil Affairs

H. Environmental Services

I. Command Relationships

J. Communications-Electronics

K. Operations Security

L. Mapping, Charting, and Geodesy

M. Execution Checklist

N. Distribution

Operation Plan in Complete Format

EXHIBIT JOP010-2

APPENDIX A

OPLAN TABLE OF CONTENTS
CLASSIFICATION

HEADQUARTERS, US EUROPEAN COMMAND
APO NEW YORK 92128
1 August 1978

USCINCEUR OPLAN 4999-79 (U)
TABLE OF CONTENTS (U)

DATE FOR ( ) DECLASSIFICATION
OR ( ) REVIEW
IS:
EXTENDED BY:
REASON:

v
CLASSIFICATION
V-9

Format,
Table of Contents
Chapter V

EXHIBIT JOP010-3

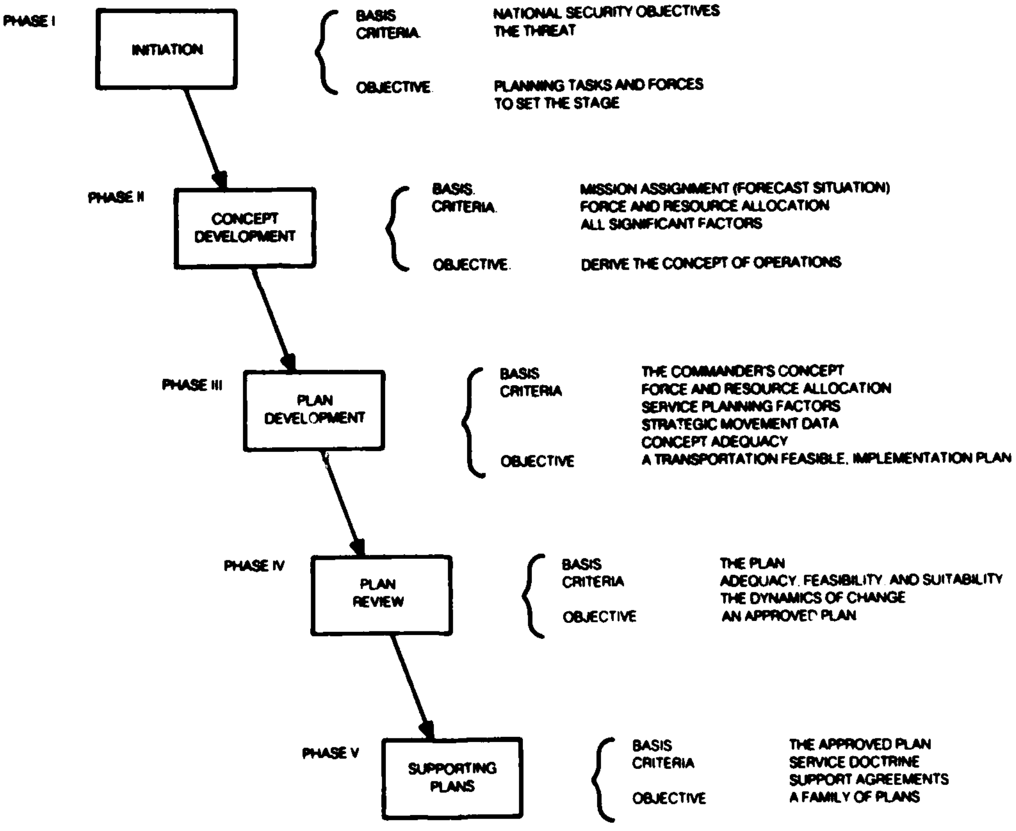
CONTENTS **PAGE NO.**

Format,
Table of Contents
Chapter V

EXHIBIT JOP010-3 (Cont)

Note: The letters "I" and "O" will not be used as annex designations.

EXHIBIT JOP010-3 (Cont)



NDA15-1

EXHIBIT JOP015-1

SECTION I--DEFINITIONS OF COMMONLY USED TERMS

The following is a list of selected terms used in the Joint Planning Community along with a practical definition of each. Exhibits JOP510-1 through JOP600-7 are for use with the old JOPS course. Section II, a table of acronyms, follows this list of definitions.)

ACTUAL PLANNING

Actual planning involves the concept of identifying actual units to fulfill plan requirements at the time the force list is initially established. This identification is accomplished by assigning Unit Identification Codes (UIC) to each requirement listed in the force list for an OPLAN.

ALERT ORDER

A formal directive issued by the JCS. It reflects an NCA decision that US military forces may be required, provides essential guidance for planning in the prevailing situation, and marks the onset of execution planning.

ANNEXES

Documents appended to the basic plan or order to make it clearer or to give further details.

ASSIGNED FORCES

Forces in being that have been placed under the operational command or control of a commander.

ATTACHMENT TO AN OPERATION PLAN OR ORDER

An attachment to an operational plan or order is a separately identifiable amplification of the basic plan or operation order. Attachments are annexes, appendices, tabs, and enclosures.

AUGMENTATION FORCES

Forces to be transferred to the operational command of a supported commander during the execution of an operation plan approved by the Joint Chiefs of Staff.

AVAILABLE TO LOAD DATE (ALD)

The data specified for each unit in an operation plan which indicates when that unit will be available for loading at the port of embarkation (POE).

BASIC PLAN

That part of an operation plan which forms the base structure for annexes and appendices. It consists of general statements related to the situation, mission, execution, logistics, administration, and command and signal.

C-DAY

The unnamed day on which deployment operations commence.

CHSTR

Characteristics of transportation resources file. A JOPS reference file that contains information concerning the planning factors and characteristics of transport aircraft, ships, and landing craft

EXHIBIT JOP510-1

required in the development and evaluation of joint operation plans.

CIVIL ENGINEERING SUPPORT PROGRAM (CESPG)

CESPG is a JOPS III subsystem. The purpose of the CESPG is to compute facility, construction force, and gross material requirements to support joint operation plans.

CIVIL ENGINEERING SUPPORT PLANNING

That part of operation plan development which concerns the improvement or expansion of resources and facilities in the area of operations.

CONPLAN

An operation plan in a abbreviated concept format which would require expansion into an OPLAN or OPORD prior to implementation.

CONCEPT OF OPERATIONS

A verbal or written statement, in broad outline, of a commander's assumptions or intent in regard to an operation or series of operations. The purpose of the concept is to give an overall view of the operation.

CRISIS ACTION SYSTEM (CAS)

CAS provides guidance and procedures for the conduct of joint planning for the use of time-sensitive situations. The procedures provide the Joint Chiefs of Staff with information to develop timely recommendations to the National Command Authorities (NCA) for decisions involving the use of US military forces.

DATA BASE

A data base is a collection of one or more files which represent all the data associated with or supporting the objective of an ADP system. For example, in JOPS, one data base supports each OPLAN. It is called the TPFDD/SRF, containing the TPFDD and the Summary Reference File.

DATA ELEMENT

The smallest intelligible piece of information relating to an overall topic. For example, in a personnel system, NAME and SSAN could be data elements. Sometimes the contents of the data elements may be a code and not the complete data itself. For example, JOPS uses the code "A" as a one-character data element to represent the transportation mode of "AIR." To identify the Military Airlift Command (MAC) as the source of transportation, another character data element is used in which the code "K" is stored.

D-DAY

The unnamed day on which an operation plan is executed.

DEPDA

The data base in the deployment reporting (DEPREP) system which contains the time-phased force, resupply, replacement personnel, and transportation data for an OPLAN. It includes:

(1) Type of units to be employed;

(2) Type unit to be deployed for the execution or support of the OPLAN and their priority for deployment;

(3) Routing of forces to be deployed;

(4) Mobility data associated with deployment;

(5) Personnel and logistic movement to be conducted concurrently with the deployment of forces; and

(6) Estimate of transportation requirements, including those to be fulfilled by assigned transport units.

DEPLOYABILITY POSTURE

The state or stage of a unit's preparedness for deployment to participate in a military operation.

DEPLOYMENT PLANNING

That part of operation planning which concerns the relocation of forces to the desired area of operation.

DEPREP

Acronym for deployment report--a formatted and coded means of conveying deployment planning and redeployment planning data among commands and agencies.

DESTINATION

The terminal geographic location in the routing scheme for forces only. Nonunit-related transportation requirements will not use the destination as part of its routing. The destination identifies the station or location in the objective area at which the unit will be deployed.

DIRECT ACCESS COMMUNICATION (DAC)

A system on the H6000 computer which allows a user to communicate with a computer program through a remote terminal. The standard JOP III software has been written to use this method of access. Each user must initiate the JOPS programs and then connect to them using the DAC system. Under DAC control, each user on the JOP III subsystem requires 38-40K of memory. This severely restricts the number of users that can efficiently use the resources simultaneously.

DISK STORAGE

A method of storing information on a device patterned after phonograph records. They are stacked, however, to provide more surface for storage. Each disk plane is recorded by tracks which are partitioned into sectors. Each sector can be magnetized to represent characters of information. Disk storage has much faster access and transfer rate than tape storage. Moreover, the capability to access data in a random manner permits direct retrieval of data rather than a sequential search of each record until the desired one is located. In JOPS, all data files and data bases are on disk storage devices during processing. However, tapes are used for permanent storage of OPLAN unique files between processing sessions. Thus the advantages of both can be attained; tapes provide a cheaper means of permanently storing information, and during processing this data is restored to disk thus giving the faster processing capability.

EARLIEST ARRIVAL DATE (EAD)

The date specified at the port of debarkation that establishes the earliest date relative to D-DAY that a unit, resupply requirement, or replacement personnel requirement can be accepted during a deployment. Used with the latest arrival date (LAD) to define a time frame window for transportation purposes.

ELEMENTS OF AN OPLAN/OPORD

An element of an OPLAN or OPORD is an item which is listed in the table of contents including attachments.

EMPLOYMENT PLANNING

That part of operation planning which concerns the strategic or tactical use of forces and material within the area of operations.

ESTIMATED DEPARTURE DATE (EDD)

A date computed by the TFE simulation module for each transportation requirement which indicates the date of departure from the POE used during the simulation. This date will reflect the earliest date, after the available to load (ALD) date at the POE, which the simulation module could load and depart with the complete requirement.

EXECUTION PLANNING

That phase of operations planning in which a plan or concept is translated into an OPORD. It includes adapting the plan or concept to the prevailing circumstances, the designation of units to satisfy force requirements, the establishment of appropriate deployability posture, the scheduling of necessary transportation resources, and the dissemination of movement tables to regulate the deployment of forces requiring common user transportation.

FEASIBLE ARRIVAL DATE (FAD)

A date computed by the TFE simulation module which indicates for each transportation requirement the earliest date after the designated EAD that the shipment could be closed. The FAD is used to determine transportation shortfalls. When the FAD is later than the LAD, a transportation shortfall exists.

FILE

A file is a collection of one or more types of records which are related for some overall topic. In JOPS, for example, the TPFDD file is made up of force records, nonunit-related cargo records, and nonunit-related personnel records. Together this information represents the overall time-phased transportation required.

FORCE CARD

DEPREP Card Type A. The force card contains identification and description data such as unit type code (UTC), unit level code (ULC), estimated personnel strength, and other supplementary data to assist in identifying the force required in the operation plan. Combined with the information from the force routing card, this data may comprise the time-phased force deployment list.

FORCE LIST

A list of forces expressed as type units consisting of the total forces required by an operation plan with assigned forces, augmentation forces, and other forces to be employed in support of the plan.

FORCE REQUIREMENT GENERATOR (FRG)

The JOPS III subsystem which supports the force and deployment planning steps in the Plan Development and Execution planning phases. The FRG provides the capability for a planner to add, delete, and modify force data and produce several different reports to aid analysis.

FORCE REQUIREMENT NUMBER

The alphanumeric code used to uniquely identify each force requirement in a given OPLAN TPFDD.

FORCE ROUTING CARD

DEPREP Card Type B. The force routing card provides information concerning intermediate location, port of debarkation, and destination of a force requirement. Associated with each of these areas are a preferred mode and source of transportation, load configuration, discharge constraints, and pertinent dates.

FORCE SHORTFALL

A deficiency in the number of type units available for planning within the time required for the performance of an assigned task.

EXHIBIT JOP510-1 (Cont)

FRAGMENTATION CODE (FRAG)

A one-character code which is the second part of the unit-line-number, used to uniquely identify units to satisfy a larger force requirement. In those cases where 2 to 35 units are required to satisfy a request for a single force requirement, FRAG will be unique for each unit and must not be zero. Codes 1-9 and A-Z are allowable. The zero must be used when only one unit satisfies the requested force requirement.

INSERT

A one-character code which is the third part of the unit-line-number used to uniquely identify units to satisfy a larger force requirement. It is used to provide unique identification for 2 to 35 units that replace a previously fragmented unit. Another function of the insert code is to provide the capability to include more than 35 units to satisfy any one force requirement. Using FRAG alone, the maximum number allowable is 35, given the values 1 through 9 and A through Z. Insert uses the same values, except for I and O. Using FRAG and INSERT combined, in excess of 1000 units may be associated with a single force requirement.

INTERMEDIATE LOCATION

An intermediate point in the deployment routing of forces used to delay the force for a specified time, normally longer than one day. It is often used to unite personnel and cargo portions of split shipments in the objective area. This point may occur between the origin and POE, the POE and POD, or the POD and destination; however, only one intermediate location may be used during the deployment of a unit.

LATEST ARRIVAL DATE (LAD)

The date specified at the port of debarkation that indicates the latest date relative to D-Day that a unit, resupply or replacement personnel shipment can be closed. The LAD is used to time-phase shipments.

LIMITING FACTOR

A deficiency in resources required to support an operation plan, such as the movement capabilities, personnel, logistics, or facilities.

MEDICAL PLANNING PROGRAM (MED or MPP)

The JOP III subsystem which allows the planner to utilize the basic plan data contained on the TPFDD to quantify the impact of a proposed operation on the medical system. The medical planner applies a combination of problem, force, or environment-related planning factors against planned force levels to determine the impact of personnel losses, hospital admissions, and the resultant levels of support such as bed, evacuation, and blood/fluid requirements.

EXHIBIT JOP510-1 (Cont)

MOBILITY ECHELON A subordinate element of a type which is scheduled for deployment separately from the parent unit.

MODULE A collection of one or more programs aligned to accomplish major functions of a subsystem. For example, JOPS Module F10 contains F10, F10A, F10B, and F10C which accomplish a specific function and contribute to the overall function of the module; OPLAN identification.

MOVEMENT REQUIREMENT GENERATOR (MRG) The JOP III subsystem which provides an automated capability to produce resupply and replacement personnel requirements for an OPLAN. These logistical support requirements are added to the TPFDD/SRF in the required format to complete the time-phased transportation requirements.

NON-ORGANIC TRANSPORTATION REQUIREMENTS Unit personnel and cargo for which the transportation source must be an outside agency; normally, the transportation operating agencies (TOAs).

NONSTANDARD UNIT A force requirement identified in an OPLAN for which movement characteristics have not been described in the TUCHA file. The planner is required to submit detailed movement characteristics for these units.

NONUNIT-RELATED CARGO All equipment and supplies (other than those of a specific unit) which require transportation to an area of operations (e.g., resupply, military support for allies, or support for nonmilitary programs, such as civil relief, etc.).

NONUNIT-RELATED PERSONNEL Personnel requiring transportation to an operations area other than those assigned to a specific unit (e.g., fillers, replacements, TDY/TAD, or civilians).

NOTIONAL PLANNING The concept of planning with type forces instead of actual forces. For example, several infantry divisions are available. When using notional planning, the requirement for an infantry division is simply shown as "An Infantry Division," thus postponing the assignment of an existing force until Execution Planning. During Execution planning, the most available unit could be designated to satisfy the requirement. When an actual unit is specified, it is called "Actual Planning," and a Unit Identification Code is assigned. For notional planning, the Unit Type Code (UTC) identifies the type unit requirement with no reference to a specific unit to fulfill that requirement.

OPERATION ORDER
A directive, usually formal, issued by a commander to subordinate commanders for the purpose of effecting coordinated execution of an operation.

OPLAN
An operation plan in complete format, including all annexes (i.e., an operation plan for the conduct of military operations which can be translated into an operation order with minimum alteration).

ORGANIC TRANSPORTATION
Transportation resources that are assigned to one's own unit and hence can provide the lift capability for all or part of the transportation requirements for that specific unit. Thus it may require little or no transportation support from the TOAs.

ORIGIN
The beginning point of a deployment which identifies the point or station at which the unit or nonunit-related equipment is located. For notional requirements, the origin will reflect the most likely station at which the requirement will become available.

PERMANENT FILE (PRMFL)
This term is used to identify disk storage that remains part of the computer resources at all times. Permanent files are labeled and accessed by name. For example, JOP/JOPSPKGA/DATA/TUCHA would be a logical label for permanent disk space allocated for the TUCHA file. Files in JOPS III which are stored on permanent files include: APORTS, PORTS, GEOFILE, TUCHA, ASSETS, CHSTR, PDD, FPF, MEF, and sometimes DEPDA.

PLAN SUMMARY
A required element of an operation plan which provides a brief recap of the mission, the general situation, the concept of operations, the major forces required, command arrangements, and the commanders' appraisal of logistic feasibility.

PORT OF EMBARKATION (POE)
The geographic point in the routing scheme where a transportation requirement will depart the continental United States. This point may or may not be collocated with the origin.

PORT OF DEBARKATION (POD)
The geographic point in the routing scheme where a transportation requirement will enter the objective area. For nonunit-related requirements, the POD will be the designated port of support for the specified type of shipment, and for planning purposes is the terminal point of the nonunit-related requirement. For unit requirements, the POD may be collocated with the destination.

PORT OF SUPPORT (POS)
A geographic location of a port in the objective area which may be used to distribute resupply or

replacement personnel where they are needed. JOPS requires use of four types of ports of support: air (for personnel), sea (for general cargo), ammo, and POL.

PROGRAM

A collection of one or more routines aligned to accomplish sequential steps of a predetermined data processing objective.

PROJECTED CLOSURE DATE

The date at which an entire unit is expected to arrive and complete unloading at the destination. (This date is provided for those units to be moved by organic transportation.)

PUNCH CARD

A storage media for information to be processed by the computer. Each card contains up to 80 columns of information. Each column can represent one letter or number (called a character). Normally punched cards have a rigid format, hence each column or group of columns must be used to represent one data element.

READY TO LOAD DATE (RLD)

A date specified for each unit in an OPLAN indicating when that unit will be ready to move from the origin.

RECORD

A collection of data elements that pertain to one logical subject. In JOPS, for example, all the data elements used to describe a force requirement and routing are stored in the force record. For resupply and replacement personnel, all the data elements are stored in nonunit-related cargo records and nonunit-related personnel records.

RETROGRADE

The movement of personnel and/or cargo from the area of deployment back to their points of origin.

REQUIRED DELIVERY DATE (RDD)

A date established for each deployable unit that reflects the date relative to D-DAY that a unit must be delivered at the destination to properly support the deployment scheme as required by the concept of operations.

ROUTINE/SUBROUTINE

The smallest logical segment of computer code, usually designed to perform one function; i.e., compute SQRT, print a line, read a remote, etc.

SHORTFALL

Transportation or other resource requirements which cannot be met due to some circumstance that exhausted the resource.

STANDARD-UNIT

A type unit whose UTC and movement characteristics are contained in the TUCHA file. The planner needs only to identify the UTC in proper format to have the movement characteristics inserted in the plan data.

SUBORDINATE COMMANDER
A commander who is subordinate to either a supported or supporting commander, normally a service component commander or the commander of a subordinate unified command. This is the level at which detailed force lists and support requirements are produced.

SUBROUTINE
See ROUTINE.

SUBSYSTEM
A collection of one or more modules which are required to support the functional objectives of a system.

SUPPORTED COMMANDER
The commander having primary responsibility for all aspects of a task assigned in the JSCP or otherwise assigned; the commander who originates operation plans in response to requirements of the Joint Chiefs of Staff.

SUPPORTING COMMANDER
A commander who provides augmentation forces or other support to a supported commander.

SUPPORTING FORCES
Forces stationed in, or to be deployed to, an area of operations to provide support for the execution of an operation plan approved by the Joint Chiefs of Staff. Operational command of supporting forces is not passed to the supported commander.

SUPPORTING PLAN
An operation plan prepared by either a supporting commander or a subordinate commander to satisfy the requirements of the supported commander's plan.

SUSTAINING SUPPLY
That material required to support a unit after arrival in theater from the time accompanying supply and PWRs are anticipated to run out; until regular resupply commences.

TAPE STORAGE
A storage medium for information to be processed by the computer. Similar to magnetic tape used for audio recording, computer tapes use arrangements of magnetic bits, activated by heads on a tape drive, to represent characters of information in records and files. In JOPS, tape storage is used for the TPFDD/SRF data base, the logistical data (LOGDA), the Planning Factors File (PFF), the TFE control file, and normally the DEPDA file. Before working with these files, they must be restored to disk.

TEMPORARY FILE (TEMP FILE)
This is a term used to describe disk space that is allocated to a user only for the time that the remote terminal is actively interfacing with the

JOPS system. Files that use this method of storage in JOPS are files that are permanently stored on tape plus temporary work files used internally by several of the modules.

TIME-PHASED FORCE DEPLOYMENT DATA (TPFDD)

The main file of the plan unique data base for an OPLAN which includes time-phased force and transportation data. This information includes:

1. Type units to be employed.

2. Type units to be deployed to support the OPLAN, with a priority indicating the desired sequence for their arrival at POD;

3. Routing of forces to be deployed.

4. Mobility data associated with deploying forces.

5. Personnel and logistic movement to be conducted concurrently with the deployment of forces; and

6. Estimate of transportation requirements for both forces and support requirements.

TIME-PHASED FORCE DEPLOYMENT LIST

A list from the data contained on the TPFDD which includes a time-phased force list, identifies type units to be deployed, and provides data concerning their destination. A list that is time-phased has been ordered by arrival at POD.

TIME-PHASED TRANSPORTATION REQUIREMENTS LIST (TPTRL)

A list from the data contained on the TPFDD which defines the movement requirements and includes a time-phased list of type units/mobility echelons, fillers and replacement personnel, and bulk supplies to be transported by air or sea to support an OPLAN; provides mobility data related to these deployments and estimates movement requirements to be fulfilled by both common-user lift resources and assigned transportation resources.

TIME-SHARING SUBSYSTEM (TSS)

A system in the H6000 computer which allows several users to share the resources of one subsystem. JOPS III modules have been recently reprogrammed to operate under the TSS supervisor. The TSS version of JOPS III allows several users to share the same programs, thus reducing the core requirements for JOPS III support.

TPFDD INTERFACE UTILITY MODULE (T/I)

This JOPS III subsystem provides the following functions:

(1) Conversion of DEPDA data to TPFDD/SRF format;

(2) A SORT function to allow users to manipulate the sequence of the TPFDD/SRF.

(3) Analysis of transportation requirements at each point in the deployment, and analysis of through-put capabilities at POEs and PODs.

TRANSPORTATION FEASIBILITY ESTIMATOR (TFE)
The JOPS III subsystem which compares transportation requirements to transportation assets and capabilities to determine whether or not OPLAN deployment requirements can be satisfied by available lift resources.

TYPE UNIT
A type of organizational entity established within the Armed Forces and uniquely identified by a UTC.

TYPE UNIT DATA FILE (TUCHA)
The TUCHA file provides the planner with standard planning data on movement characteristics for personnel, cargo, and accompanying supplies associated with deployable type unit of fixed composition. The file contains the weight and cube of selected cargo categories, physical characteristics of the cargo, and the number of personnel requiring non-organic transportation.

UNIT DESIGNATION LIST
A list of actual units by UIC designated to fulfill requirements of a force list.

UNIT IDENTIFICATION CODE (UIC)
A six-character, alphanumeric code that uniquely identifies each Active, Reserve, and National Guard unit of the Armed Forces.

UNIT TYPE CODE (UTC)
A five-character alphanumeric code which is associated with and allows each type unit to be categorized into a kind or class having distinguishing characteristics.

UNIT-RELATED EQUIPMENT AND SUPPLIES
All equipment and supplies requiring transportation to an area of operations that are assigned to a specific unit or that are designated as accompanying supplies.

VISUAL INFORMATION PROCESSOR (VIP)
A remote terminal which employs a cathode-ray tube with a keyboard to allow communication between the computer and the user.

WARNING ORDER
A communication device used by commanders to advise subordinates of impending action. A warning order is generally a prelude to the ALERT ORDER, OPORD, or FRAGORD. The JCS may use the warning order as a planning directive to initiate Phase III of the Crisis Action System.

SECTION II -- ACRONYMS AND ABBREVIATIONS

The following is a list of selected acronyms that are frequently used in planning. Generally acronyms and abbreviations should be avoided in the writing of joint plans and orders. If a long title must be used repeatedly, the acronym or abbreviation may be employed provided the first time it is used the long title is spelled out fully along with its related acronym or abbreviation.

ACP	Allied Communications Publication
AD	Advanced Deployment Posture
ADP	Automatic Data Processing
ADPE	Automatic Data Processing Equipment
AEC	Atomic Energy Commission
AFM	Air Force Manual
ALCANUS	Alaska-Western Canada-Western US
APOD	Aerial Port of Debarkation
APOE	Aerial Port of Embarkation
APORTS	Aerial Ports and Operating Bases
ARRS	Aerospace Rescue and Recovery Service
ASAP	as soon as possible
ASCP	Army Strategic Capabilities Plan
ASSETS	Transportation Assets File
ASW	Antisubmarine Warfare
AUTODIN	Automatic Digital Network
BIDE	Basic Identity Data Elements
CAS	Crisis Action System
CBR	Chemical, Biological, Radiological
C&D	Cover & Deception
C-E	Communications-Electronics
CEF	Civil Engineering Files
CENTO	Central Treaty Organization
CEP	Circular Error Probable
CESP	Civil Engineering Support Plan
CESPG	Civil Engineering Support Plan Generator
CHSTR	Characteristics of Transportation Resources
CIA	Central Intelligence Agency
CINCUNC	Commander In Chief, United Nations Command
COACT	Combat Activities Report
COI	Communications Operating Instructions
COMINT	Communications Intelligence
COMUSJTF	Commander, US Joint Task Force
CONPLAN	Operation Plan in Concept Form
CONUS	Continental United States
COP	Contingency Operation Plan
CRIMREP	Crisis Management Information Report
CRITIC	Critical Intelligence
CRT	Cathode-Ray Tube or VIP Terminal
CUSRPG	Canada-US Regional Planning Group
CW	Chemical Warfare
DCA	Defense Communications Agency
DCS	Defense Communications System

DEFCON	Defense Condition
DEPDA	Deployment Data File
DEPREP	Deployment Reporting System
DFSC	Defense Fuel Supply Center
DGZ	Desired Ground Zero
DIA	Defense Intelligence Agency
DISUM	Daily Intelligence Summary
DMA	Defense Mapping Agency
DNA	Defense Nuclear Agency
DOD	Department of Defense
DODI	Department of Defense Instruction
DSA	Defense Supply Agency
DTG	Date-Time-Group
EAD	Earliest Arrival Date
ECCM	Electronic Counter-Countermeasures
ECM	Electronic Countermeasures
EDP	Emergency Defense Plan
EEFI	Essential Elements of Friendly Information
E&E	Evasion and Escape
EEI	Essential Elements of Information
ELINT	Electronic Intelligence
EW	Electronic Warfare
FIDP	Foreign Internal Defense Plan
FM	Army Field Manual
FMFM	Fleet Marine Force Manual
FORSTAT	Force Status and Identity Report
FPF	Force Package File
FRG	Force Requirements Generator
FYDP	Five-Year Defense Plan
GEOFILE	Specified Geolocation Code File
GEOLA	Geolocation Code File
GEOLOC	Geolocation Code
GEOREF	Geographic Reference System
HIS	Honeywell Information System
HUMINT	Human Source Intelligence
IADB	Inter-American Defense Board
ICR	Intelligence Collection Requirement
ID	Increased Deployment Posture
IMAPS	Integrated Military Airlift Planning System
INTSUM	Intelligence Summary
IPSP	Intelligence Priorities for Strategic Planning
JANAP	Joint Army, Navy, Air Force Publications
JCS	Joint Chiefs of Staff
JCSM	Joint Chiefs of Staff Memorandum
JDAC	Direct Access Communication
JIEP	Joint Intelligence Estimate for Planning
JIS	JOPS Interim Software
JLPPG	Joint Logistics, Personnel & Policy Guidance
JLREID	Joint Long-Range Est Intelligence Document
JLRSS	Joint Long-Range Strategic Study
JOPS	Joint Operation Planning System
JOP III	WWMCCS Standard Computer-Based System to Support Joint Operations Planning
JPAM	Joint Program Assessment Memorandum

EXHIBIT JOP510-1 (Cont)

JRS	Joint Reporting Structure
JSCP	Joint Strategic Capability Plan
JSPD	Joint Strategic Planning Document
JSPS	Joint Strategic Planning System
JTD	Joint Table of Distribution
JTF	Joint Task Force
JUWTF	Joint Unconventional Warfare Task Force
LAD	Latest Arrival Date
LD	Loaded Deployment Posture
LERTCON	Alert Condition
LOC	Line(s) of Communication
MAAG	Military Assistance Advisory Group
MAC	Military Airlift Command
MAF	Marine Amphibious Force
MAP	Military Assistance Program
MAPS	Mobility Analysis & Planning System (MTMC)
MCP	Marine Corps Capabilities Plan
MD	Marshalled Deployable Posture
MEQPT	Major Equipment Code File
MFF	Medical Factors File
MILCON	Military Construction Program
MOP	Memorandum of Policy
MPP	Medical Planning Program
MRG	Movement Requirements Generator
MSC	Military Sealift Command
MTMC	Military Traffic Management Command
NATO	North Atlantic Treaty Organization
NAVSECGRUCOM	Naval Security Group Command
NCA	National Command Authorities
NCP	Navy Capabilities Plan
ND	Normal Deployment Posture
NMCC	National Military Command Center
NMCS	National Military Command System
NOFORN	Not Releasable to Foreign Nationals
NSA	National Security Agency
NSC	National Security Council
NSDA	Non-Self-Deployable Aircraft
NSP	Navy Support Plan
NWP	Naval Warfare Publication
OJCS	Organization of the Joint Chiefs of Staff
OPCON	Operational Control
OPLAN	Operation Plan in Complete Form
OPPA	Operation Plan Package Appraisal
OPREP	Commander's Operational Report
OPSEC	Operations Security
OSD	Office of Secretary of Defense
PDD	Package Designation and Description
PFF	Planning Factors File
PIN	Plan Identification Number
POD	Port of Debarkation
POE	Port of Embarkation
POL	Petroleum, Oils, and Lubricants
PORTS	Port Characteristics
PPBS	Planning, Programming, and Budgeting system

EXHIBIT JOP510-1 (Cont)

PSYOP	Psychological Operations
PUB	Publication
PW (POW)	Prisoner of War
PWIN	Prototype WWMCCS Intercomputer Network
PWRS	Prepositioned War Reserve Stock
RDD	Required Delivery Date
RLT	Regimental Landing Team
SAF	Special Action Force
SAR	Search and Rescue
SEACOP	Strategic Sealift Contingency Planning System
SEASTAG	Southeast Asia Standardization Agreement
SEATO	Southeast Asia Treaty Organization
SERE	Survival, Evasion, Resistance and Escape
SIGINT	Signal Intelligence
SITREP	Situation Report
SITSUM	Situation Summary
SM	System Monitor
SNL	Standard Nomenclature List
SOLOG	Standardization of Operations and Logistics
SOP	Standard Operating Procedure
SRF	Summary Reference File
STANAG	Standardization Agreement
TFE	Transportation Feasibility Estimator
T/I	TPFDD Interface
TO	Table of Organization
TOA	Transportation Operating Agency
TOE	Table of Organization and Equipment
TPFDD	Time-Phased Force and Deployment Data
TPFDL	Time-Phased Force and Deployment List
TPTRL	Time-Phased Transportation Requirements List
TUCHA	Type Unit Characteristics File
TYPEA	Unit Type Code File
UCP	Unified Command Plan
UIC	Unit Identification Code
ULC	Unit Level Code
UNAAF	Unified Action Armed Forces
USBRO	US Base Requirements Overseas
USG	US Government
USIA	US Information Agency
USREDCOM	US Readiness Command
UTC	Unit Type Code
UW	Unconventional Warfare
VOL	Volume
WECON	Weather Control
WMP	USAF War and Mobilization Plan
WWMCCS	Worldwide Military Command and Control System

EXHIBIT JOP510-1 (Cont)

In this simulation you will respond to the dialogue of the file paging module. Specifically, you will respond as follows:

1. System will display: ENTER MODULE TO BE EXECUTED

1A. You will respond -------------- F12G or *PAGE (If F12G, go to step 3; else go to next step.)

2. System will display: ENTER FILE TO BE PAGED

2A. You will respond -------------- TPFDD

3. System will display: THIS MODULE ALLOWS SCANNING OF THE TPFDD FILE AND DETAILED INSPECTION OF SPECIFIC FORCE RECORDS. DO YOU WANT TO CONTINUE?

3A. You will respond -------------- YES or NO. If no, go to step 5; else go to next step.

4. System will display: THE TPFDD FILE IS READY FOR SCANNING, ENTER FP TO DISPLAY THE FIRST PAGE. A DETAIL DISPLAY OF A SPECIFIC TPFDD RECORD CAN BE PROVIDED AT ANYTIME THE PROGRAM IS WAITING A USER RESPONSE BY ENTERING THE RECORD SEQUENCE NUMBER OF THAT FORCE. TO TERMINATE SCANNING, ENTER *ENDPAGE.

4A. You will respond -------------- FP to display the first page, then any of the other paging commands as desired, FP/NP/PP/LP or PAGE NNN. Enter sequence number for detail display or *ENDPAGE to terminate the module. If *ENDPAGE, go to step 5 when executing as a stand alone module or if under the monitor paging capability, return to the point from which you came. Any valid page will display and await further instructions.

5. System will display: ENTER MODULE TO BE EXECUTED.

EXHIBIT JOP510-2

LESSON JOP510 - The Basis for an Operation Plan

Questions similar to those below will be asked in the final examination for lesson JOP510. You should study the JOPS manuals and lesson material in an off-line mode prior to taking the final exam. When you are satisfied with your answers, re-enter the lesson and take the exam. You must achieve a score of 70% or better to satisfactorily complete the lesson.

1. Identify the phases in the planning cycle.

2. What factor determines the need for an Operation Plan (OPLAN)?

3. Who assigns responsibility for developing OPLANs?

4. Who is responsible for developing OPLANs?

5. What planning tools are used in OPLAN development?

6. What reporting system was used prior to formalizing JOPS and is currently used in conjunction with JOPS?

7. Identify the seven basic subsystems of JOPS III (ADP Support).

8. Which JOPS III subsystem allows interaction between the planner and JOPS III software?

9. Which JOPS III subsystem is used to convert an existing OPLAN to JOPS format?

10. Which JOPS III subsystem is used to select forces and determine routing?

11. Which JOPS III subsystem is used to determine nonunit-related requirements?

12. Which JOPS III subsystem is used to test an OPLAN for transportation feasibility?

13. What is a Time-Phased Force Deployment Data (TPFDD) file?

EXHIBIT JOP510-3

TITLE: Final Project for Lesson JOP510

The final project for this lesson is to simulate paging the TPFDD file using the JOPS System Monitor. The coursewriter will simulate the JOPS III Software and you, as the planner, will provide the necessary responses.

The situation is similar to that discussed in JOP510. A threat has been identified; country Zebra is being threatened by country X-Ray. JCS has tasked your commander (CINCBOLD) with the responsibility for developing an OPLAN to defend Zebra. You are the lead planner.

You have decided to convert an existing OPLAN rather than begin a new one, if a suitable OPLAN already exists. You have found two OPLANs that may satisfy the requirements, but you want to page through their TPFDD files page by page to study them.

The Project: When directed by JOP510, you are to input responses to accomplish the following actions. Assume the job initiation procedures have been done and the message "ENTER MODULE TO BE EXECUTED" has been displayed.

A. Initiate file paging feature with the general paging command.
B. Page the TPFDD file.
C. Display the first page.
D. Display page 100.
E. Display the previous page.
F. Display the last page.
G. Terminate the file paging feature.
H. Terminate JOPS.

EXHIBIT JOP510-4

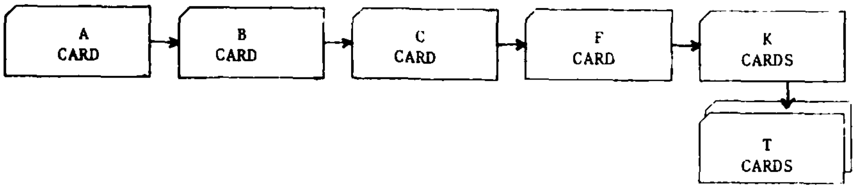
FBN	FORCE CATEGORY	DESCRIPTION	PIC
A1	GROUPING	INFANTRY DIVISION	P
A1A	INDEPENDENT	INFANTRY BRIGADE	
A1C	PRIMARY PARENT	INFANTRY BRIGADE	X
A1C1	SUBORDINATE	BRIGADE HEADQUARTERS	
A1C2	SECONDARY PARENT	INFANTRY BATTALION	X
A1C21	SUBORDINATE	BATTALION HEADQUARTERS	
A1C22	SUBORDINATE	INFANTRY COMPANY	

EXHIBIT JOP510-5

FRN	FORCE CATEGORY	DESCRIPTION	PIC
A1		INFANTRY DIVISION	X
A1A		INFANTRY BRIGADE	X
A1A1		BRIGADE HEADQUARTERS	
A1A2		INFANTRY BATTALION	
A6A		ARMORED BRIGADE	
A2B		ARTILLERY BATTALION	X
A2B1		BATTALION HEADQUARTERS	
A2B2		ARTILLERY BATTERY	
A3A		ARMORED BRIGADE	
A3A1		BRIGADE HEADQUARTERS	
A3A2		ARMORED BATTALION	X
A3A21		BATTALION HEADQUARTERS	
A3A22		TANK COMPANY	

EXHIBIT JOP510-6

NON-SPLIT SHIPMENT FORCE REQUIREMENT



SPLIT SHIPMENT FORCE REQUIREMENT

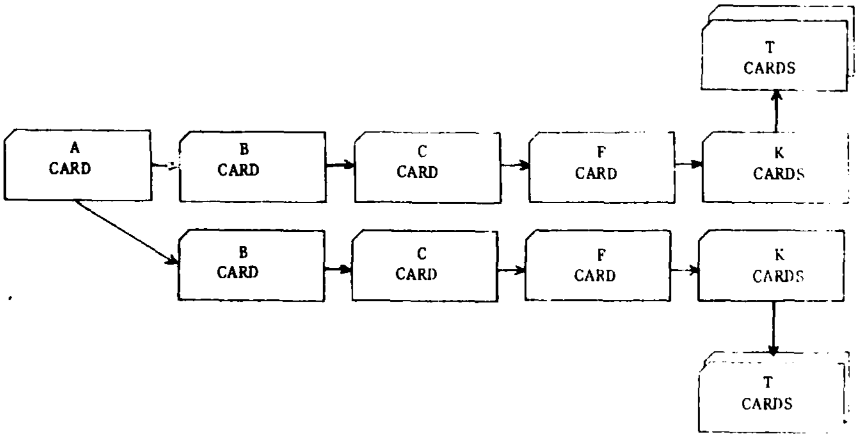


EXHIBIT JOP510-7

LESSON JOP520 - Plan Development; Convert Existing Plan

Questions similar to those below will be asked in the final examination for lesson JOP520. You should study the JOPS manuals and lesson material in an off-line mode prior to taking the final exam. When you are satisfied with your answers, re-enter the lesson and take the exam. You must achieve a score of 70% or better to satisfactorily complete the lesson.

1. Does JOPS have the capability for selecting previously prepared OPLAN's to counter a new threat?

2. What is the main difference between DEPREP card format and the DEPDA file?

3. What is the primary difference between the DEPDA and TPFDD files?

4. Can an existing OPLAN, not in JOPS format (but in DEPREP format), be used by JOPS III software?

5. Which JOPS subsystem converts a DEPDA file into TPFDD/SRF formats?

6. Which module within the Interface Utility (T/I) subsystem makes the conversion of DEPDA to TPFDD?

7. Which module in the T/I subsystem sorts the TPFDD?

8. What information does the TUCHA file provide?

EXHIBIT JOP520-1

TITLE: Intermediate Simulation of Module DØ1A

As an intermediate project for lesson JOP520, you are to convert an existing OPLAN in DEPDA file format into a TPFDD and associated SRF. The lesson contains a simulation of the user's interaction with module DØ1A. You are expected to answer questions posed by DØ1A as a planner would based upon the following criteria:

A. The problem DEPDA has previously been loaded on line via F60.
B. The problem DEPDA is a temporary file on disk.
C. OPLAN number: 9876X.
D. Convert the 600 force records on the DEPDA and place them on the TPFDD file.
E. Convert the 499 nonunit-related records and place these on the TPFDD file. (The nonunit-related forces refer to resupply and personnel.)
F. Security Classification: UNCLASSIFIED.
G. Produce the record summaries of these forces.
H. Produce applicable reports on both force and nonunit-related records.
I. Send all output to remote line printer "AB".

EXHIBIT JOP520-2

TITLE: Intermediate Simulation of Module DØ1B

As an intermediate project for lesson JOP520, you are to update an existing TPFDD/SRF from an updated DEPDA. The lesson contains the simulation of DØ1B and will anticipate responses using the following criteria:

A. The problem DEPDA has been loaded on line via F60 start option.
B. OPLAN ID: 9876X.
C. The TPFDD is not in the correct sequence.
D. User-supplied date: 750101.
E. Update force records.
F. Update only nonunit-related cargo records (G Card data).
G. Produce no record summaries.
H. Produce maintenance report on remote line printer "AB".

EXHIBIT JOP520-3

TITLE: An Intermediate Simulation of DØ3

As an intermediate project for Lesson JOP520 you are to sort the TPFDD into POD-LAD ascending sequence. The lesson contains the simulation of DØ3 and will anticipate responses using the following as criteria:

A. Sort the entire file according to POD-LAD.
B. The force record filed for POD-LAD is coded as '55'.
C. Save sequence numbers to allow the TPFDD to be re-sorted into its present sequence.

You will sort the TPFDD file as directed above. This will enable you to execute module M10 at a later date.

EXHIBIT JOP520-4

TITLE: Final Project for Lesson JOP520

As a final project for lesson J0520 you are to convert an existing OPLAN in DEPDA file format into a TPFDD file. The coursewriter will simulate the JOPS III Software, and you as the planner will input the required data to accomplish the conversion.

The situation is similar to that discussed in the lesson, country Zebra is threatened by country X-ray. You are the planner. You must develop an OPLAN to defend the country Zebra. You have found an OPLAN (Number 9876X) that contains nearly all of the necessary forces and it can be modified to satisfy the new planning requirement. Now you must convert the DEPREP file to a TPFDD file.

After the final exam in JOP520 and when directed by the coursewriter, begin the simulation by requesting the TPFDD Interface module "DØ1A." After the TPFDD has been created, place the TPFDD file on magnetic tape (reel #2780) using FRG module F60. The following necessary information is provided:

A. The problem DEPDA has been loaded to disk via F60.
B. OPLAN Number (ID): 9876X
C. Classification: UNCLASSIFIED
D. Convert only the 400 force records.
E. Do not convert nonunit-related cargo and personnel requirements.
F. Produce applicable reports.
G. Direct output to remote printer AB.

You may reference the T/I manual and the FRG manual as necessary.

EXHIBIT JOP520-5

TITLE: Simulation of FRG Modules F6Ø, F1Ø, F11, and F14

The practice exercises for this lesson consist of four simulations which are to be executed in sequence. Recalling from JOP520, you converted a DEPDA file to a TPFDD file and placed the file on magnetic tape. In this lesson, you are required to continue development of your OPLAN, using several Force Requirements Generator (FRG) modules.

This exhibit provides information for the necessary inputs. All displays, input data, and files are unclassified and resemble the actual system only to the point necessary to simulate meaningful results. Responses will parallel system responses as closely as possible.

Simulation 1.

Input the correct responses to transfer the TPFDD file (OPLAN 9876X) from magnetic tape (Reel #2780) to disk using FRG Module F6Ø. There is no TFE control file at this time. Do not change identification data.

Simulation 2.

Since OPLAN 9876X is an OPLAN developed to counter a threat to country Tango rather than country Alpha, you want to modify the OPLAN identification data. Using FRG Module F1Ø, provide the necessary input to reflect the following facts:

A. Produce a list of available JOPS modules at Remote "AB."
B. OPLAN Date: 15 June 1975.
C. Classification: UNCLASSIFIED
D. New OPLAN Number: 4321A
E. OPLAN Identification: CINCGULF Plan 4321A (U) For Alpha.
F. Task Organization: GULFCOM Defense of Alpha.
G. Objective Area: Alpha and All Bordering Countries.
H. Exemption Category: No change.
I. OPLAN Change Number: ØØ
J. Concept of Operations:
 Line 1: CINCGULF OPLAN to Defend Alpha
 Line 2: Utilizing All Available Forces
K. Produce an OPLAN Identification Report at Remote "AB."

Simulation 3.

Using FRG Module F11, provide the essential information to add the following standard forces to the TPFDD file (OPLAN 4321A).

REFCODE	ORG	POD	DEST	ONCALL	INPLACE
3FQAA	F	ZEB3	ZEB1	NO	
24B9E	A	OOAN	OOAT		NO
2QQQQ	A				NO

Additional Instructions:
A. Print the force list at Remote "AB".
B. Do not print nonforce records.

Simulation 4.

Using FRG Module F14, input the correct data to add the following nonstandard forces to the TPFDD file (OPLAN 4321A). Only partial information exists at planning time.

A Card Information

FRN	PROV ORG	SERV	UTC	ULC	Force Description	FIC	AUTH PERS	PIC
AØC	7	A	Z99BB	CO	Infantry Company	8	00050	
J1B	F	F	Z99BB	SQ	Fighter Squadron	8	00018	

B Card Information

	Intermediate Location		POD		Destination	
FRN	MODE	GEOLOC CODE	MODE	GEOLOC CODE	MODE	GEOLOC CODE
AØC	A	SOMW	A	ZEB3	A	ZEB1

(Do not create a B card for FRN J1B.)

C Card Information

		I	*	* * UIC * *				F				
	F	N	S					K	ORIG	POE	M	S
	R	S	E	COMP				I	GEO-	GEO-	O	R
	A	R	R	ORG			Description or	N	LOC	LOC	D	C
FRN	G	T	V	IDEN	UTC	ULC	Unit Name	D	CODE	CODE	E	E
AØC	Ø	Ø	W	ABTMM	Z99BB	CO	3950 INFCO	Y	USAT	USA9	L	G

MOVEMENT CHARACTERISTICS OF CARGO

FRN: AØC

Item	Cargo Category Code
Jeep	A2F
Truck	A1F
Packaged Fuel	J7A
Ammunition	M7C

NOTE: Specific details for "K" and "T" formats have not been provided and will not be required.

TITLE: Simulation of FRG Modules F2Ø and F21

The practice exercises for this lesson consist of two simulations which are to be executed in sequence, F2Ø and F21. As in prior lessons, the coursewriter will provide for simulated JOPS III Software dialogue and you, as the planner, must input the required responses to accomplish the actions requested.

The situation posed in the lesson exercises in JOP510 through JOP530 is continued. You are a planner responsible for creating an OPLAN to defend country Alpha. In JOP520, you converted existing OPLAN 9676X which was similar in composition to what is required in the new OPLAN. In JOP530, you changed the OPLAN Number to 4321A and added standard and nonstandard forces. Now in JOP540 you are to modify certain forces to bring them in line with requirements. As you are instructed by the coursewriter, complete the following simulations in sequence. All files and displayed data are fictitious and unclassified.

You find that you must delete several forces from the TPFDD file. Using FRG Module F2Ø, provide responses to accomplish the following actions:

A. Delete the following forces from the Force List: Line Item Units with Sequence Numbers 1, 3 through 5 and 7.
B. Do not save the deleted units.
C. Produce a report of the forces deleted and print a Revised Force List through remote printer "AB".
D. Do not print nonforce records.

Simulation 2:

In your present TPFDD file, you discover two forces which may be tailored by modifying specific fields. Using FRG Module F21, provide correct responses to accomplish the following specifications:

A. Make the following changes to Line (SEQ.#) 9:
   Code A: Change Service Code from US Army (A) to Joint (J).
   Code B: Show the Mode to the Intermediate Location to be "A" (AIR) and Source (SRCE) of Transportation to be "K" (MAC).
   Code C: Change the SERV Code to "D" to conform to Code A.
B. Make the following changes to Line 18:
   Code A: Change the ULC to "MAG" (Marine Air Group).
   Code B: (No change)
   Code C: (No change)
C. To Remote "AB", print a report of modifications made to the Force List, and a copy of the Revised Force List.
D. Do not print nonforce records.

EXHIBIT JOP540-1

TITLE: Final Project for FRG Lessons, JOP530 and JOP540

In order to satisfactorily complete the FRG lessons (JOP530 and JOP540), you must complete the following six simulations involving generating (selecting and tailoring) forces. As in prior lessons, the coursewriter will provide for simulated JOPS dialog, and you as the planner must input the required responses to accomplish the actions requested.

The situation posed in final ,rojects of JOP510 and JOP520 is continued. You are a planner who is responsible for creating an OPLAN to defend country Zebra. In JOP520, you converted an existing OPLAN (9876X) which was similar in composition to what is required in the new OPLAN. In JOP530, you did not have a final project; you only practiced adding force requirements to the TPFDD. Therefore, in the final project of this lesson, you will generate the force requirements. This involves changing identification data, adding and tailoring forces on the TPFDD.

Simulation 1.

Input the correct responses to transfer the TPFDD file (OPLAN 9876X) from magnetic tape (Reel 2780) to disk using FRG Module F6Ø. Do not change identification data.

Simulation 2.

Since OPLAN 9876X is an OPLAN developed to counter a threat to country Tango rather than country Zebra, you want to modify the OPLAN identification data. Using FRG Module F1Ø, provide the necessary input to reflect the following facts.

A. Produce a list of available JOPS Modules at Remote "AB".
B. OPLAN Date: 05 June 1979
C. Classification: Unclassified
D. New OPLAN Number: 1234Z.
E. OPLAN Identification: "CINCBOLD Plan 1234Z (U) for Zebra."
F. Task Organization: "BOLDCOM Defense of Zebra."
G. Objective Area: "Country Zebra."
H. Exemption Category: No change.
I. OPLAN Change Number: 00.
J. Concept of Operations: "CINCBOLD OPLAN To Defend Country Zebra."
K. Produce an OPLAN Identification Report at Remote "AB".

Simulation 3.

Using FRG Module F11, provide the essential information to add the following standard forces to the TPFDD file (OPLAN 1234Z).

REFCODE	ORG	POD	DEST	ONCALL	INPLACE
3FQAA	7	ZEB3	ZEB1	NO	
24B9E	A	GKUB	VUYC		NO
9GSAA	M				NO

Additional Instructions:
A. Print the force list at Remote "AB".
B. Do not print nonforce records.

EXHIBIT JOP540-2

Simulation 4.

Using FRG Module F14, input the correct data to add the following nonstandard forces to the TPFDD file (OPLAN 1234Z). Only partial information exists at planning time.

A Card Information

	PROV							
FRN	ORG	SERV	UTC	ULC	Force Description	FIC	PERS	PIC
---	-	-	-----	---	-----------------	-	-----	-
AØC	7	A	Z99BB	CO	Infantry Company	8	00050	
J1B	F	F	Z99BB	SQ	Fighter Squadron	8	00018	

B Card Information

		Intermediate										
		Location				POD				Destination		
FRN	MODE	SOURCE	GEOLOC	CODE	MODE	SOURCE	GEOLOC	CODE	MODE	SOURCE	GEOLOC	CODE
---	-	-	----		-	-	---		-	-	---	
AØC	A	K	SOMW		A	K	ZEB3		A	K	ZEB1	

(Do not create a B card for FRN J1B.)

C Card Information

		I	*	* UIC *				F				
	F	N	S					K	ORIG	POE	M	S
	R	S	E	COMP				I	GEO-	GEO-	O	R
	A	R	R	ORG			Description or	N	LOC	LOC	D	C
FRN	G	T	V	IDEN	UTC	ULC	Unit Name	D	CODE	CODE	E	E
AØC	Ø	Ø	W	ABTMM	Z99BB	CO	3950 INF CO	Y	USA9	USAT	L	G

MOVEMENT CHARACTERISTICS OF CARGO

FRN: AØC

Item	Cargo Category Code
-----	---
Jeep	A2F
Truck	A1F
Packaged Fuel	J7A
Ammunition	M7C

NOTE: Specific details for "K" and "T" formats have not been provided and will not be required.

Simulation 5:

You find that you must delete several forces from the TPFDD file. Using FRG Module F2Ø provide responses to accomplish the following actions:

A. Delete the following forces from the Force List: Line Item Units with Sequence Numbers 1 and 5 through 7.
B. Do not save the deleted units.
C. Using Remote "AB", print a report of the forces deleted and print a Revised Force List.
D. Do not print nonforce records.

Simulation 6:

In your present TPFDD file, you discover two forces which may be tailored by modifying specific fields. Using FRG Module F21, provide correct responses to accomplish the following specifications:

A. Make the following changes to TPFDD Sequence Number 9:
   Code A: Change Service Code from US Army (A) to Joint (J).
   Code B: Show the Mode to the Intermediate Location to be "A", and Source (SRCE) TO BE "K" (VIA MAC).
   Code C: Change the SERV Code to "J" to conform to Code A.
B. Make the following changes to TPFDD Sequence Number 18:
   Code A: Add Code "F" for providing organization, and change the ULC to "SQ".
   Code B: (No change)
   Code C: (No change)
C. To Remote "AB", print a report of modifications made to the Force List, and a copy of the Revised Force List.
D. Do not print nonforce records.

TITLE: Extract of the PFF Report

UNCLASSIFIED

JOPS INTERIM SOFTWARE
PLANNING FACTORS REPORT

TABLE OF CONTENTS

EXHIBIT JOP550-1

TITLE: Example of Edit Report Listing

UNCLASSIFIED

```
                                 JOPS INTERIM SOFTWARE                    (DATE)

                                      EDIT LISTING

             CARD IMAGE

         1         2         3                                        8  INVALID
1234567890123456789012345678890 -------------------------------------0  COLUMNS

A45MD5W890123456789                                                      NONE

A45MD5A890123456789                                                      NONE

A46FDIR890123456789                                                      3

A20 E MD5M010030045                                                      5

                                                                         10
```

EXHIBIT JOP550-2

TITLE: Example of Update Error Report

UNCLASSIFIED

```
                                 JOPS INTERIM SOFTWARE

                                  UPDATE ERROR REPORT

             CARD IMAGE                                          (DATE)

         1         2                                      8      INV REASON REJECTED
12345678901234567890 ----------------------------------  890     COLS ERROR DESCRIPTION

A45MD5M                                                          4-7 CANNOT DELETE-
                                                                 RSPLY CATEGORY MD5M
                                                                 NOT IN FILE.
```

EXHIBIT JOP550-3

TITLE: Example of Consistency Check Error Report

```
UNCLASSIFIED

                              JOPS INTERIM SOFTWARE                  (DATE)

                              PFF CONSISTENCY CHECKS

COUNTRY CODE IN RECORD A25 AND RECORD A60 DOES NOT MATCH             ZPBEKS

                                                               UNCLASSIFIED
```

EXHIBIT JOP550-4

TITLE: MRG Report Summary

REPORT	MODULE	DESCRIPTION
Planning Factors	M10	Reflects contents of PFF
Edit	M10	Lists every PFF input and indicates fields in error
Update	M10	List errors detected when updating input on PFF
Consistency Check Error	M10,M20	Lists discrepancies among record types on the PFF
PWRS	M20	Lists consumption requirements for first 44 days
TPFDD Summary Input	M20	Lists MRG requirements defined in TPFDD

EXHIBIT JOP550-5

TITLE: Service ID Codes (Table 3, Appendix C, MRG Users Manual)

TAB 3. Service ID Codes

Code	Meaning
A	U.S. Army
F	U.S. Air Force
M	U.S. Marine Corps
N	U.S. Navy
D	Allied Marine Corps
Q	Allied Air Force
T	Allied Navy
V	Allied Army

TITLE: Supply Class/Subclass Codes (Table 5, Appendix C, MRG Manual)

TAB 5. Supply Class/Subclass Codes

Class		Subclass
1 -	Subsistence	A - Air (Inflight rations) R - Refrigerated subsistence S - Nonrefrigerated subsistence C - Combat rations
2 -	Clothing, individual equipment, tentage, organizational tool sets and tool kits, hand tools, etc.	B - Ground support material E - General supplies F - Clothing and textiles M - Weapons T - Industrial supplies
3 -	POL	A - Air W - Ground
4 -	Construction	None
5 -	Ammunition	A - Air W - Ground
6 -	Personal demand items	None
7 -	Major end items	A - Air B - Ground Support Material D - Administrative vehicles G - Electronics K - Tactical vehicles L - Missiles M - Weapons N - Special Weapons
8 -	Medical Materiel	None

EXHIBIT JOP550-7

TITLE: Supply Class/Subclass Codes (Table 5, Appendix C, MRG Manual)

TAB 5. Supply Class/Subclass Codes

Class		Subclass
9 -	Repair parts (less medical peculiar repair parts): All repair parts and components, including kits, assemblies and subassemblies required for maintenance support for all equipment.	A - Air B - Ground support material D - Administrative vehicles G - Electronics K - Tactical vehicles L - Missiles M - Weapons N - Special Weapons T - Industrial supplies
0 -	Material to support military programs	None

TITLE: Security Classification Codes (Tab 6, Appendix C, MRG Manual)

TAB 6. Security Classification Codes

Code	Meaning
U	Unclassified
E	EFTO
C	Confidential
H	Confidential NOFORN
S	Secret
L	Secret NOFORN
T	Top Secret
O	Top Secret NOFORN

EXHIBIT JOP550-8

TITLE: Simulation of MRG Module M60

Execute M60 three (3) times according to the specification below:

1. You are required to generate the nonunit-related requirements for OPLAN 4321A. This is the first MRG operation on the plan; therefore, you must have a PFF with a good set of planning factors constructed on it. But you do not have a PFF; so you must first establish a PFF. Input the proper response to establish a PFF.

2. There may be times when you will need to establish and save more than one PFF in order to get around certain limitations. So, just for practice, you are required to save to tape the PFF which you just established in part 1.

3. If you ever need to establish and save more than one PFF to tape, you should know how to restore the PFF back to disk from the save tape. Therefore, just for practice, you are required to restore to disk the PFF which you just save to tape 9999T in part 2.

EXHIBIT JOP550-9

TITLE: Master Display (PFF UPDATE PARAMETERS)

MASTER PAGE LIST OF PFF PARAMETER ENTRY FORMATS

A01-PLANNING FACTORS REPORT COMMENTS
A10-MISCELLANEOUS PLANNING FACTORS
A11-TIME PERIODS SPECIFICATIONS
A12-RESUPPLY AND BUILD-UP POE'S
A13-AIR RESUPPLY PERIODS
    SPECIFICATIONS
A15-PERCENT DISTRIBUTION OF RESUPPLY
    AND BUILD-UP AMONG POE'S
A20-SUPPLY BUILD-UP POLICIES
A25-RETROGRADE CARGO PERCENTAGES
A26-INTRACOASTAL CARGO PERCENTAGES
A27-OBJECTIVE COUNTRY UNKNOWN GEOCODE
A35-SUPPLY CLASS CONSUMPTION FACTORS
A45-CONSUMPTION FACTORS
    AGGREGATION CRITERIA
A50-THEATER ADJUSTMENT MULTIP
A55-OFF-SHORE COUNTRY DEFIN
A56-U.S.PERSONNEL IN-PLACE IN
    OFF-SHORE COUNTRIES
A60-OBJECTIVE AREA COUNTRY DEFIN
A61-PWRS CUT-OFF DAYS
A65-FWMAF PERSONNEL STRENGTHS
A70-SUPPLY CLASS DESCRIPTIONS &
    ST-TO-MT CONVERSION MULTIP
A80-CRITERIA FOR AGGREGATING FORCE LIST UNITS

A. TO SELECT THE NEXT MASTER PAGE, ENTER M.
B. TO SELECT A PFF PARAMETER ENTRY FORMAT, ENTER THE CORRESPONDING 3 CHARACTER IDENTIFIER FROM THE LIST ABOVE (E.G., A20).
C. TO TERMINATE PROCESSING THE PFF FILE, ENTER END.

EXHIBIT JOP550-10

TITLE: Master Display (Overall Composition Rules)

MASTER PAGE OVERALL COMPOSITION RULES

In preparing inputs to modify the PFF file for a particular OPLAN, rules must be adhered to in order to insure that a consistent set of planning factors will be defined. The MRG function cannot proceed until consistent planning factors exist. Error messages will be generated when inconsistencies result.

Resupply Categories--A resupply category is defined as a cargo category and an associated supply class/subclass. Supply class/subclasses will be aggregated into 14 or fewer unique resupply categories via entry format A45. Only the resupply categories defined by A45 may be used which require/permit the entry of a resupply category code. These inputs are A13, A15, A20, and A61.

FWMAF Services--In addition to US Forces, the forces of up to 3 FWMAF services may be included in the evaluation of an OPLAN via entry format A35. Only the FWMAF services defined by A35 may be used which require/permit the entry of a FWMAF service code. These inputs are A12, A15, A26, A50, A65, and A70.

TO SELECT THE NEXT MASTER PAGE, ENTER M.

EXHIBIT JOP550-11

TITLE: Master Display (MRG Procedure Guidelines)

MASTER PAGE   MRG PROCEDURE GUIDELINES

The functions performed by the MRG are dependent on the specifics contained in the PFF. The planner need only specify those parameters essential to satisfy his MRG requirements. For example, if resupply to off-shore countries is not required, do not define off-shore countries in the PFF. The minimum PFF specifics needed to execute the MRG are:

1. OPLAN END DAY AND SECURITY CLASSIFICATION (A10).
2. RESUPPLY INCREMENTS FOR ONE TIME PERIOD (A11).
3. ONE NON-POL SPLY CLASS/SUBCLASS CONSUMPTION FACTOR FOR EACH US SERV OR ONE POL CLASS/SUBCLASS CONSUMPTION FACTOR FOR EACH US SERV (A35).
4. ST-TO-MT CONVERSION MULTIP FOR EACH NON-POL CLASS/SUBCLASS (A70).
5. ONE NON-POL RESUPPLY CATEGORY DEFINITION OR ONE POL RESUPPLY CAT DEFINITION DEPENDING ON WHETHER THE CLASS/SUBCLASS IS NON-POL OR POL (A45).
6. PERCENT DISTRIBUTION AMONG POES FOR EACH RESPLY CAT AND SERV (A15).
7. ONE OBJECTIVE AREA COUNTRY DEFINITION. ONLY THE COUNTRY CODE IS MANDATORY (A60).

TO SELECT THE NEXT PASTER PAGE, ENTER M.

EXHIBIT 550-12

TITLE: PFF Entry Format "A35" Display

ENTER SUPPLY CLASS CONSUMPTION FACTORS.
SV-SERVICE CODE: CAN BE FOR A U.S. SERVICE OR A FWMAF SERVICE.
11-SUPPLY CLASS AND SUBCLASS CODES.
2-UNITS OF MEASURE: 1=LB/MAN/DAY: 2=GAL/MAN/DAY.
3333-INTEGRAL PART OF CONSUMPTION FACTOR.
44-FRACTIONAL PART OF CONSUMPTION FACTOR.

A. TO ADD OR CHANGE A CONSUMPTION FACTOR, ENTER SUPPLY CLASS AND SUBCLASS CODES, UNITS OF MEASURE, AND THE DESIRED FACTOR.
B. TO DELETE AN EXISTING CONSUMPTION FACTOR, ENTER THE SUPPLY CLASS AND SUBCLASS CODES: LEAVE THE UNITS OF MEASURE AND THE FACTOR BLANK.
C. TO DELETE ALL EXISTING CONSUMPTION FACTORS FOR A SERVICE, JUST ENTER THE SERVICE CODE.
D. ASTERISKS INDICATE REQUIRED DATA:REPEATED ENTRIES ALLOWED: THE PFF CAN CONTAIN UP TO 34 CONSUMPTION FACTORS PER SERVICE.
E. ENTER A BLANK LINE FOR RETURN TO MASTER SET.

```
A35 V   FACTOR   FACTOR   FACTOR   FACTOR   FACTOR   FACTOR   FACTOR
*** *112333344112333344112333344112333344112333344112333344112333344
```

EXHIBIT JOP550-13

TITLE: Simulation of MRG Module M10

You've established a PFF with M60, which will be used by M20 to generate the nonunit-related requirements. But, you don't have any data on this file yet. Therefore, you are required to construct some planning factors on the PFF. You are required to simulate only three formats. They are A10, All, and A35, in that order. The following specifications are provided:

A. Obtain a copy of all reports; and direct all output to Remote "AB".

B. Using format A10, construct the following miscellaneous planning factors on the PFF:
   1. The OPLAN End Day is 320 days after D-Day; the Plan ID is 4321A.
   2. Do not change the Theater Adjustment Factor; and do not enter any Attrition Factors.
   3. The OPLAN is unclassified.

C. Using format All, construct the Time Periods Specifications on the PFF, according to the following table:

Period	Start Day	Days Per Increment
1	0	3
2	30	2
3	60	5
4	(not required)	

D. Using format A35, construct one supply consumption factor on the PFF, for each U.S. Service. Specifications are as follows:
   1. Army: Subsistence Refrigerated consumption factor is 25.5 Lbs/Man/Day.
   2. Air Force: Ammunition/Ground consumption factor is 5.7 Lbs/Man/Day.
   3. Navy: Repair parts/Electronics consumption factor is 2.3 Lbs/Man/Day.
   4. Marine: POL/Ground consumption factor is 1.2 Gal/Man/Day.

EXHIBIT JOP550-14

TITLE: Simulation of MRG Module M20

You have established and constructed a PFF. Now you can generate the nonunit-related requirements for your OPLAN, because M20 needs the PFF to know which TPFDD items require nonunit-related requirements. You are required to simulate M20 three times to give you practice in the optional ways which you may want to execute M20. There may be times when you will only need the PWRS and/or TPFDD Summary Input reports, and there may be other times when you will only want to generate nonunit-related requirements with no need for any of these reports. Then, there may be times when you will want all the reports and generate nonunit-related requirements during the same execution.

The following simulations will give you practice in each of the ways described above. The following specifics are provided:

1. You must generate nonunit-related movement requirements for OPLAN 4321A. Obtain a copy of all reports and do not purge nonunit-related requirements that could have been generated in a previous MRG execution. (There are no nonunit-related requirements since this is the first MRG operation on the OPLAN.)

2. Generate only the PWRS report. Do not generate nonunit-related requirements.

3. Generate the nonunit-related requirements. Do not generate any reports (PWRS nor TPFDD Summary Input Report) during this run. Purge any nonunit-related requirements which were generated during the previous run (part 1). (Since your PFF did not change, the requirements you purge will be replaced by the same requirements.)

4. Terminate MRG processing at this time.

EXHIBIT JOP550-15

TITLE: Final Project for Lesson JOP550

You must satisfactorily complete this project prior to advancing to the next lesson. The situation that has been developing in the projects of previous lessons is continued. You are the planner responsible for creating the OPLAN, 1234Z, to defend country Zebra. To this point, you have created a TPFDD file, selected forces, and tailored the forces to your needs. Now in this lesson, you must use the MRG to generate movement requirements for nonunit-related cargo and personnel. After you complete the training of JOP550, and when instructed to start, you are to begin the simulations. All displays and files used in the simulations are unclassified. Execute the following three simulations in the order which they are listed.

Simulation 1:

Execute MRG Module M60 and input correct response(s) to establish a new Planning Factors File (PFF). Obtain a copy of all reports, and direct all output to Remote Print "AB".

Simulation 2:

Execute MRG Module M10 to construct the PFF (create planning factors on the PFF) which you established with M60. Print a copy of all reports, directing all output to Remote "AB". Follow the order listed below:

A. With A10, the following miscellaneous factors are to be constructed on the PFF.
   1. OPLAN ID, 1234Z, is unclassified and has an End Day of 60 days after D-Day.
   2. No changes are required to Theater ADJ nor Attrition Factors.
B. With A11, construct the following Time Periods Specifications on the PFF:
   1. Days Per Increment for Period 1 is 3.
   2. There are no other periods required.
C. With A35, construct one supply consumption factor on the PFF for the U.S. Army. The specifics are:

   Subsistence/Combat Rations will be consumed at a rate of 25.5 Lbs/Man/Day.

Simulation 3:

A. Call the proper T/I Module to sort the TPFDD in POD-LAD Sequence.
B. Execute MRG Module M20 to generate and add the nonunit-related movement requirements to the TPFDD file. Produce all reports. Purge from the TPFDD any nonunit-related movement requirements that may have been generated previously.
C. Call the proper FRG module to save your environment (TPFDD/SRF/PFF) to tape. Then terminate MRG processing with *FINISH.

EXHIBIT JOP550-16

TITLE: Input for T/I Simulations (Modules DØ5, DØ6, and DØ7)

Lesson JOP055 will teach you how to perform a preliminary transportation feasibility estimate using the utility modules of the T/I. During the lesson, three simulations will be presented. Following are the facts to be used throughout the simulations.

Simulation 1: DØ5

Module DØ5 operates in two modes: generate and update. You are to execute DØ5 first in the generate mode and then again in the update mode.

Generate Mode:

Generate the SR8/SR8A records for your TPFDD file. Use the following as criteria for executing DØ5 in the generate mode:

A. Enter the generate mode of DØ5.
B. Produce all reports on remote line printer "AB".
C. Return control to the System Monitor Level after completing DØ5.

Update Mode:

Update the SR8 records as described below:

A. Enter the update mode of DØ5.
B. Update geoloc code "LQCN" with the following:
   1. Set switch to place this geoloc in Tab A to the Transportation Report.
   2. Change daily cargo offload capability to '008000' S/T.
   3. Change road to '000900' S/T.
   4. Change rail to '001200' S/T.
   5. Change inland water to '005100' S/T.
C. Update "LQCN" with the above data.
D. Produce all reports on remote line printer "AB".
E. After completing DØ5, continue with module DØ6.

Simulation 2: DØ6

This module allows you to compute a Mobility Facility (GEOLOC) Passenger Cargo Offload Workload Summary. Use the following as criteria in executing this module:

A. Edit geolocs in the TPFDD against SR8/SR8A records.
B. Produce a CINC Mobility Facility Daily Workload Report and include the following:
   1. POE's
   2. POD's
   3. Intermediate Locations
C. Use default values for Sea and Air Closure Time Interval Factors.
D. Produce all output on remote line printer "AB".

EXHIBIT JOP555-1

TITLE: Input for T/I Simulation (Modules DØ5, DØ6, and DØ7)

Simulation 3: DØ7

Module DØ7 allows you to produce reports that aid the transportation planner in analyzing transportation workload requirements for selected mobility facilities by predefined OPLAN time periods. Use the following as criteria for executing DØ7:

A. Produce a Strategic Lift Requirement Report.
B. Produce a detailed Intra-Theater Lift Requirement Report and include the following:
   1. All POD's
   2. All Destinations
C. Produce a detailed Destination Reception Requirement Report and include the following:
   1. All final destinations
   2. TPFDD records for which final destinations are identical to the POD geoloc's.
D. Produce a detailed Lift Requirements Analysis Report as described below:
   1. ORGPOE (Origin to POE)
   2. Include all Origins
   3. Include all POE's.
E. Send all output to remote line printer "AB".

TITLE: Final Project for Lesson JOP555

You must complete the following three partial simulations in order to receive credit for lesson JOP555. After completing the training portion of the lesson and when directed by the coursewriter, enter the simulations and provide the responses according to the instructions below.

Let me refresh your memory on the status of your OPLAN (1234Z) to defend country Zebra. You have added, deleted and tailored the forces--recall the FRG in lessons JOP530 and JOP540. In the last lesson, JOP550, you developed the movement requirements for nonunit-related requirements. Now your OPLAN is complete and is ready to be tested for feasibility. You have two choices at this point in OPLAN development--first of all, you can go directly to the TFE for feasibility testing, or second, you can execute the Transportation Evaluation Modules of the T/I. In this course we suggest the latter as the preliminary method of feasibility evaluation, followed by the necessary adjustments, then TFE testing. All displays and data shown are unclassified and purely fictitious information. Specific instructions include:

Simulation 1:

1. Enter module DØ5 to accomplish the preprocessing (build the initial facilities used (SR8) records required for module DØ6. Do not change (update) the SR8 records on this run.
2. Print the "Facility Offload/Clearance Capabilities Planning Factors Report" on remote line printer 'AA'.
3. After accomplishing DØ5, enter module DØ6 automatically via this module.

Simulation 2:

1. Execute module DØ6 to produce a Mobility Facility Daily Workload Report.
2. The geoloc edit will be executed automatically.
3. Produce the CINC version of the report. Request data for all POE's, Intermediate Locations, POD's, and Destinations.
4. Use the default values for Air and Sea Closure Time Interval Factors.
5. Direct the output to remote line printer 'AA'.

Simulation 3:

1. Execute module DØ7 to produce the following two reports:
   A. Strategic Lift Requirements Report.
   B. Destination Reception Requirements Report--summarized version. (Include only the following geolocs in the report - ABCD, NBBB, and ZAAA.)
2. Direct output to remote line printer 'AA'.

EXHIBIT JOP555-2

TITLE : FACILITY OFFLOAD/CLEARANCE CAPABILITIES PLANNING FACTORS REPORT

CLASSIFICATION - UNCLASSIFIED JIS/D05 DATE - 01 JAN 1975

OPLAN - 1234Z **FACILITY OFFLOAD/CLEARANCE CAPABILITIES PLANNING FACTORS REPORT** PAGE NUMBER - 1

GEOLOC AND COUNTRY CODES	FACILITY NAME	FACILITY TYPE	FACILITY FUNCTION							DAILY OFFLOAD CAPAB. (S/T)	DAILY CLEARANCE CAPABILITIES TO OBJECTIVE AREAS (S/T)				FACILITY CAPAB. FACTOR SOURCE	TAB A
			ORG	INT	POE	INT	POD	INT	DES		AIR	ROAD	RAIL	INLAND WATER		
RCFD GE	MUNICH	CITY					X			999999	999999	999999	999999	999999	D05	YES
RJUW IT	NAPLES	SEAPORT			X		P									YES
RQNK 28	NEW ORL NS	SEAPORT	X	X	X			X		999999	999999	999999	999999	999999	PORTS	YES

EXHIBIT JOP555-3

```
TITLE:  CINC MOBILITY FACILITY WORKLOAD SUMMARY REPORT

                                          UNCLASSIFIED

TAB A TO APPENDIX A TO ANNEX D LOGISTICS
----------------------------------------

OPLAN 1234Z
-----------

MOBILITY SUP:  T FACILITIES
---------------------------

----- ------- ------- -------- -------- -------- -------- ------- ------- ------- ----------------------------------
 DAY   DAILY  AVERAGE    TOTAL   TOTAL    BULK   OVERSIZE OUTSIZE NON-AIR NUMBER  DAILY CLEARANCE CAPABILITIES      N
      OFFLOAD  DAILY  OFFLOAD    CARGO    CARGO    CARGO   CARGO   TRANS.          TO OBJECTIVE AREAS  (S/T)       O
       CAPAB.  UTIL.  CAPAB.                                       CARGO    OF                        INLAND       T
(D+ )  (S/T)   (S/T)   (S/T)     (S/T)    (S/T)    (S/T)   (S/T)   (S/T)    PAX    AIR   ROAD   RAIL  WATER  TOTAL  E
----- ------- ------- -------- -------- -------- -------- ------- ------- ------- ------*------*------*------*------ --
(AIRPORTS AND SEAPORTS)

(FIVE DAY INTERVAL FOR FIRST 30 DAYS:  TEN DAY INTERVALS THEREAFTER).

              ------------------------ --
              TOTALS  FOR THIS PLAN ARE
              ------------------------ --

------------
GRAND TOTALS                                  --------  --------  -----
------------                                  AIRPORTS  SEAPORTS  OTHER
                                              --------  --------  -----
  XX,XXX    -----NUMBER OF PASSENGERS-----     XX,XXX    XX,XXX   X,XXX

  XX,XXX    ------TOTAL BULK CARGO--------      X,XXX    XX,XXX     XXX
  XX,XXX    ------TOTAL OVERSIZE CARGO----      X,XXX     X,XXX       X
   X,XXX    ------TOTAL OUTSIZE CARGO-----      X,XXX     X,XXX       X
  XX,XXX    ------TOTAL NON-AIR TRANS-----      X,XXX     X,XXX       X
 XXX,XXX    ---------TOTAL CARGO----------     XX,XXX    XX,XXX   X,XXX


                                          UNCLASSIFIED
```

EXHIBIT JOP555-4

TITLE: Strategic Lift Requirement Report

```
                          UNCLASSIFIED

CINC GULF OPLAN 1234Z
---------------------
STRATEGIC LIFT REQUIREMENT REPORT
---------------------------------

---------- ----- ------- ------- -----------------------------------
 CLOSURE   TOTAL OUTSIZE NON-AIR TPFDD SEQUENCE NUMBER SUFFIXED BY
  PERIOD   CARGO  CARGO  TRANS.  - -CONTAIN OUTSIZE CARGO
    AT                    CARGO  * -CONTAIN NON-AIR TRANS CARGO
   POD     (S/T)  (S/T)   (S/T)  + -CONTAIN OUTSIZE AND NON-AIR TRANS
---------- ----- ------- ------- -----------------------------------

AIR TO POD
----------
   0000                          00004- 00043- 00079- 00027-
   0004  1341252  1983      0
   0005                          00006- 00054-
   0009    13372  1279      0
                          UNCLASSIFIED
```

EXHIBIT JOP555-5

UNCLASSIFIED

INTRA-THEATER LIFT REQUIREMENT SUMMARY REPORT

CLOSURE DAY AT DEST		TOTAL NUMBER OF PAX	TOTAL CARGO (S/T)	BULK CARGO (S/T)	OVER-SIZE CARGO (S/T)	OUTSIZE CARGO (S/T)	NON-AIR TRANS. CARGO (S/T)
	FROM BITBURG BSGX TO MUNICH RCFD						
D030 D039	PERIOD TOTALS	61	37420	96	37521	73	0
	FROM BRUSSELS COOH TO MUNICH RCFD						
D110 D119	PERIOD TOTALS	0	0	0	0	0	0
	FINAL TOTAL THIS DESTINATION	61	37420	96	37251	73	0

REPORT CONSISTS OF PAGES

CHANGE TO CINCGULF OPLAN 4321A

EXHIBIT JOP555-6

UNCLASSIFIED

DESTINATION DAILY RECEPTION REQUIREMENT REPORT

CLOSURE DAY AT DEST	SEQ. NO. IN TPFDD	FORCE REQ. NO.	UNIT DESCRIPTION	CC GEO. NO LOC. TD CODE YE POD	MODE OF TRANS TO DEST	TOTAL NUMBER OF PAX	TOTAL CARGO (S/T)	BULK CARGO (S/T)	OVER-SIZE CARGO (S/T)	OUTSIZE CARGO (S/T)	NON-AIR TRANS. CARGO (S/T)
			FINAL DEST -- RCFD -- MUNICH								
D035	00050	F2500	4TH POWERDIVE GROUP	GE-2SGK	AIR	61	37420	96	37251	73	0
D035					TOTAL	61	37420	96	37521	73	0
D030 D039			PERIOD TOTALS			61	37420	96	37521	73	0
D112	00023	A2C32	3RD GRAPESQUEEZER CO	BE-CQQH	AIR	0	0	0	0	0	0
D112					TOTAL	0	0	0	0	0	0
D110 D119			PERIOD TOTALS			0	0	0	0	0	0
			FINAL TOTAL THIS DESTINATION			61	37420	96	37251	73	0

REPORT CONSISTS OF PAGES.

CHANGE TO CINCGULF OPLAN 4321A

EXHIBIT JOP555-7

UNCLASSIFIED

```
   SUMMARY LIFT REQUIREMENT ANALYSIS REPORT
   ----------------------------------------

-------                                                     -------  -------  -------  -------  -------  -------
CLOSURE                                                      TOTAL    TOTAL    BULK    OVER-   OUTSIZE  NON-AIR
  DAY                                                       NUMBER    CARGO    CARGO    SIZE    CARGO    TRANS.
  AT                                                          OF                      CARGO             CARGO
 DEST                                                        PAX     (S/T)    (S/T)   (S/T)    (S/T)    (S/T)
-------                                                     -------  -------  -------  -------  -------  -------

   FROM BITBURG           BSGK  TO  MUNICH            RCFD
   ---------------------------------------------------------
D030                                                          61     37420      96    37251      73        0
D039               PERIOD TOTALS
   FROM BRUSSELS          COQH TO MUNICH              RCFD
   ---------------------------------------------------------
D110                                                           0         0       0        0       0        0
D119               PERIOD TOTALS
                   FINAL TOTAL THIS DESTINATION               61     37420      96    37251      73        0




   REPORT CONSISTS OF   PAGES.

   CHANGE    TO CINCGULF OPLAN 4321A
```

UNCLASSIFIED

EXHIBIT JOP555-8

TITLE: Input Information for TFE Simulations

Lesson JOP560 will teach you how to initialize the operating parameters of the simulation models. During the lesson, six simulations will be presented. Below are the facts and constraints to be used throughout the simulations.

Simulation T10A:

Module T10A reads the TPFDD File and extracts movement requirements and channel information. The information is written to the Movement Requirements Files and Channel Summary File. Two reports are produced which are merely listings of the two files produced. Specific information follows:

A. Print reports from the Remote Printer "AB".
B. Output is listed as unclassified.
C. Modify the Vessel Selection Matrix. On the first page, for Cargo Code B, change the priority 1 ship type from 7 to 6.
D. Make no changes to the Unit or Nonunit Cargo Code Matrices.
E. Create the movement requirements for scheduling.
F. Use the TPFDD File as input.

Simulation T10B:

Module T10B provides an update capability which allows the user to change or correct any data that was extracted from the TPFDD File during the last simulation (T10A). Your response of "Air" or "Sea" associates the correct file to be changed; for example, the "Air Movement Requirements File" or the "Sea Movement Requirements File". In this lesson we will only exercise the "Air" side of the module since the "Air" and "Sea" operations are parallel. Specific information follows:

A. Enter T10B and process "Air" moves.
B. Make the following "change" to the Air Movement Requirements File: For record sequence number 2 (FRN K2A). Change the number of passengers (PAX) from "230" to "144", and vehicle square feet (VEHSQFT) from '000000' to '003950'.
C. Use the same distance calculating option to generate distances for all channels - use the "MAC LEG CALCULATOR".
D. Do not add, change or delete GEOLOC Codes or Channels.

Simulation T11:

Module T11 processes the updated movement requirement records extracted and modified by T10A and T10B. Air and Sea Files are processed separately. First Geolocation (GEOLOC) Codes in the Movement Requirements File also exist in the Channel Summary File. Next in the cycle is to sort the Channel Summary File on POE and POD GEOLOC Codes. Then build a table of all possible channels between POE's and POD's. Finally the movement requirements are matched to the channels, and sequence numbers are assigned for control. There are two outputs: First the TFE Control File is updated to include movement requirement records and secondly a channel record from each POE code to each POD code is added to the Channel Summary File. Specific information for this simulation -- Process "Air" moves only.

Simulation T12:

Module T12, Route Planning, is used to develop the distances of each channel that was created in T11. After the distances have been generated, all of the channels will be displayed for the planner's review. If a distance figure must be changed for some reason, T12 provides the capability.

Specific instructions: Assume Module T11 processing is complete and you have entered "DSTAN". Execute Module T12 and review the distances that have been generated. Change distance from POE "ABCD" to POD "EFGH" from 1161 miles to 1234 miles.

Simulation T13:

Module T13, Port Retrieval, has two purposes. The first purpose is to retrieve airport or seaport statistics from JCS developed files. The second purpose is to provide the planner with the capability for altering the port statistics. The "Air" and "Sea" portions must be executed separately. Records generated by T13 are stored on the TFE Control File and are further developed by Modules T14 and T15 to be discussed later.

A. Enter Module T13 and process the "Air" Version.
B. Retrieve airport data from the standard JCS "APORTS" File.
C. Do not use the "maximum capacity" option for any of the airports. Use the capacity statistics contained in the JCS APORTS File.
D. Do not use the "bomb damage" assessment option. At this time it does not appear that country Bravo will strike the ports.
E. If any airport GEOLOC Code cannot be found in the APORTS File, generate the airport with the following constraints: From day 001, maximum cargo tonnage = 005280 tons, maximum sorties = 089, and maximum personnel throughput = 012345.
F. View data for all airports. For airport with GEOLOC "ADZM" (Port Delta), delete the constraints for day 003. Make no changes to the other ports.

Simulation T14:

Module T14, Aircraft Retrieval, allows the user to select, build and/or modify aircraft availabilities and characteristics. The first step in the execution to the module is to select aircraft type and availability data from the Transportation Assets. The next step is to retrieve aircraft characteristic data (speed, fuel, cargo canacity, etc.) from the Characteristics of Transportation Resourcus (CHSTR) File. The final step is to modify the type, availability or characteristic data as necessary for the applicable OPLAN. Specific insti ctions for the simulation:

A. Enter T14 and do not use data from the TFE Control File.
B. Enter "1" for CRAF Mobilization Group.
C. Select aircraft types from a list of "GULFCOM" Organic Aircraft - List Code Number 36.
D. For this OPLAN, use only C130 and C141 aircraft - delete all others.
E. Regarding the Allowable Cabin Load, input code "02" which means "GULFCOM during war conditions".
F. Change the aircraft characteristic record for C130's to reflect - cargo capacity in cubic feet to be 4800. Make no change to the C141 record.
G. Reflect the maximum number of hours C130 and C141 aircraft can fly as 20 hours per day for the first week, then beginning on day 8, decrease the maximum hours per day to 16.
H. Allow aircraft to fly into all ports - there are no restrictions.
I. Do not generate attrition. All planes are expected to reach their destinations safely.

NOTE: TFE Module T15, Ship/Landing Craft Retrieval, will not be simulated in JOP560 because its purpose and operation parallels Module T14, Aircraft Retrieval. The final project in Lesson JOP561 will allow you to simulate processing Module T15. Basically Module T15 selects and modifies Ships/Landing Craft and their characteristics in much the same way as Module T14 selects and modifies aircraft. Module T15 must be performed prior to executing the "Sea Scheduler" just as Module T14 must be performed prior to executing the "Air Scheduler".

TITLE: Vessel Selection Matrix

CARGO	TYPE SHIP SELECTION PRIORITY									
CODE	1	2	3	4	5	6	7	8	9	10
A	1	4	6	95	9	10	3	5		
B	2	95	4	6	3	5	7	14	10	9
C	1	4	6	95						
D	1	4	6	95	3	10	5	7	14	
E	1	95	4	6	12	7	14	3	10	5
F	1	95								
G	1	95	9	10						
H	1	95	9	10	6	3	5			
I	9	10	5							
J	11	8	5							
K	8	11	17	14	13	5				
L	4	6	1	95						
M	4	6	7	14	3	17	12			
N	12	17	14	7	10	3	6	9		
O	12	17	9	14	7	10	3	6		
P	15	16								
Q	6	7	14	3						
R	12	7	14	17	3					
S	18	19	20							
T	1	95	4	6						
U	3	5	7	14	16	9				
V	0	0								
W	0	0								
X	0	0								

NOTE: This matrix shows the priority that the simulation model will select ships to move different cargo categories (types of material and methods of loading). For example, to move Cargo Category A the model will first select ship type 1, then ship type 4, followed by types 6, 95, 9, and so on. Ship types and cargo categories from the matrix are shown on pages 2 and 3 of this exhibit.

EXHIBIT JOP560-2

SHIP TYPE LIST

(Sample data only for
Instruction Purposes)

Ship Type Number	Ship Type Description
1	LST (Conventional)
2	Well Deck Amphibious
3	BB Cargo
4	Well Deck Amphibious II
5	Passenger Ship
6	Other Amphibious Ships (Not Well Deck or LST)
7	LASH
8	Aircraft Ferry Ships
9	RO/RO
10	Heavy Lift
11	LPH
12	Container Ship (Self-Sustaining)
13	Container Ship (Nonself-Sustaining)
14	Sea Barge
15	Tankers
16	Oilers
17	Sea Train
18	Escorts DE
19	Escorts DD
20	Escorts DLG
95	LST (Gooseneck)
96	LST (unused)
97	LST (unused)
98	LST (unused)
99	LST (unused)

CARGO CATEGORIES

(Sample Data Only for
Instructional Purposes)

Code	Description
A	Unit Deployment, Tactical Loading
B	Unit Deployment, Admin Loading
C	Unit Deployment, Amphibious Assault
D	Unit Deployment, Amphibious Landing Follow On
E	Unit Deployment, Admin Follow On for Amphib Landing
F	Armor or Mech Force, Amphib Landing
G	Armor or Mech Force, Unit Deployment, Tactical Loading
H	Armor or Mech Force, Unit Deployment Admin Loading
I	Heavy Lift
J	Unit Deployment Significant Aircraft Lift
K	Unit Deployment, Predominantly Aircraft Lift
L	Accompanying Supply, Amphib Discharge
M	Accompanying Supply, General Discharge
N	Supply Buildup, General/Bulk Cargo
O	Supply Buildup, Ammunition
P	Bulk POL
Q	Resupply - RDD D + 5, Non-POL
R	Resupply - RDD D + 5, Non-POL
S	Special - Convoy
T	Organic Lift
U	Nonorganic Lift
V - Y	Reserved for future use

EXHIBIT JOP560-2 (Cont)

TITLE: Final Project for Lesson JOP560

This project is in lieu of a final exam in lesson JOP560. The project continues with the theme of previous final projects. You are preparing an OPLAN to defend country Zebra. In preceding projects, you converted an existing OPLAN to JOPS format, selected new forces, tailored forces, and generated both unit-related and nonunit-related movement requirements. You now have a complete OPLAN, but is it feasible? This project will set up the parameters of the simulation model. Primary emphasis will be placed on the Sea portion since JOP560 instruction taught the Air side. There are eight simulations which must be accomplished in sequence. All displays and files used throughout the simulations are unclassified. Use remote printer AB to produce all reports.

Simulation 1:

Load the TPFDD/SRF to disk from tape number 00333 (password = DELTA).

Simulation 2:

Enter responses necessary to allow TFE Module T10A to select (from the TPFDD file) the force requirements to be scheduled by the simulation models. Specific information follows:

A. Modify the Vessel Selection Matrix. For "Cargo Code (VS) B", change the priority one ship from "type 8" to "type 5". Do not change the other matrices.
B. Do not add alternate PODs.
C. Convoy POL shipping.
D. Produce all applicable reports using remote printer AB.

Simulation 3:

Use Module T10B to process and change the "Sea Movement Requirements File". Specific Instructions:

A. For the record with sequence number "7", change the POE from New York (GEOCODE = AAAA) to Charleston (GEOCODE = BBBB). Do not add, change, or delete other records.
B. Use the 'MSC Calculator" to determine distances.
C. Do not add, change or delete "channel" records.
D. Print all applicable reports through remote printer AB.

Simulation 4:

Execute Module T1 to generate "Sea Channel Records". Do not add channel records.

EXHIBIT JOP561-1

Simulation 5:

Using Module T12, review the "Sea Channel Records" generated by T11. Reflect an order from your commander (CINCBOLD) to avoid ocean area near country X-Ray by changing the total distance from POE "BBBB" to POD "CCCC" from 850 miles to 1000 nautical miles.

Simulation 6:

Retrieve seaport characteristic data from the PORTS File. Use port constraint data provided in the file; do not use unconstrained ports. Specific instructions follow:

A. *Review port characteristic data for all seaports. Do not change any data* except for Port Romeo (GEOCODE - GRRRR). For Romeo, add "day 005" to show daily capacity as "2500" measurement tons.
B. Do not add "Berth", "Beach", or "Landing Craft" data to those parts that do not already contain the data.
C. Bring "100" additional "type 35" landing craft to PODs by ship.

Simulation 7:

Retrieve ship type, availability and characteristic data from the Transportation Assets and the Characteristics of Strategic Transportation Resources Files, using Module T15. Specific instructions:

A. Select ships from your commander's (CINCBOLD) authorizations on "List 55." Use only the ships shown on line numbers 4, 5, and 6; delete all others. Do not change the data shown for lines 4, 5, and 6.
B. Convoying specifics: Convoy ships on the round-trip voyage, using twenty-five (25) escort ships each way. The breakpoint speed which determines if a ship will convoy or sail independently is 20 knots.
C. Do not make changes to data in the "Master Display". (See exhibit JOP561-2 or the TFE Users Manual for the display headings.)
D. Make all ships available to all ports.
E. Use "Administrative" type loading.
F. Review all ship characteristics, but make only the following change: For ship type Ø5LSTØ125" change troop capacity from "500" to "750".
G. Do not enter additional data for landing craft.

Simulation 8:

Save the TFE Control File to tape number "00555", and terminate the JOPS System.

"MASTER RECORD"

NOTE: RECORD CONTAINS DATA WHICH WILL DETERMINE HOW LONG A SHIP WILL WAIT FOR A PREFERRED DISCHARGE LOCATION BEFORE SWITCHING TO ANOTHER LOCATION,I.E., FROM PIER TO STREAM OR FROM STREAM TO PIER FOR UNLOADING PURPOSES. IT ALSO CONTAINS DATA WHICH DETERMINES BEACH USAGE, BASED ON TIDAL RESTRICTION. SHIP CONVOYING DATA IS ALSO CONTAINED HERE.

```
              EACH DAY TIDAL
           CONDITIONS PERMIT USE
             OF BEACHES DURING           (HRS)   (HRS)
(HRS HRS)  TWO INTERVALS.  ENTER                  TIME
            HOURS EACH INTERVAL           MAX      TO
TIME TIME     BEGINS AND ENDS.            TIME  ------
TO   TO    -----------------------        TO       RE- PER
WAIT WAIT     LST      LAND CRAFT         WAIT A  D  A CNTG                (HRS)
FOR  IN-     BEACH       BEACH     CON-   TO   S  I  S DIST  BROKEN         TIME
PIER STRM   --------    --------   VOY    OB-  S  S  S ADD- STOWAGE    1ST  TO
BE-  BE-    INTERVAL    INTERVAL   SIZE   TAIN E  P  E  ON   FACTOR     DAY  WAIT
FORE FORE  ----------- ----------- ----   MAX  M  E  M  DUE   FOR     FOR  FOR
IN- SWITCH  1ST   2ND   1ST   2ND  M  M   CON- B  R  B  TO   PHASE   PHASE  SHIP
STRM TO    ----- ----- ----- ----- I  A   VOY  L  S  L  ZIG- -------    2   OF
UNLD PIER  BG EN BG EN BG EN BG EN N  X   SIZE E  E  E  ZAG   1    2   LOAD  PREF
---- ----  ----- ----- ----- ----- -- --   -- -- -- -- --- ---- ----   ---  ----
XX.X XX.X  XX XX XX XX XX XX XX XX XX XX   XX XX XX XX XXX X.XX X.XX   XXX  XXX
23.0 23.0  00 23 23 23 00 23 23 23                     000 0.00 0.00   364  000
MAKE DESIRED CHANGES, ADHERING TO FORMAT ABOVE.  IF NO CHANGES, ENTER -NO-
```

(FORMAT EXTRACTED FROM TFE USER's MANUAL)

EXHIBIT JOP561-2

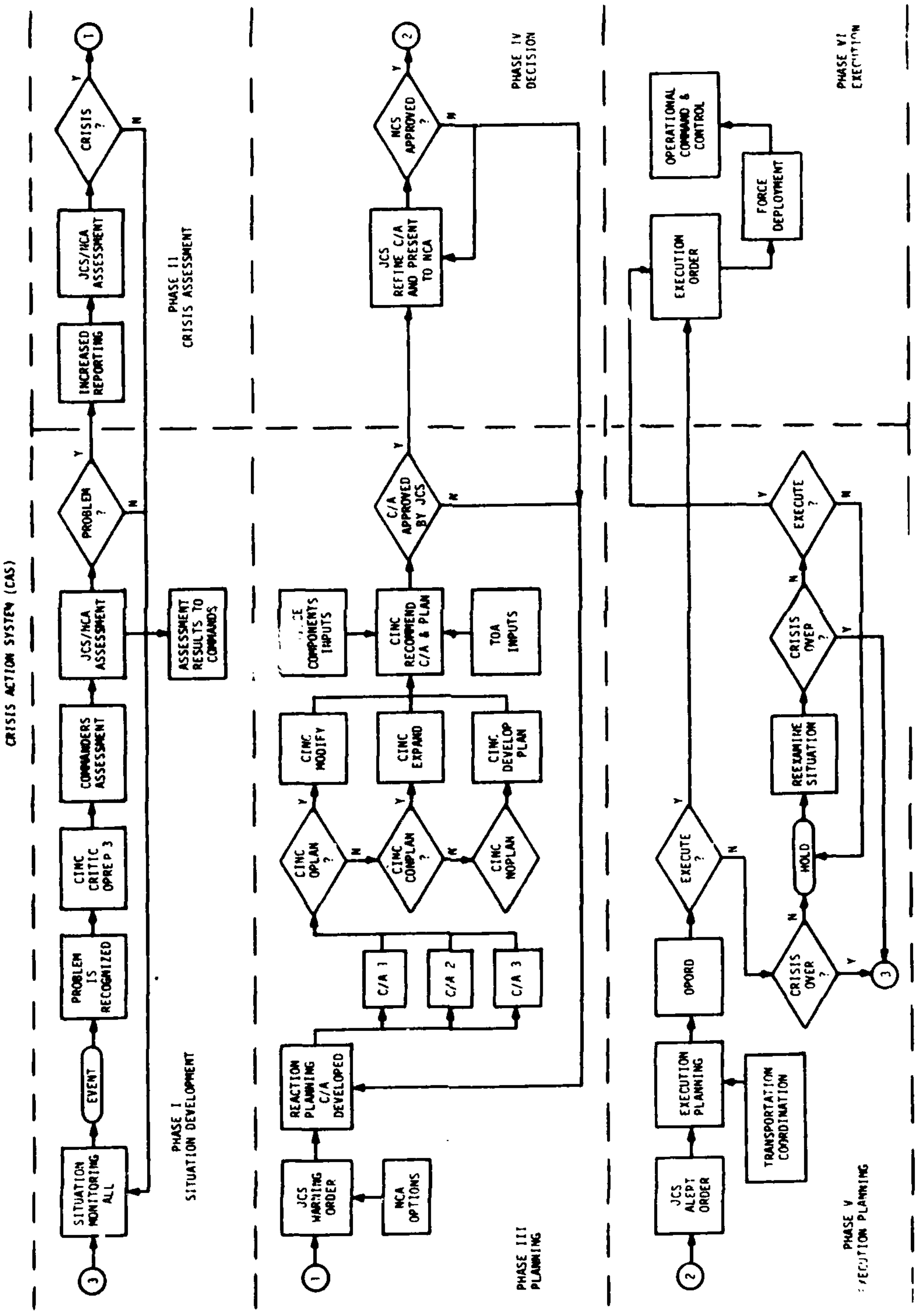


EXHIBIT JOP600-1

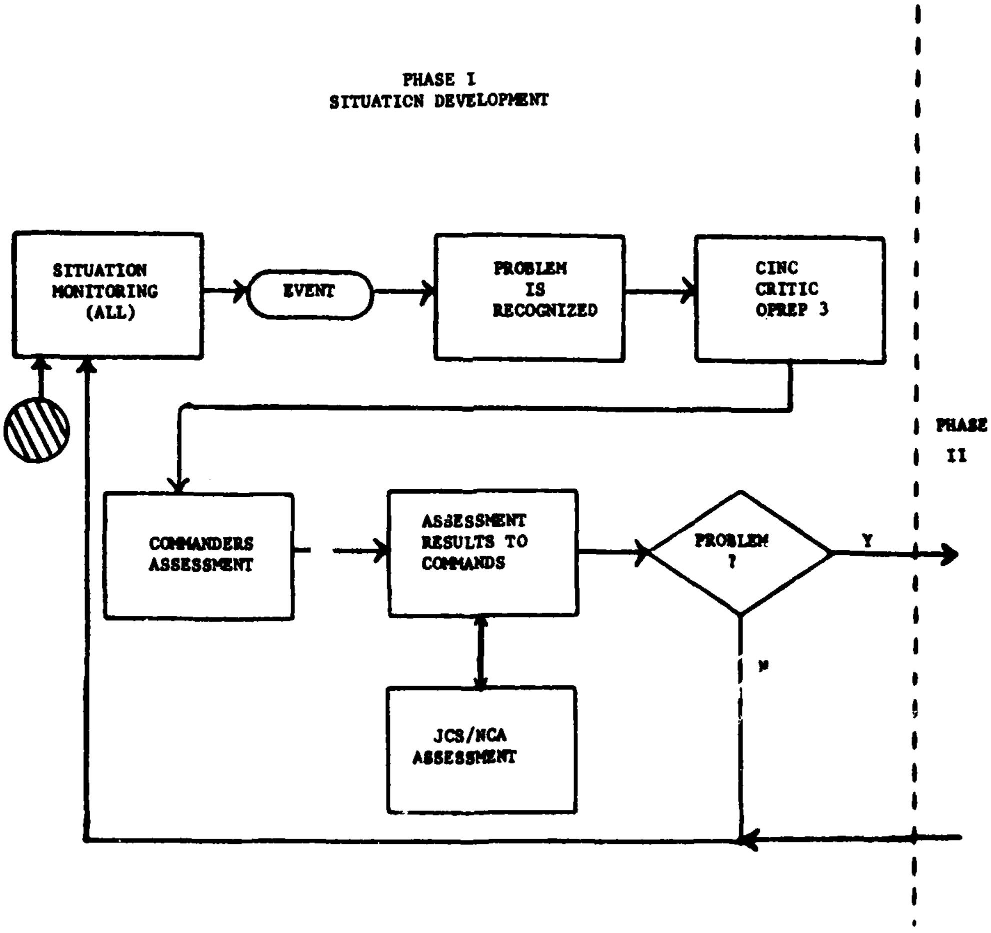


EXHIBIT JOP600-2

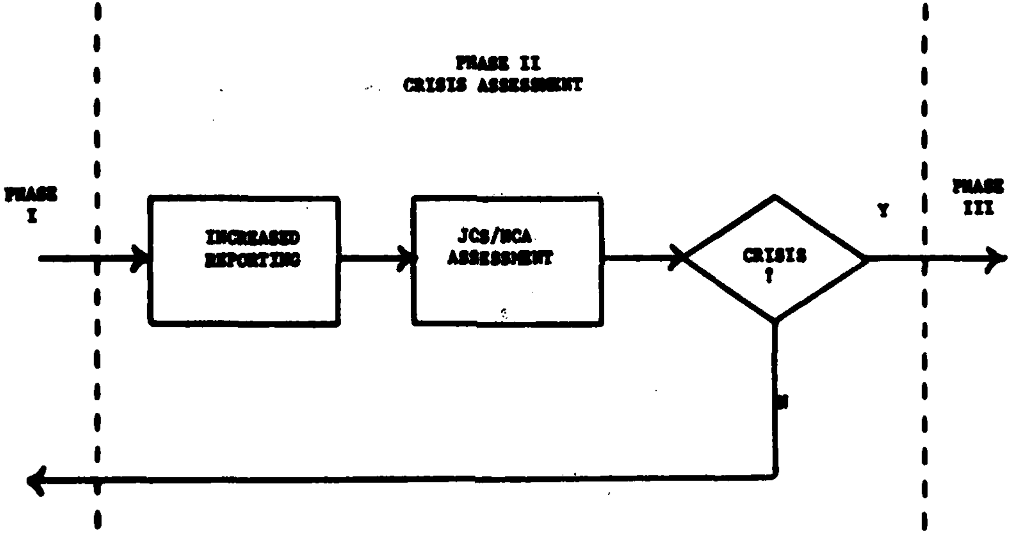


EXHIBIT JOP600-3

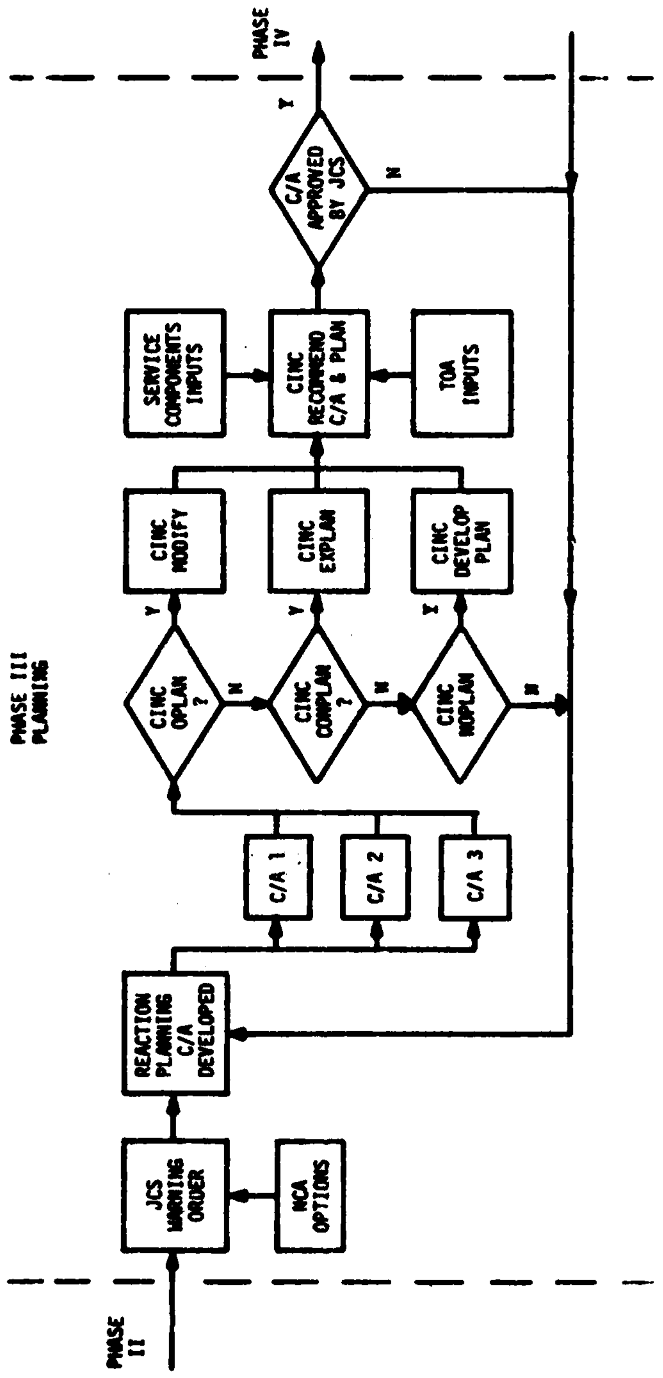


EXHIBIT JOP600-4

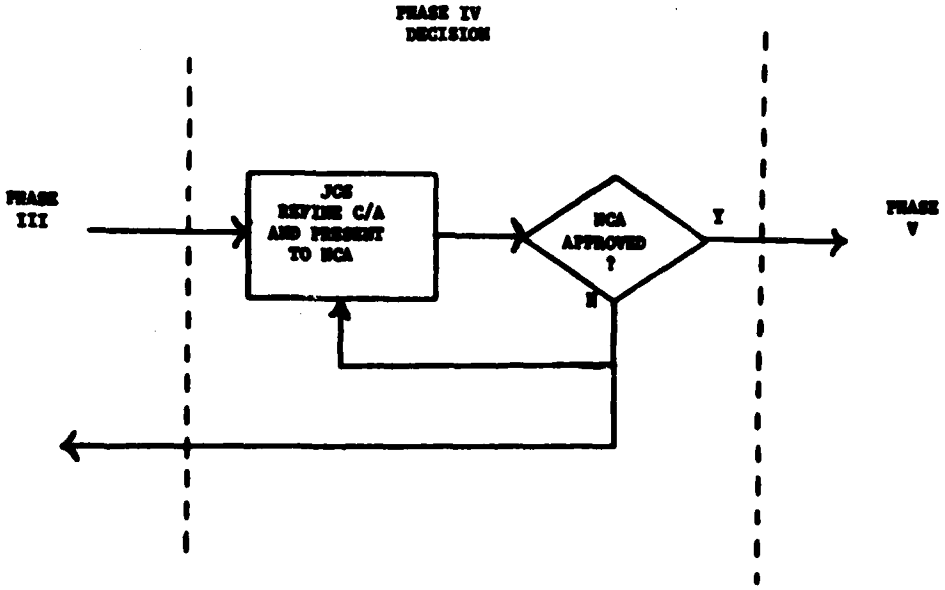


EXHIBIT JOP600-5

PHASE V
EXECUTION PLANNING

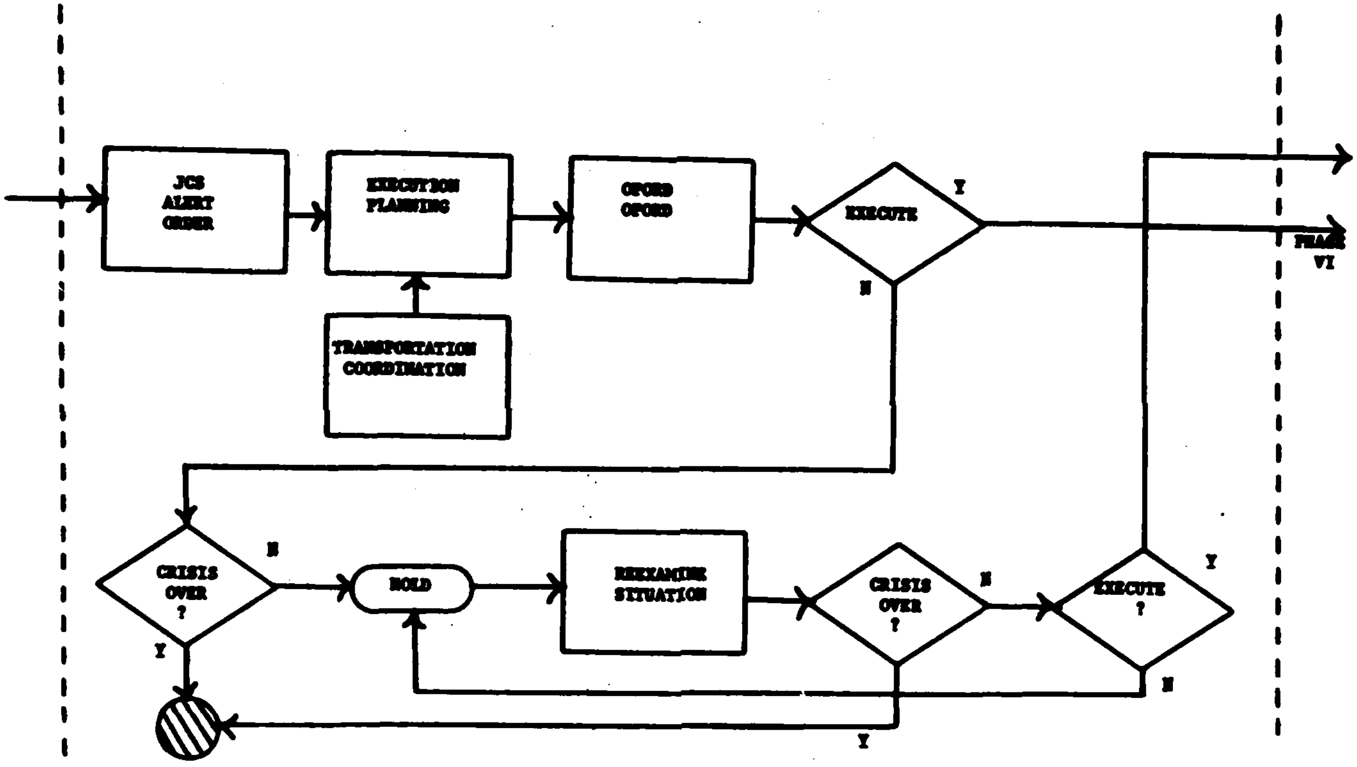


EXHIBIT JOP600-6

PHASE VI
EXECUTION

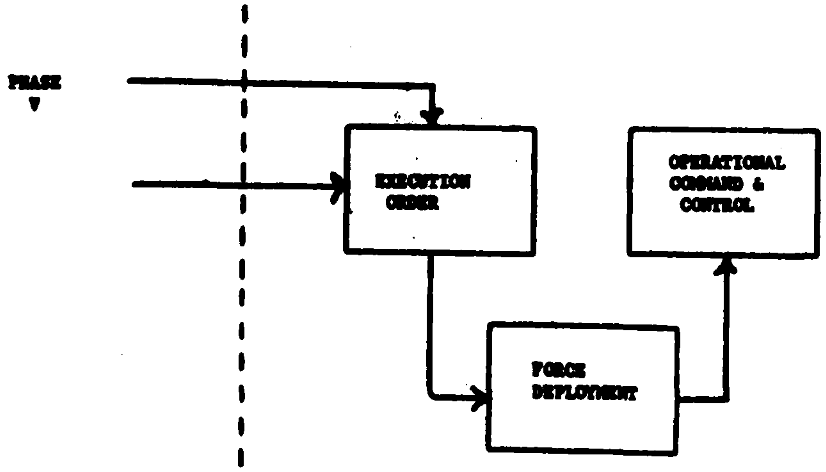


EXHIBIT JOP600-7

INTRODUCTION TO H6000 SERIES COMPUTER (H6000-CDT) COURSE EXHIBITS

OPERATING SYSTEM

1. Job Flow

   a. Input Phase

   b. Time-sharing

   c. Scheduler Phase

   d. Peripheral Allocation Phase

   e. Core Allocation Phase

   f. Execution Phase

   g. Termination Phase

   h. Output Phase

2. GCOS Composition

   a. Core Layout

   b. Slave Area

   c. System Library and User Program Area

3. Remote Processing

   a. Remote Batch

   b. Transaction Processing

   c. Time-sharing

EXHIBIT INT010-1

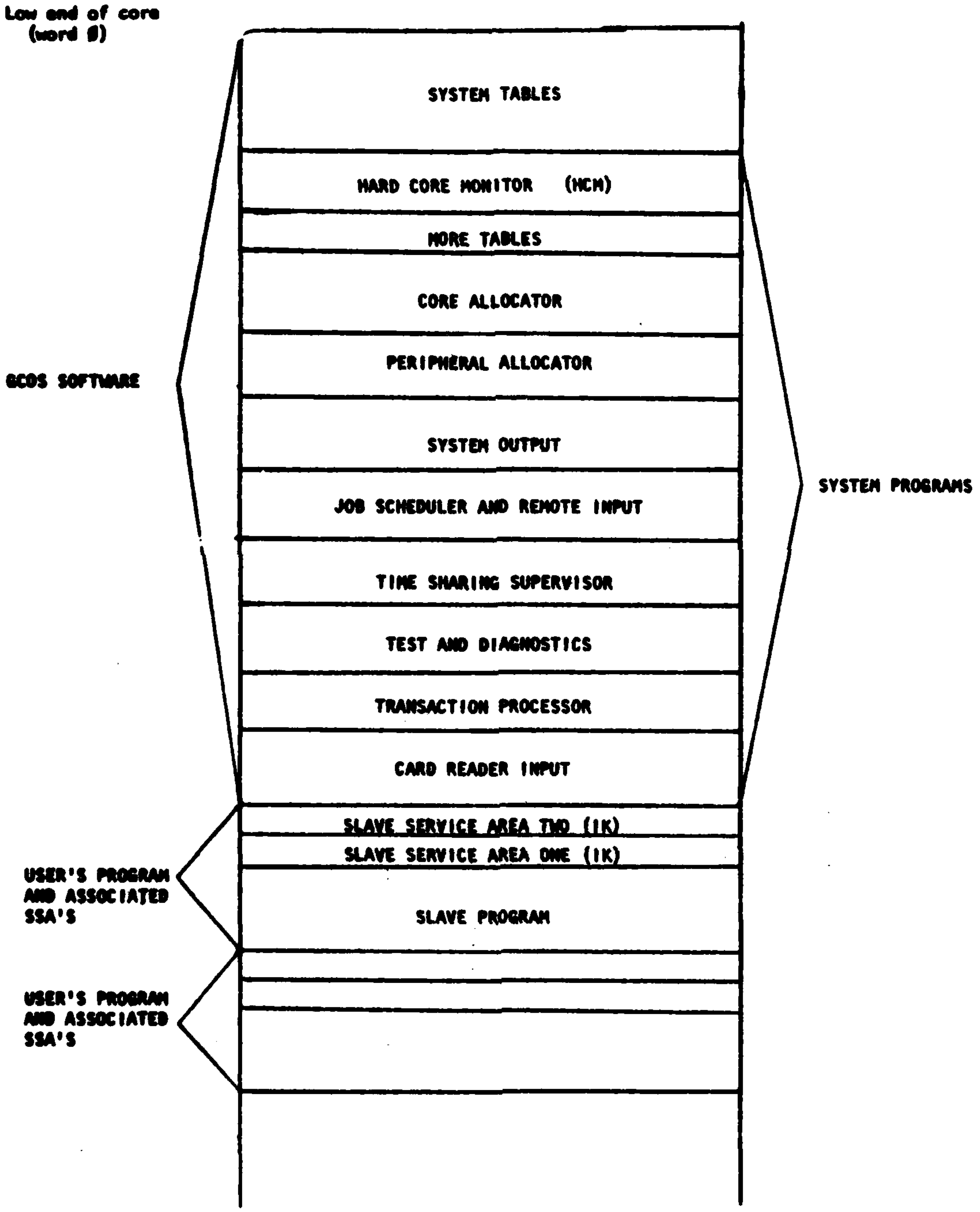


EXHIBIT INT010-2

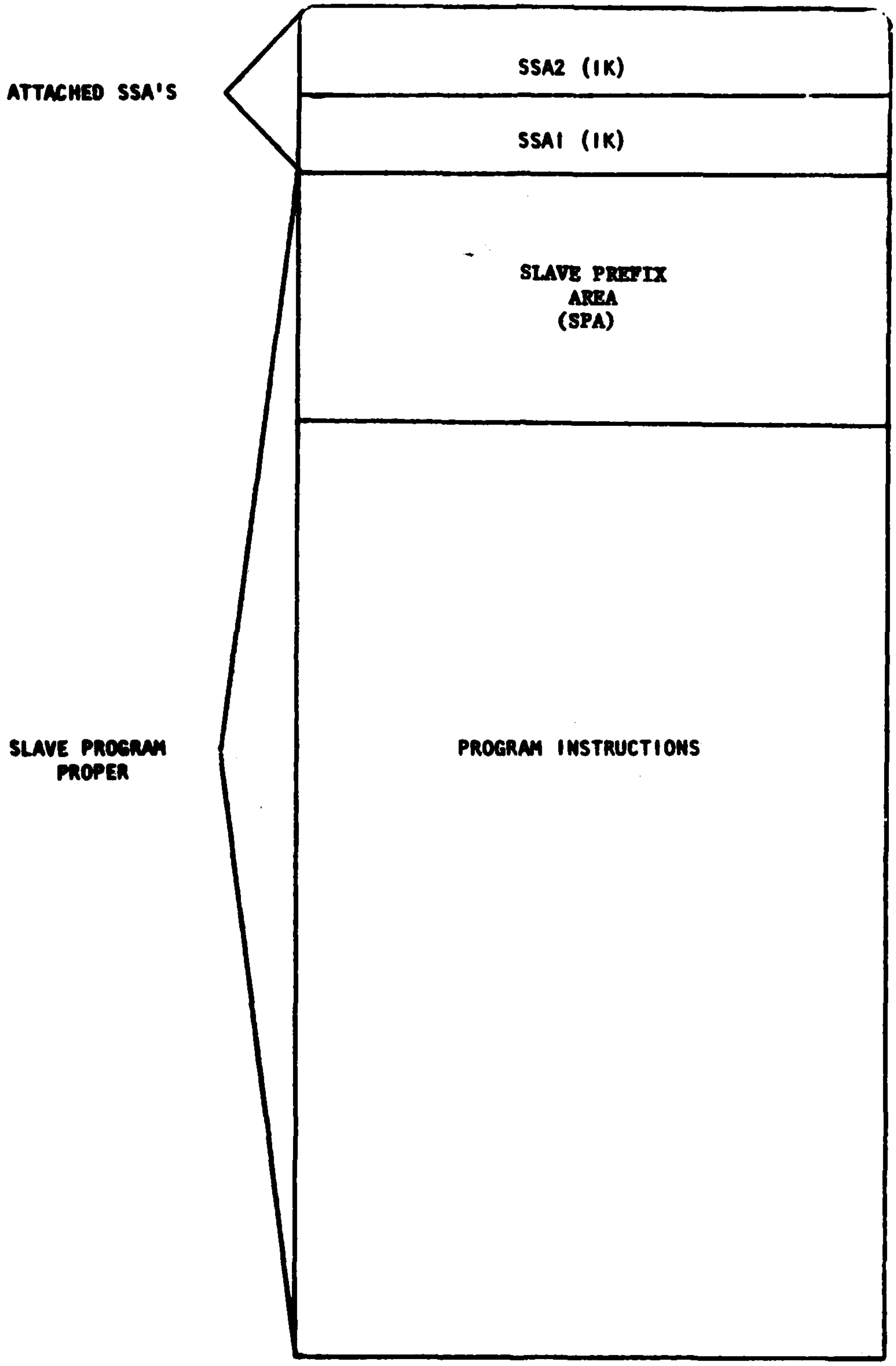


EXHIBIT INT010-3

SOFTWARE OVERVIEW

SECTION

1. Language Processors

   a. COBOL

   b. FORTRAN

   c. GMAP

   d. JOVIAL

   e. SIMSCRIPT

2. System Software

   a. File System

   b. Sort/Merge

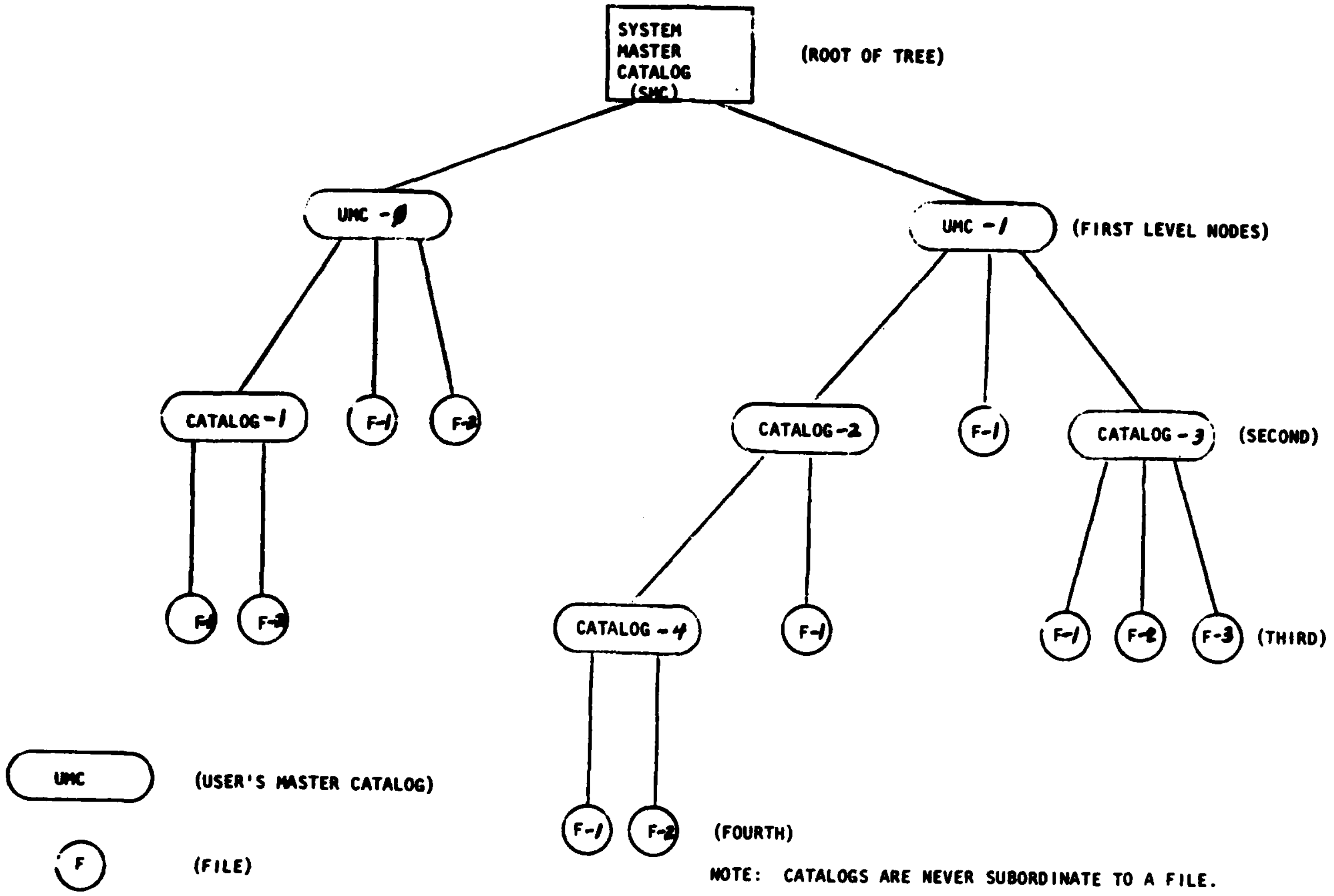
   c. Utility

   d. Bulk Media Conversion

   e. Library Editors

   f. Data Base Processing

EXHIBIT INT020-1



NOTE: CATALOGS ARE NEVER SUBORDINATE TO A FILE.

EXHIBIT INT020-2

JOB CONTROL LANGUAGE PART ONE

SECTION

1. Purpose of JCL
2. Job and Activity Definitions
3. General Format Rules
4. Job Set-up Explanation
5. Job Set-up Simulation

EXHIBIT INT030-1

BASIC COBOL JOB

```
              1      8       16
              ______________________________________
              $      SNUMB   ØØ345,5
              $      IDENT   6ZU P8ØØ 31   ,F222RXXXX
              $      USERID  WATC$STUDY/ZZZ
              $      COBOL   NDECK                  (ACTIVITY DEFINING CARD)
              $      LIMITS  2Ø,5ØK,,P5ØØØØ
              ______________________________
   activity   (Select Card-file, Assign to AA.)
     one
              (Select Print-file, Assign to PP.)
                                                    (COBOL SOURCE DECK)
              ______________________________
job
              ______________________________
              $      EXECUTE                        (ACTIVITY DEFINING CARD)
              $      DATA    AA
   activity   ______________________________
     two      ______________________________        (DATA DECK FOR COBOL PROGRAM)
              ______________________________
              $      SYSOUT  PP
              $      ENDJOB
              ***EOF
```

EXHIBIT INT030-2

SYSTEM FLOWCHART OF JOB ILLUSTRATED IN INT030-2

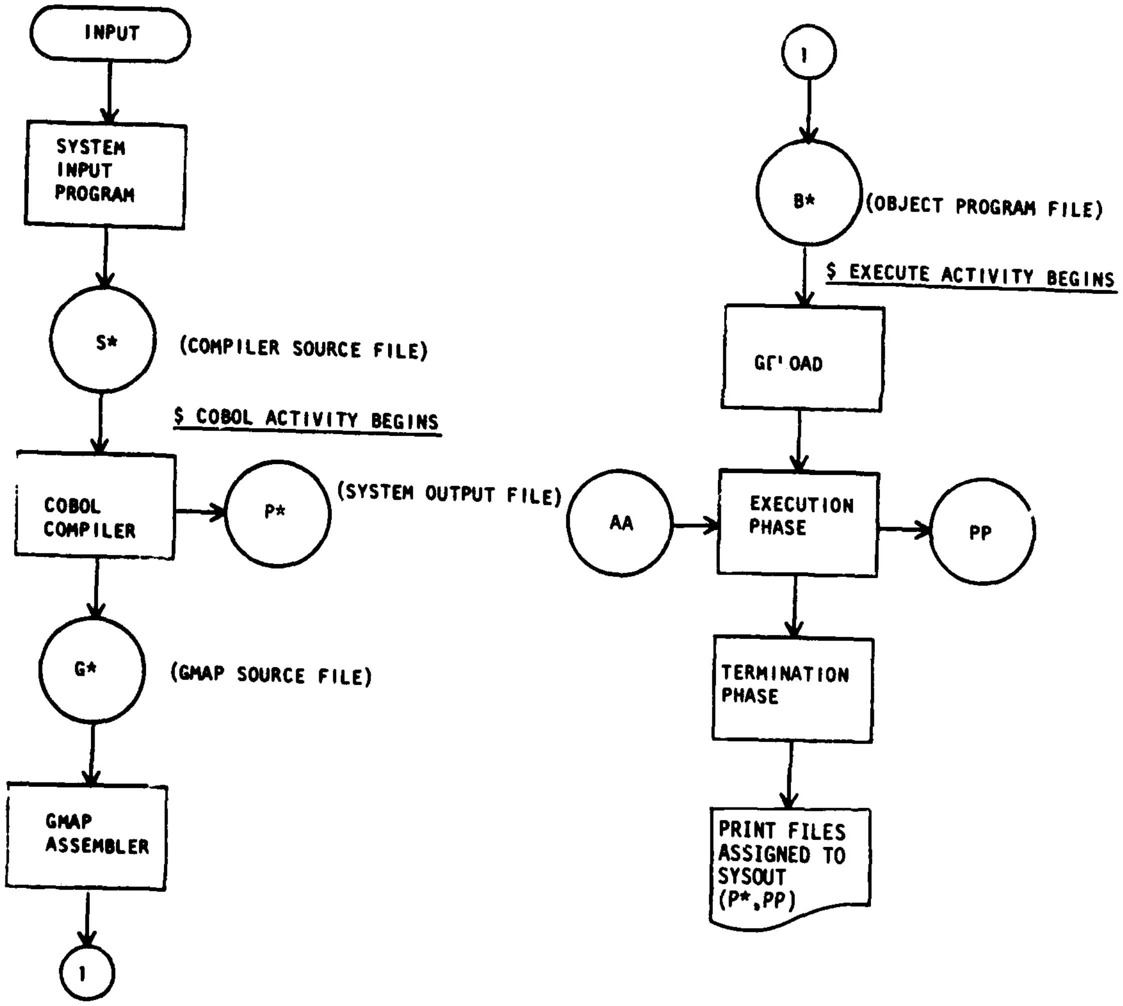


EXHIBIT INT030-3

JOB CONTROL LANGUAGE PART TWO

SECTION

1. EXHIBIT INT040-2
   a. $ DATA
   b. Logical Unit Designator
   c. $ FILE
2. EXHIBIT INT040-3
   a. $ PRMFL
   b. $ ETC
3. EXHIBIT INT040-4
   a. $ SELECT
   b. $ TAPE
   c. $ REMOTE
4. EXHIBIT INT040-5
   a. $ OBJECT
   b. $ DKEND
   c. $ OPTION
5. EXHIBIT INT040-6
   a. $ FORTRAN
   b. $ FFILE
6. Job Simulation
7. Execution Report

EXHIBIT INT040-1

COBOL job - emphasis on $ LIMITS, $ DATA, and $ FILE cards.

```
1      8       16
__________________

$      SNUMB   ØØØ346,5
$      IDENT   6ZU P8ØØ 31   ,F222RXXX
$      USERID  WATC$STUDY/ZZZ

$      COBOL   NDECK
$      LIMITS  2Ø,5ØK,,P5ØØØØ

__________________
Select card-file, assign to X1.
Select print-file, assign to AA.     (COBOL Source Deck One)

__________________
__________________
$      EXECUTE
$      DATA    X1S
__________________                   (Data Deck for COBOL Source Deck One)
__________________
__________________

$      COBOL   NDECK
__________________
Select card-file, assign to AX.
Select print-file, assign to AA.  (COBOL Source Deck Two)
__________________
__________________

$      EXECUTE
$      FILE    AX,X1R
$      SYSOUT  AA

$      ENDJOB

***EOF
```

EXHIBIT INT040-2

COBOL job - emphasis on $ PRMFL, $ ETC cards (two activities).

```
               1      8       16
               __________________

               $      SNUMB   ØØ347,5
               $      IDENT   6ZU P8ØØ 31   ,F222RXXX
               $      USERID  WATC$STUDY/ZZZ

               $      COBOL   NDECK
               ______________________________
               Select card-file, assign to L1.
1              Select print-file, assign to AA.     (COBOL SOURCE DECK)
               ______________________________
               ______________________________

               $      EXECUTE
               $      PRMFL   L1,R/W,S,WATC/STUDENT$FILES/
2              $      ETC     DATAFIL1
               $      SYSOUT AA

               $      ENDJOB
               ***EOF
```

FIG 1

File structure for $ PRMFL card illustrated above.

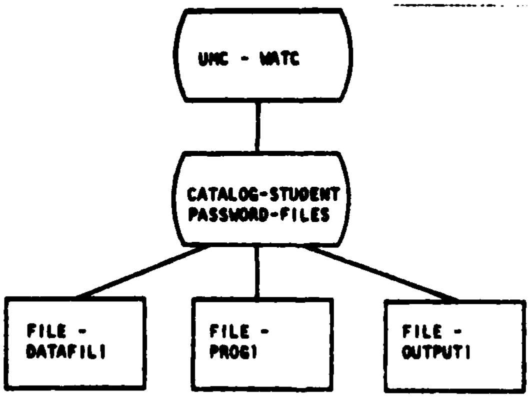


FIG 2

COBOL job - emphasis on $ SELECT, $ TAPE, and $ REMOTE cards (two activities).

```
                 1      8         16
                 __________________

                 $      SNUMB     ØØ348,5
                 $      IDENT     6ZU P8ØØ 31,F222RXXX
                 $      USERID    WATC$STUDY/ZZZ

1                $      SELECT    WATC/STUDENT$FILES/PROG1

                 $      TAPE9     B1,X1D,,ØØØ21,,DATA1
2                $      REMOTE    AA,AB

                 $      ENDJOB
                 ***EOF
```

FIG 1

Figure 2 reflects the card images stored on the selected file - PROG1. See EXHIBIT INT030-3, figure 3 for the file structure of PROG1.

```
                 1      8         16
                 __________________

                 $      COBOL     NDECK

                 ________________________
                 Select tape-file, assign to B1
1                Select print-file, assign to AA      (COBOL SOURCE PROGRAM CARD IMAGES)
                 ________________________

                 ________________________
                 $      EXECUTE
2                $      TAPE9     B1,X1D,,ØØØ22,,DATA2
```

NOTE: THE $ TAPE card in Figure 2 is replaced by the $ TAPE card in Figure 1.

FIG 2

EXHIBIT INT040-4

COBOL job - generates object deck (one activity).

```
1      8         16
----------------------

$      SNUMB     ØØ349,5
$      IDENT     6ZU P8ØØ 31   ,F222RXXX
$      USERID    WATC$STUDY/ZZZ

$      COBOL

-------------------------------------
Select tape-file, assign to A1
Select print-file, assign to AA        (COBOL SOURCE DECK)
-------------------------------------

-------------------------------------
$      ENDJOB
***EOF
```

FIG 1

COBOL job - runs object deck generated from figure 1 job.

```
1      8         16                           60                     80
-------------------------------------------------------------------------

$      SNUMB     ØØØ35Ø,5
$      IDENT     6ZU P8ØØ 31   ,F222RXXX
$      USERID    WATC$STUDY

$      OPTION    COBOL
$      OBJECT    NSEQ                         C16.687112574.......ØØ

----------------------
                           (COBOL OBJECT DECK)
----------------------

----------------------
$      DKEND

$      EXECUTE
$      DATA      A1

----------------------
                           (DATA DECK FOR COBOL OBJECT PROGRAM)
----------------------

----------------------
$      SYSOUT    AA

$      ENDJOB
***EOF
```

FIG 2

EXHIBIT INT040-5

FORTRAN job - emphasis on $ OPTION AND $ FFILE cards (two activities).

```
               1      8       16
               __________________

               $      SNUMB   ØØ35Ø,5
               $      IDENT   6ZU P8ØØ 31 ,F222RXXXX
               $      USERID  WATC$STUDY/DNM

               $      OPTION  FORTRAN
               $      FORTRAN NDECK
               __________________
1              Read (5,...)
               Write (6,...)  (FORTRAN SOURCE PROGRAM)
               __________________
               __________________

               $      EXECUTE
               $      FFILE   Ø6,NOSLEW
2              $      PRMFL   Ø5,R,S,WATC/STUDENT$FILES/DATAFIL2
               $      SYSOUT  Ø6

               $      ENDJOB
               ***EOF
```

FIG 1

File Structure for $ PRMFL card illustrated above.

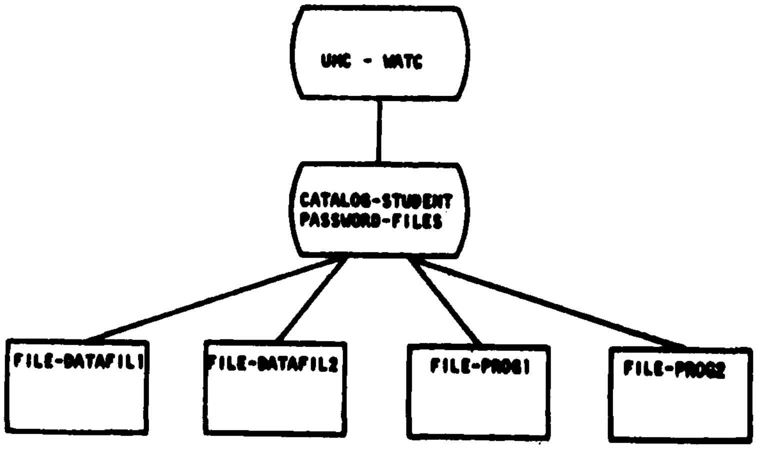


FIG 2

EXHIBIT INT040-6

General flowchart of FORTRAN job illustrated in exhibit INT030-6

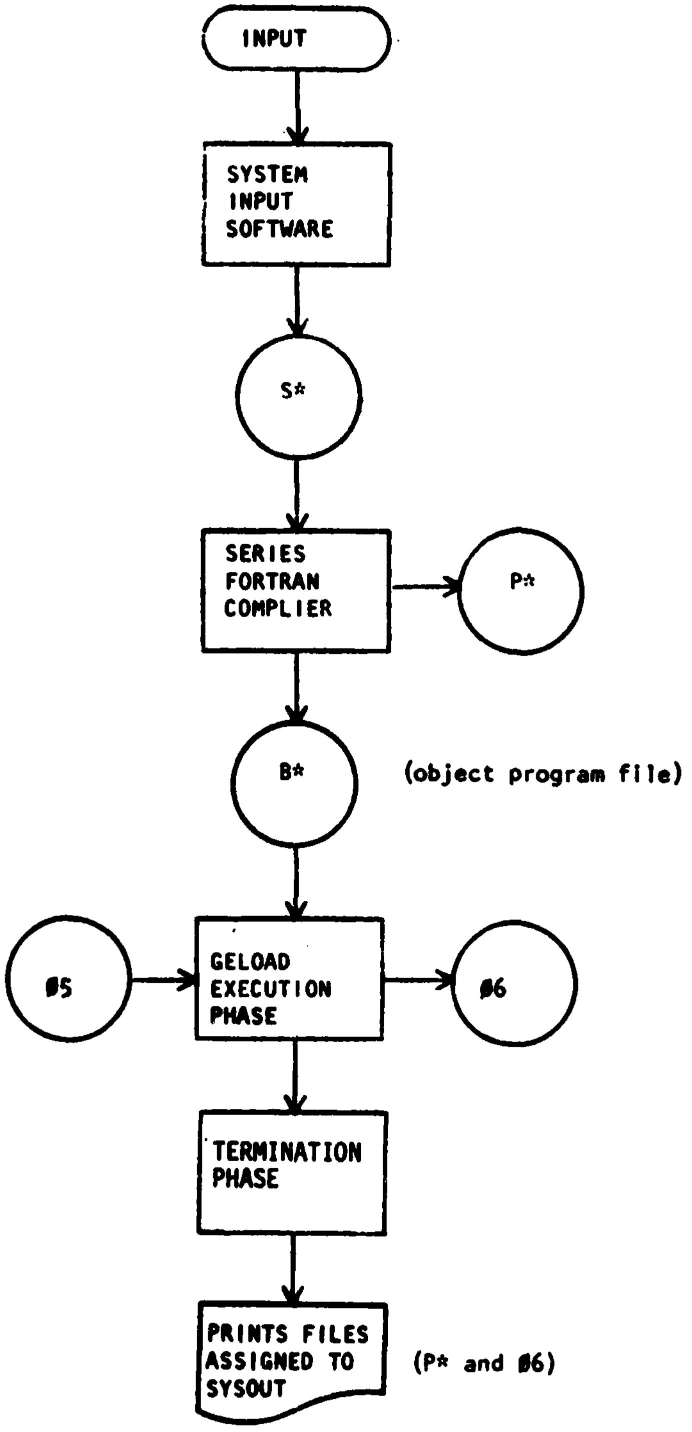


EXHIBIT INT040-7

JCL LESSON PROJECT

1. Upon completion of this project, have your training monitor check your work. Retain the listings from this project for an explanation of the execution report in lesson INT040.

2. Task I.

a. A COBOL source program without any control cards, and the data for the program are stored on two separate sequential files. The catalog/file strings for these files are: CDTSCDTS/INT/COBOL040 and CDTSCDTS/INT/DATA040. The input file code used in the COBOL program is AA and the output file code is PP.

b. Set up a job with the proper control cards that will cause the job to accomplish the following:

(1) Compile the source program.

(2) Execute the program.

(3) Printing a listing of the source program on the peripheral line printer.

(4) Print a listing of the assembled program.

(5) Print the results of the execution.

(6) Punch an object deck.

c. When you receive your punched object deck from Task I, you will have to strip (remove) the banner cards and any extraneous cards from the deck before you can put it in your Task II job (see exhibit INT040-9). Banner cards can be identified by the punches in columns 1 and 80 (punched in all rows except rows 11 and 8). When you have removed all extraneous cards, you will have the following cards remaining:

```
$       OBJECT
(Object deck cards)     (NOTE:  These cards are not interpreted.)
$       DKEND
```

Insert this combination of cards in your Task II job.

3. Task II.

a. The COBOL source program that was run in Task I computes the area and circumference of a circle when given the radius. The data must be formatted on an 80-column card image. Each radius must be placed on the card image in 5-digit increments starting in column one. Nonnumerics are not permitted, so filling the unused portion of the field with zeros may be necessary. Be aware that spaces and periods are nonnumeric and cannot be used. The assumed decimal point for the field is between the 4th and 5th digit of the field (e.g., 00555 would be treated as 55.5).

b. Keypunch one or two cards of data in the format specified above. (You do not have to completely fill the card; any number of fields will work.) Using the object deck you obtained from Task I and the data you have created, set up a job that will execute your data and print a listing of the results on the remote line printer. Ask your training monitor for the alphanumeric designator of the local remote line printer. If you don't have a local remote line printer configured that would be practical to use, output to the on-line printer.

EXHIBIT INT040-8 (Cont)

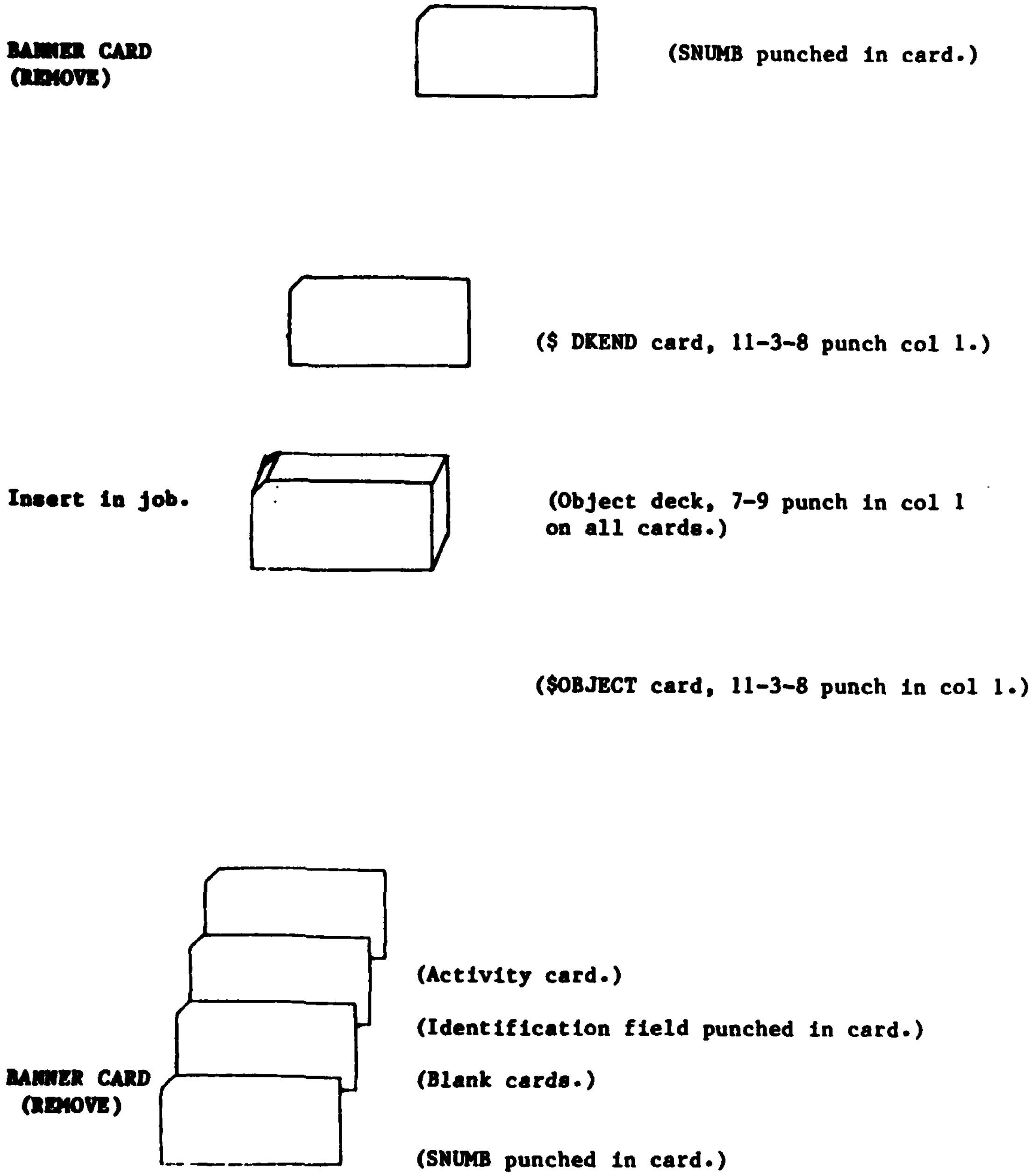
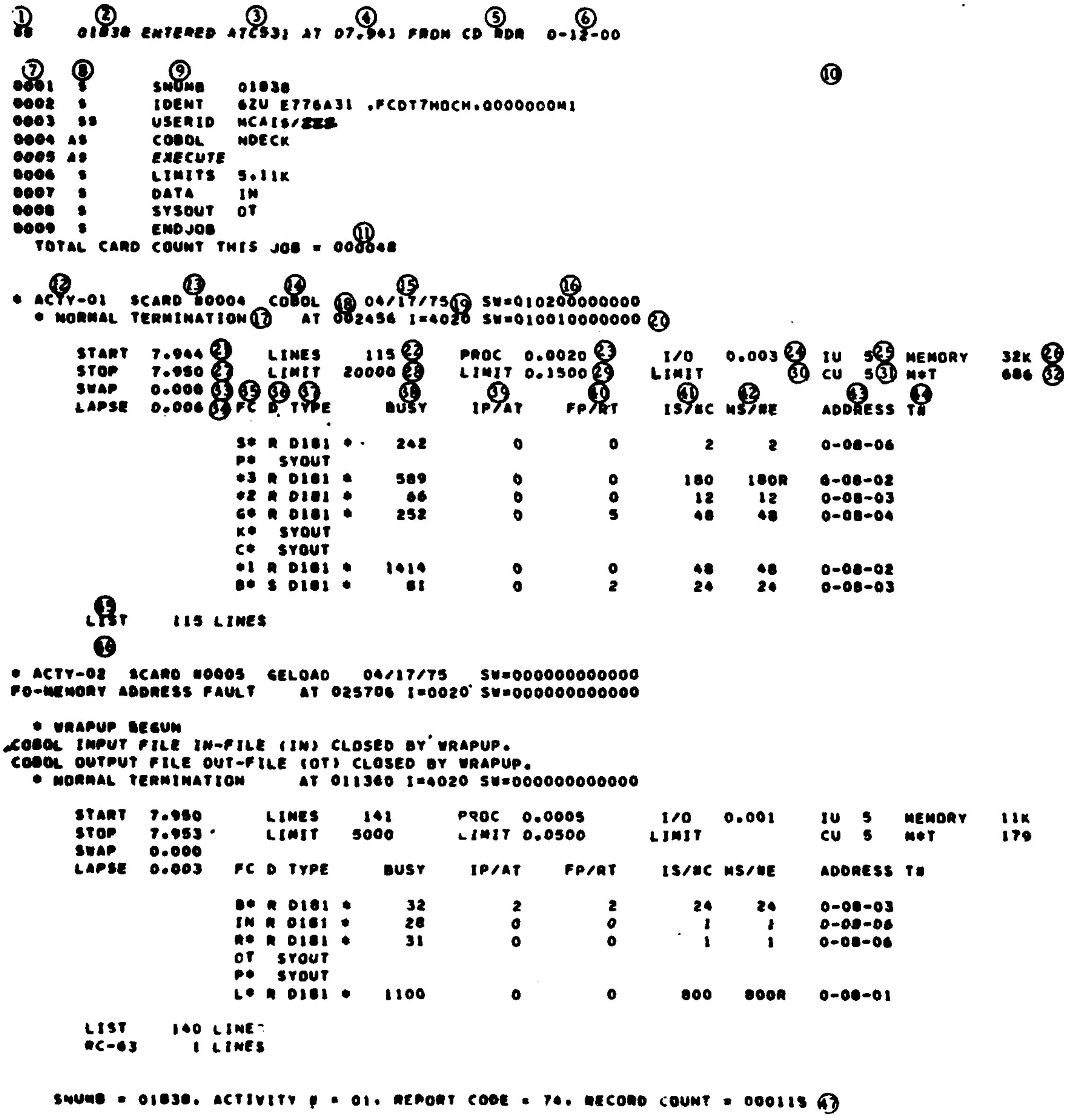

BANNER CARD
(REMOVE)
(SNUMB punched in card.)
($ DKEND card, 11-3-8 punch col 1.)
Insert in job.
(Object deck, 7-9 punch in col 1 on all cards.)
($OBJECT card, 11-3-8 punch in col 1.)
(Activity card.)
(Identification field punched in card.)
(Blank cards.)
BANNER CARD
(REMOVE)
(SNUMB punched in card.)

EXHIBIT INT040-9



```
$$     01838 ENTERED ATC531 AT 07.941 FROM CD RDR  0-12-00

0001  $        SNUMB    01838
0002  $        IDENT    6ZU E776A31 .FCDT7HDCH.0000000M1
0003  $$       USERID   NCAIS/[redacted]
0004 A$        COBOL    NDECK
0005 A$        EXECUTE
0006  $        LIMITS   5.11K
0007  $        DATA     IN
0008  $        SYSOUT   OT
0009  $        ENDJOB
  TOTAL CARD COUNT THIS JOB = 000048

* ACTY-01  $CARD #0004  COBOL   04/17/75  SW=010200000000
  * NORMAL TERMINATION   AT 002456 I=4020 SW=010010000000

     START  7.944     LINES    115     PROC  0.0020    I/O   0.003   IU  5  MEMORY   32K
     STOP   7.950     LIMIT  20000     LIMIT 0.1500    LIMIT           CU  5  M*T      686
     SWAP   0.000
     LAPSE  0.006  FC D TYPE      BUSY     IP/AT     FP/RT     IS/NC MS/NE   ADDRESS TN

                   S* R D181 *     242        0         0         2     2    0-08-06
                   P*  SYOUT
                   *3 R D181 *     589        0         0       180  180R    6-08-02
                   *2 R D181 *      66        0         0        12    12    0-08-03
                   G* R D181 *     252        0         5        48    48    0-08-04
                   K*  SYOUT
                   C*  SYOUT
                   *1 R D181 *    1414        0         0        48    48    0-08-02
                   B* S D181 *      81        0         2        24    24    0-08-03

      LIST    115 LINES

* ACTY-02  $CARD #0005  GELOAD   04/17/75   SW=000000000000
F0-MEMORY ADDRESS FAULT     AT 025706 I=0020 SW=000000000000

  * WRAPUP BEGUN
COBOL INPUT FILE IN-FILE (IN) CLOSED BY WRAPUP.
COBOL OUTPUT FILE OUT-FILE (OT) CLOSED BY WRAPUP.
  * NORMAL TERMINATION      AT 011360 I=4020 SW=000000000000

     START  7.950      LINES   141     PROC  0.0005    I/O   0.001   IU  5  MEMORY   11K
     STOP   7.953      LIMIT  5000     LIMIT 0.0500    LIMIT           CU  5  M*T      179
     SWAP   0.000
     LAPSE  0.003  FC D TYPE      BUSY     IP/AT     FP/RT     IS/NC MS/NE   ADDRESS TN

                   B* R D181 *      32        2         2        24    24    0-08-03
                   IN R D181 *      28        0         0         1     1    0-08-06
                   R* R D181 *      31        0         0         1     1    0-08-06
                   OT  SYOUT
                   P*  SYOUT
                   L* R D181 *    1100        0         0       800  800R    0-08-01

      LIST    140 LINE-
      RC-63     1 LINES

   SNUMB = 01838, ACTIVITY # = 01, REPORT CODE = 74, RECORD COUNT = 000115
```

EXHIBIT INT040-10

Definition of Fields in the Execution Report (see exhibit INT040-10)

INDEX DEFINITION

1 - Report code for the execution report. Indicates that the execution report will follow.

2 - SNUMB

3 - System identification contains six alphanumeric characters. Can contain the command and software version (e.g., ATC4ØØ). First digit = major release from Honeywell; second digit = subrelease; and third digit = WWMCCS change.

4 - Time in hours and thousandths of hours.

5 - Input source (card reader, tape, remote, etc).

6 - IOM, channel (PUB) number, and device number.

7 - Sequence number of control card images.

8 - A$ identifies the activity defining control cards, $$ identifies that the file system software had to be called to process the control card. An asterisk (*) following the sequence number indicates that the control card resided on a permanent file (e.g., *A$).

9 - Control card name.

10 - Columns 73 through 80 of the control cards.

11 - Total card count for this job.

12 - Activity number within the job.

13 - Sequence number of activity defining control card.

14 - Activity type (FORTRAN, GMAP, COBOL, GELOAD, etc.).

15 - Month/Day/Year.

16 - Status of the program switch word at the beginning of the activity. See GCOS manual for further explanation of the program switch word.

17 - Abort message texts and/or reason codes appear in place of the NORMAL TERMINATION portion of the termination message. These termination messages can be found in Appendix A of the GCOS manual.

18 - Memory location at activity termination.

19 - Contents of processor indicator register at termination of the activity. Format of the indicator register may be found in the GMAP manual.

EXHIBIT INT040-11

20 - Status of program switch word at the end of the activity.

21 - Time activity started in hours and thousandths of an hour.

22 - Total number of printed lines and cards punched for the activity (includes system output).

23 - Processor time consumed by activity in hours and ten thousandths of an hour.

24 - I/O time for this activity in hours and thousandths of an hour. (Does not include print or punch time.)

25 - Urgency level initially assigned to the activity.

26 - Core allocated to this activity (not including slave service area).

27 - Time activity stopped in hours and thousandths of an hour.

28 - Limit on number of lines of output for the activity. Does not include the number of lines output for the system.

29 - Limit on processor time for this activity in hours and ten thousandths.

30 - Limit on I/O time for the activity in hours and thousandths. If this limit is not specified on the $ LIMITS card, the I/O time is not limited and this field is blank on the execution report.

31 - Current urgency is the urgency attained by the activity before being allocated peripheral and core requirements. Urgency level is incremented regularly as an activity waits for core space; length of wait is directly related to amount of core required.

32 - Core usage is equal to the number of 1024-word blocks used multiplied by the time for which they were used. Expressed in block-seconds. Used for charging time and core to the user.

33 - Total time activity was swapped out of core storage.

34 - Total elapsed time for activity in hours and thousandths of an hour. Equal to STOP time minus START time.

35 - File codes used in the activity. An explanation of the system files can be found in the GCOS manual.

36 - Disposition Code:

R - Release
S - Save
D - Dismount
C - Continue
P - Release and purge

EXHIBIT INT040-11 (Cont)

37 - Media type according to the following list. A P or * to the right of the device type indicates a System Input created data file (*) or a cataloged permanent file (P).

Media Type	Device
D180	DSS180 Removable Disk
D181	DSS181 Removable Disk
D190	DSS190 Removable Disk
TAPE	7-track Magnetic Tape
TAP9	9-track Magnetic Tape
READ	Card Reader
PNCH	Card Punch
PRT3	PRT300 Printer
P203	PRT203 Printer
P303	PRT303 Printer
PRNT	PRT201 Printer
PPT	Paper Tape Reader
TYPE	Console Typewriter
SCC	System Control Center
CSS	Communication Subsystem

38 - Amount of time the peripheral device channel was used in the activity. If less than or equal to 999999, time is in milliseconds. If greater than 999999, time is in hours.

39 - Initial file position at start of activity. This field contains the alternate tape unit address when an alternate tape unit is specified on the $ TAPE card. Format is R/FF for tape files where R is number of records passed; F is number of files passed. If format is NNN, it represents the number of 320-word blocks passed for disk/drum files.

40 - Final file position at end of activity. Expressed same as 39.

41 - Initial size and mode of disk file (R=random, L=linked) or if tape file, number of times the file was accessed. Size is in 320-word blocks.

42 - Final size and mode of disk file or if tape file, number of errors encountered. Size is in 320-word blocks.

43 - IOM number/channel number/device number (or disk pack number).

44 - Link number (if mass storage) or tape serial number. If the device is a fixed disk pack, the initial link number will be shown in the last column under L#/T#. If a removable disk pack is specified, the disk number will be shown. (Later releases do not reflect link numbers.)

45 - Lines printed for the activity.
Compressed deck cards punched (COMDK).
Object cards punched (DECK).
May indicate a second report with a code of XX has been requested by a $ REPORT control card (RC-XX).

46 - One of the following message may appear in this space:

1. *** FILE XX IS DEFECTIVE *** (Part of file, number XX, could not be read because of defective space.)

2. Name of the activity deleted, if the activity was deleted.

47 - This line advises that a certain number of lines (record count) of output follow and that they are assigned a certain report code by the system software. For example, everything printed following the report code $$ (the execution report) was assigned the report code $$. A new report code designates that another report will follow.

EXHIBIT INT040-11 (Cont)

BULK MEDIA CONVERSION

SECTION

1. Bulk Media Conversion

   a. Basic BMC Job

      (1) $ CONVER

      (2) $ (FILE) Cards

   b. Functions

      (1) Files Card

      (2) $ FORM

      (3) $ MULTI

2. Nonstandard I/O Advanced BMC Training

   a. Block Size Options (BXXXX)

   b. Format Options

      (1) Low Density (LDENS)

      (2) BCD Mode (MBCD)

      (3) No Block Serial Number (NSER)

   c. Character Set

      (1) Transliteration (IBMF, IBMC, IBMEL)

      (2) Character Substitution (TAKEC)

   d. Record Form and Mode

      (1) Fixed Length (FXXX)

      (2) Variable Length

      (3) Binary Mode

      (4) Mixed Length (MIXL)

      (5) Input Media Conversion Tape (IMCV)

      (6) Ignore Media Code (NMEDIA)

   e. Printer Slew (SLEWX)

EXHIBIT INT050-1

f. Error Recovery

(1) Use or Skip Physical Record (UXXXX,SXXXX)

(2) Restart Run (XNNNN)

(3) Block Serial Error Alert (SERALT)

g. Limits

3. Summary

BMC OVERVIEW

```
             1        8          16
             ---------------------
             $        SNUMB
             $        IDENT      .......
             $        USERID     .......
             $        CONVER
             $        LIMITS     ,5K
             $        TAPE       IN,X1D,,888888,,TAPENAME
ACT 1        $        INPUT      B4ØØ,LDENS,NLABEL,NSER,IBMF,TAKE#
             $        INPUT      U1ØØ
             $        TAPE       OT,X2C,,777777,,NAMETAPE
             $        MULTI      1-2/5/9
             $        CONVER
             $        LIMITS     ,4K
ACT 2        $        TAPE       IN,X2C
             $        PUNCH      OT
             $        INPUT      NMEDIA
             $        CONVER
             $        LIMITS     ,4K
ACT 3        $        TAPE       IN,X2D
             $        PRMFL      OT,R/W,S,CATA/FILESTG
             $        ENDJOB
```

EXHIBIT INT050-2

UTILITY

SECTION

1. Basic Study Section

   a. Utility Control Cards

      (1) $ UTILITY

      (2) $ (FILES)

      (3) $ FUTIL

   b. Basic Utility Functions

      (1) COPY and MCOPY

      (2) COMP

      (3) SKIP

      (4) DUMP, DDUMP, ADUMP, and AADUMP

      (5) RWD or REW

2. Advanced Study Section

   a. Advanced Utility Functions

      (1) RPT

      (2) HOLD

   b. Physical Record Processing

      (1) $ FFILE

      (2) Utility Functions

         (a) BKSP

         (b) RSAVE, RREST, and RCOPY

   c. Utility Options ($ QUTIL)

      (1) EOF/ALL

      (2) ASIS

      (3) TERM

      (4) CMPERR/N

EXHIBIT INT060-1

(5) NBYPSS

(6) USE/N AND IGNORE/N

(7) RESEQ

(8) USER

d. $ ABORT

EXHIBIT INT060-1 (Cont)

INTEGRATED DATA STORE (IDS) (H6000-CDT) COURSE EXHIBITS

INTEGRATED DATA STORE

A SOFTWARE SYSTEM DESIGNED TO CREATE AND MAINTAIN A NETWORK DATA BASE, USING AN IMBEDDED LINK STRUCTURE.

IDS SOFTWARE SYSTEM

1. IDS LANGUAGE TRANSLATOR
2. DATA BASE MANAGER
3. UTILITY ROUTINES

EXHIBIT IDS010-1

CREATION AND MAINTENANCE

BY THE USER	BY SYSTEMS PERSONNEL
STORE DATA	DATA BASE ANALYSIS
MODIFY DATA	RECOVERY
RETRIEVE DATA	REORGANIZATION
DELETE DATA	

EXHIBIT IDS010-2

IMBEDDED LINK STRUCTURE

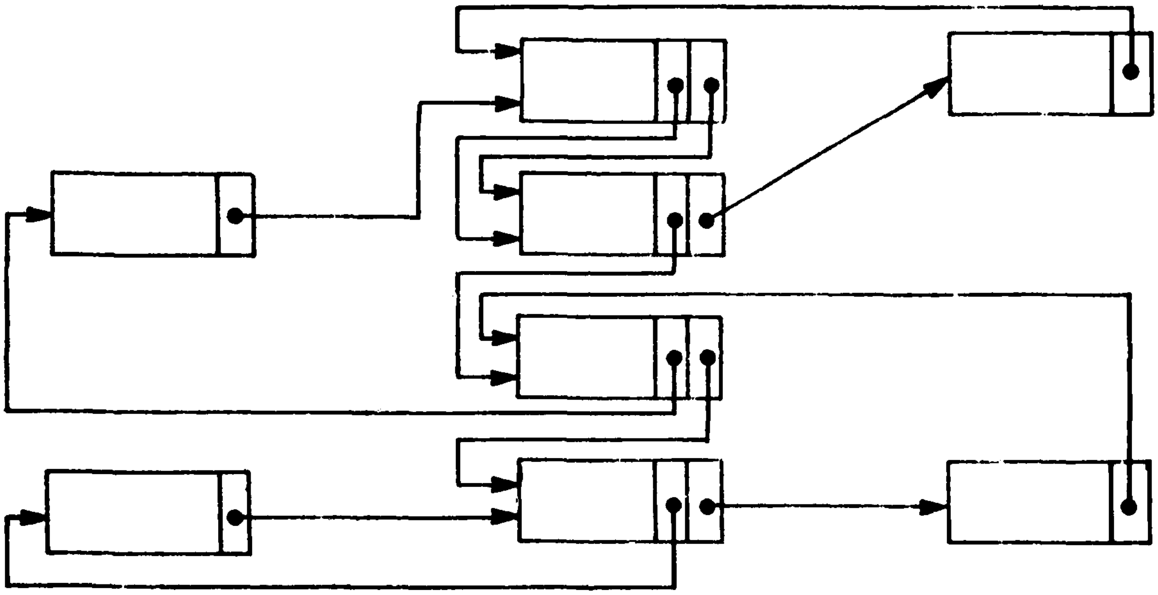


EXHIBIT IDS010-3

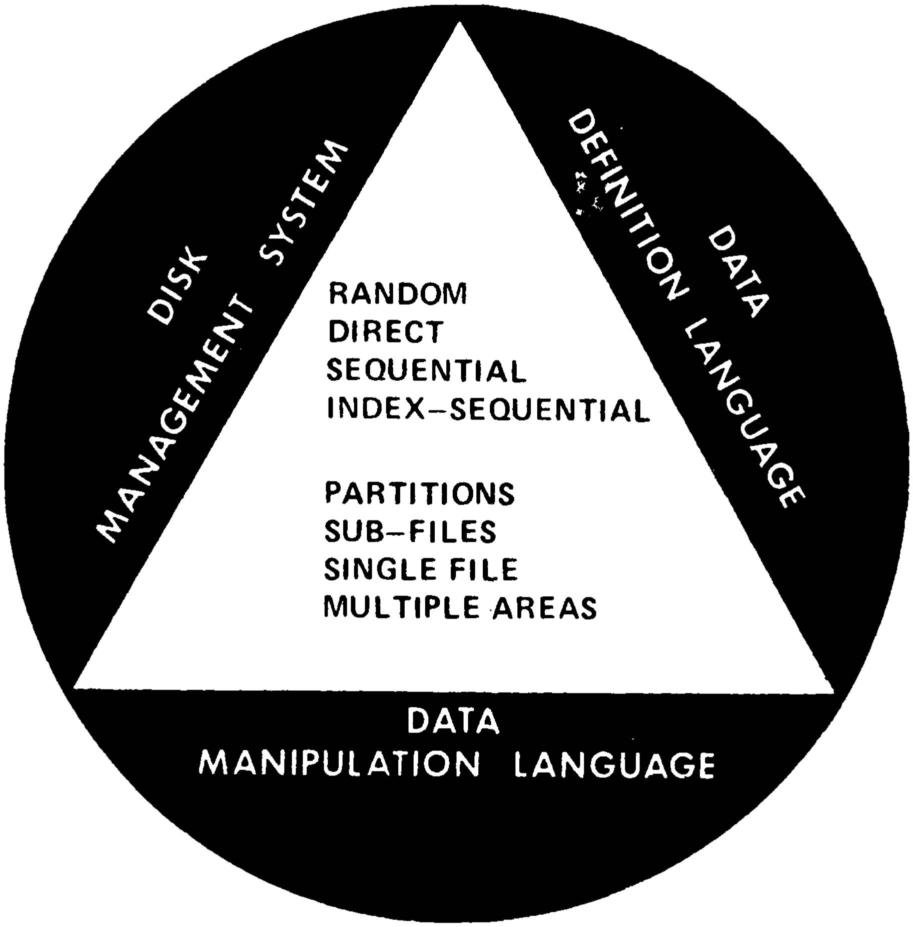


EXHIBIT IDS010-4

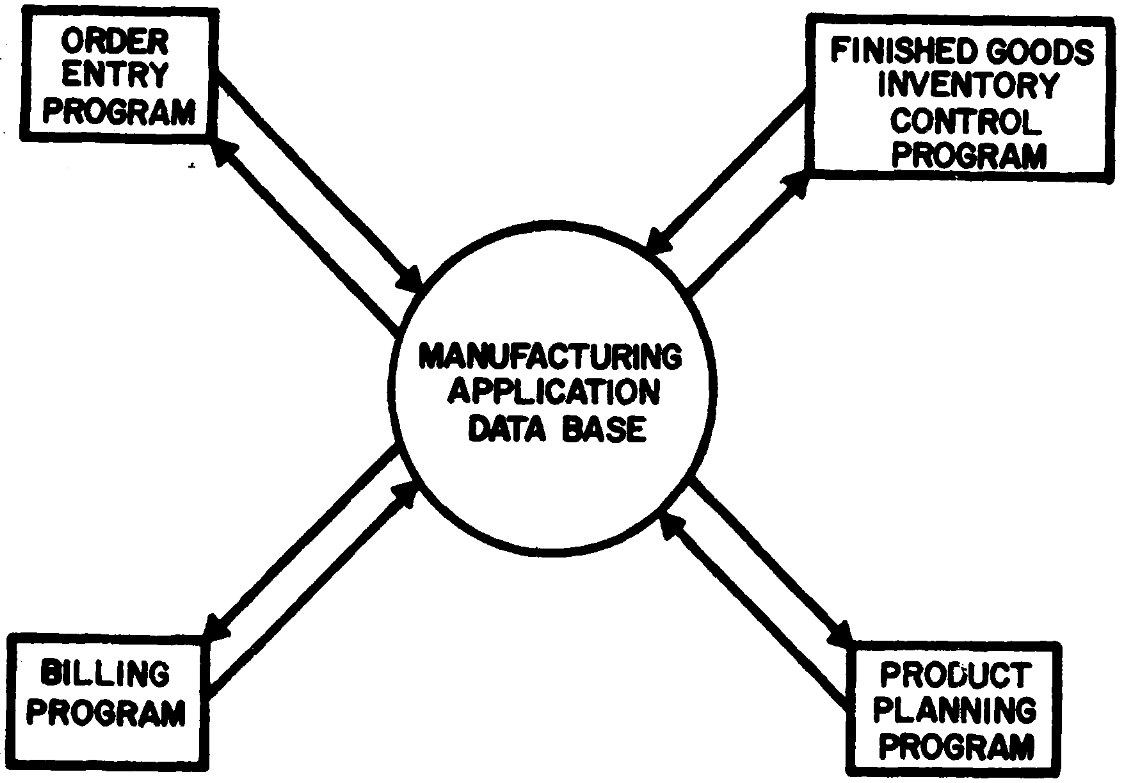

ORDER ENTRY PROGRAM
FINISHED GOODS INVENTORY CONTROL PROGRAM
MANUFACTURING APPLICATION DATA BASE
BILLING PROGRAM
PRODUCT PLANNING PROGRAM

EXHIBIT IDS010-5

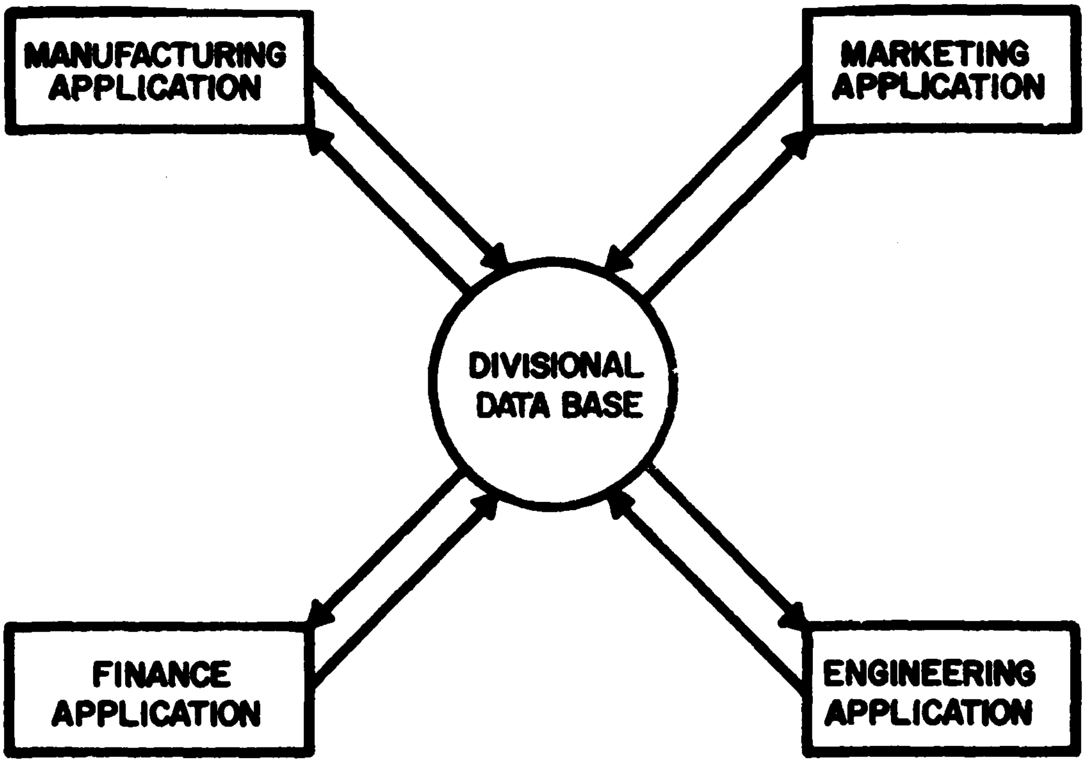


EXHIBIT IDS010-6

FEATURES

(1) Allows users to describe the data relationships that exist in the user's area of concern permitting diverse users to access common information.

(2) Can eliminate data redundancy.

(3) Provides for a variable number of entry points into the data base.

(4) Provides a simple, powerful, easy-to-use language.

(5) Has a graphic documentation technique.

(6) Supported by extensive utility routines.

(7) Permits executing more than one program at a time against one data base.

(8) Provides users with a constantly updated data base.

EXHIBIT IDS010-7

IDS/COBOL COMPILATION

CONTROL CARDS

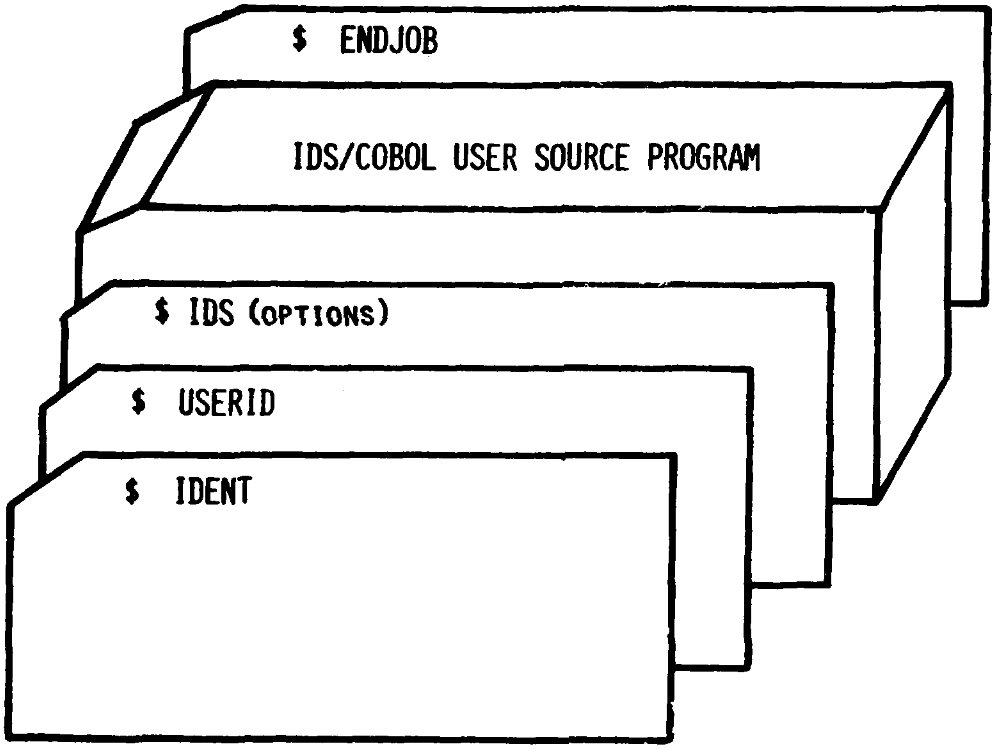


EXHIBIT IDS010-8

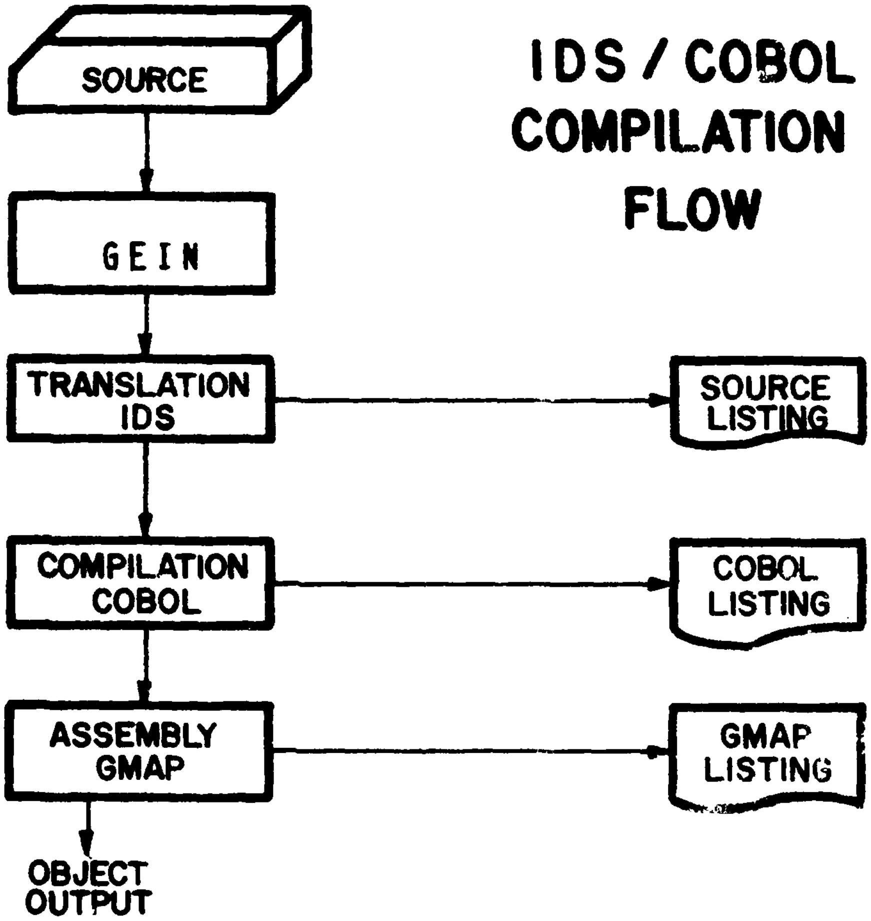
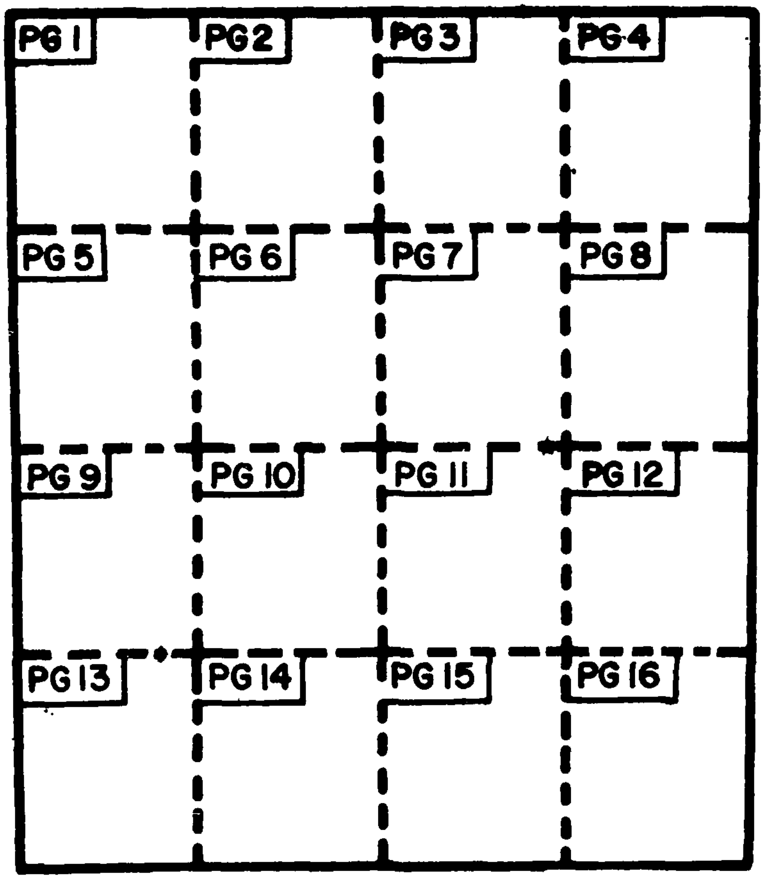


EXHIBIT IDS010-9

IDS PAGE/FILE CONCEPT

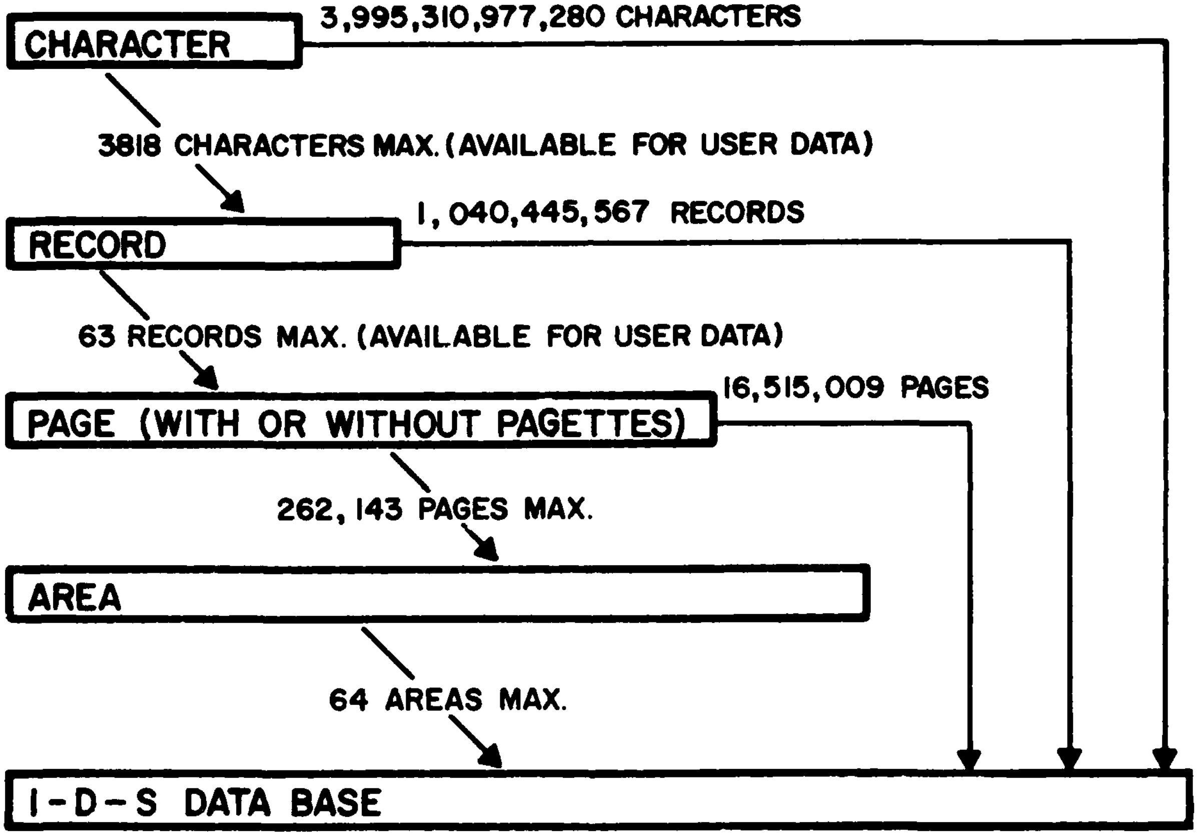


CHARACTERISTICS: 1. Each page is a data storage area which is the "unit of transfer" between disc and core.

2. The length of the "page" (in sector multiples) can be varied.

EXHIBIT IDS011-1

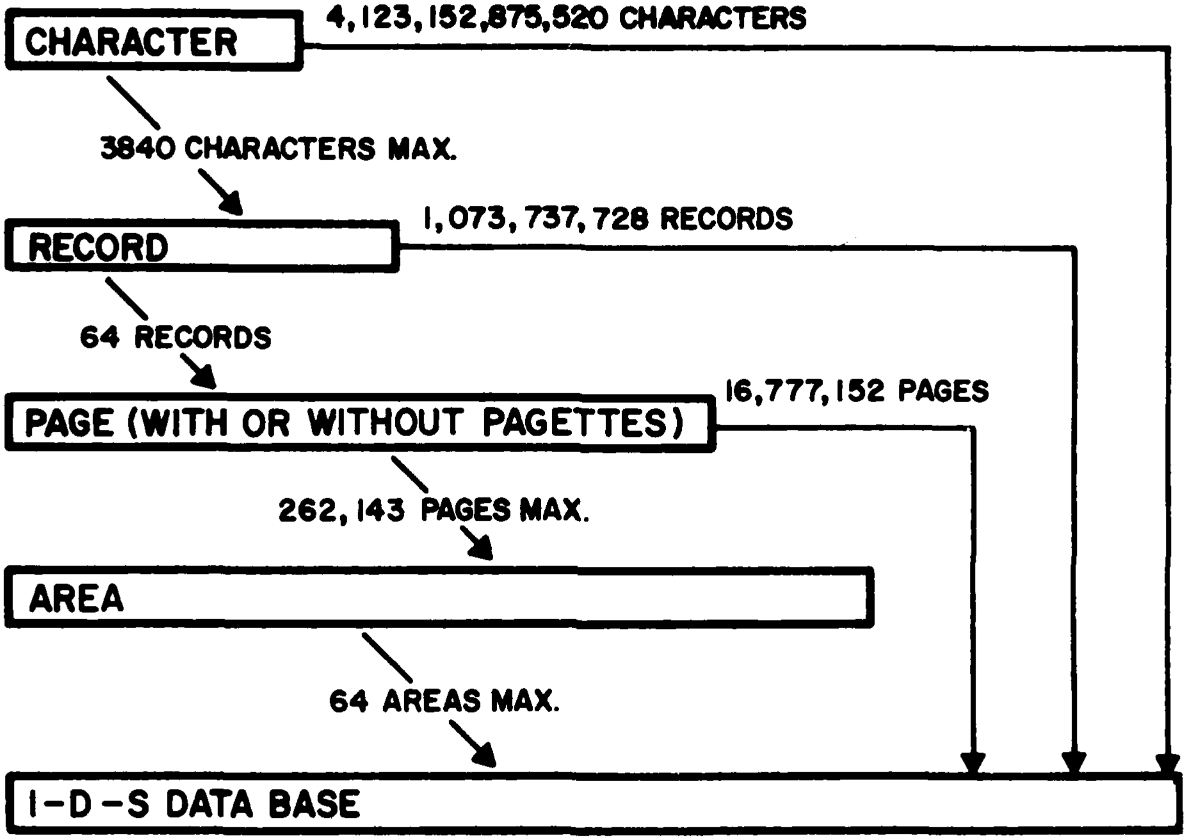
IDS CHARACTERISTICS


CHARACTER
3,995,310,977,280 CHARACTERS
3818 CHARACTERS MAX. (AVAILABLE FOR USER DATA)
RECORD
1,040,445,567 RECORDS
63 RECORDS MAX. (AVAILABLE FOR USER DATA)
PAGE (WITH OR WITHOUT PAGETTES)
16,515,009 PAGES
262,143 PAGES MAX.
AREA
64 AREAS MAX.
I-D-S DATA BASE

NOTE: These figures represent only the 63 user data lines (not line #0).

EXHIBIT IDS011-2

IDS CHARACTERISTICS



NOTE: These figure include line #0.

EXHIBIT IDS011-3

AN IDS CHAIN

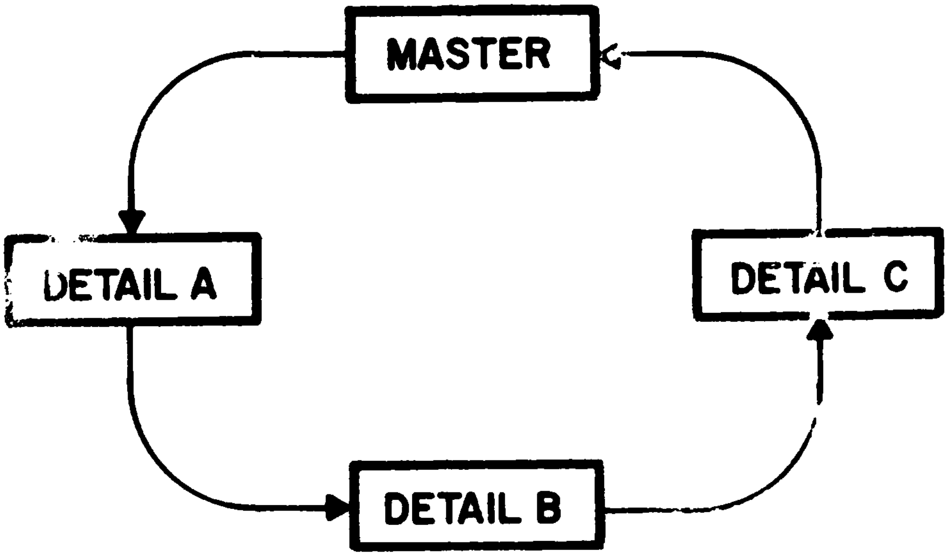


EXHIBIT IDS011-4

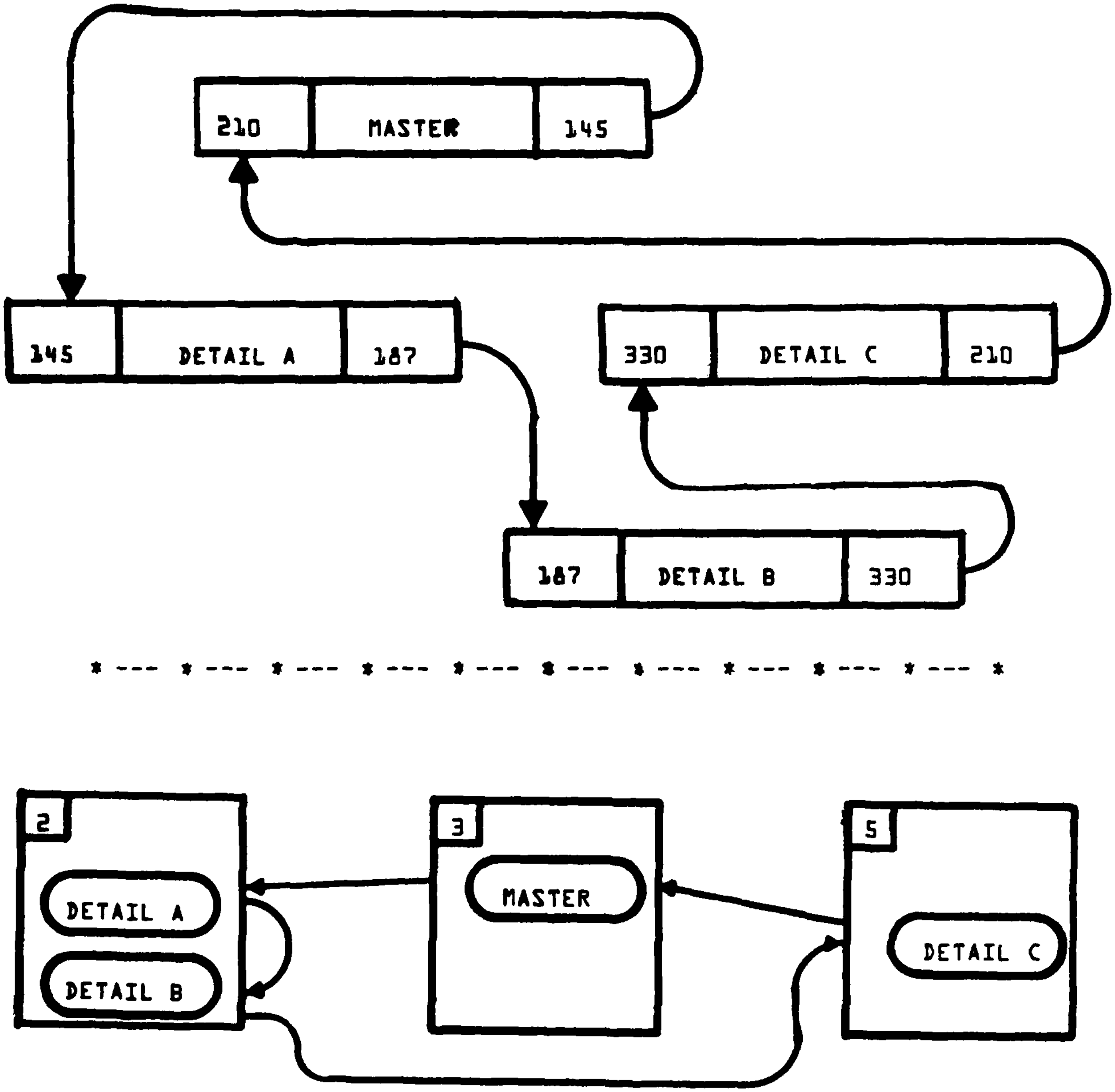


EXHIBIT IDS011-5

CHAIN WITH "NEXT" POINTERS

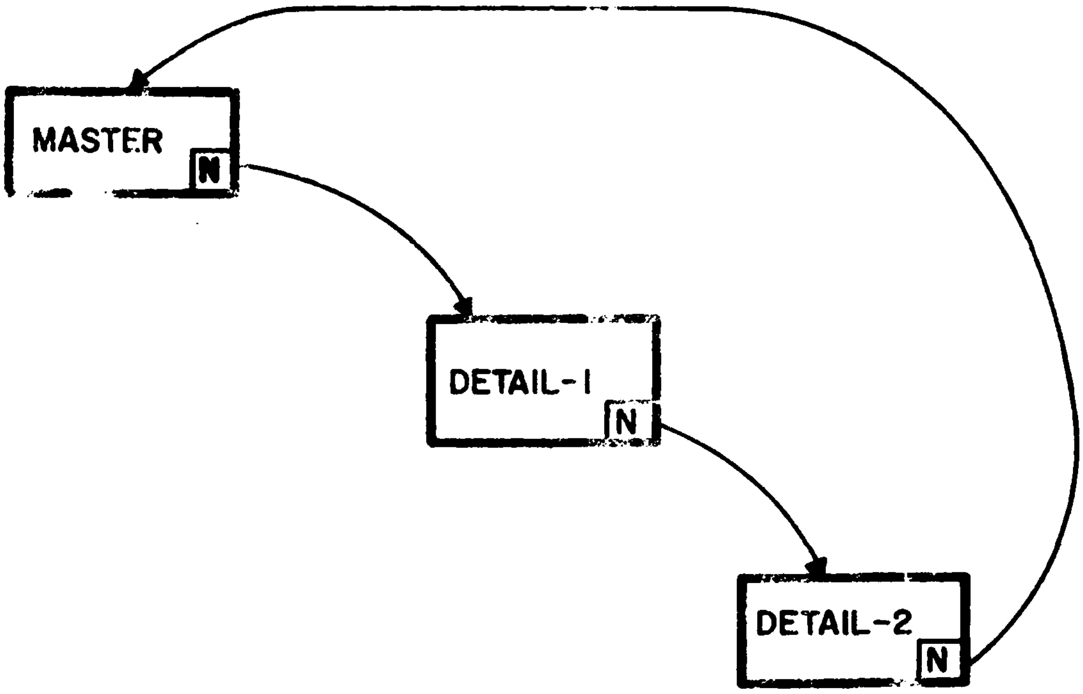


EXHIBIT IDS011-6

CHAIN WITH "NEXT" POINTERS
"PRIOR" POINTERS

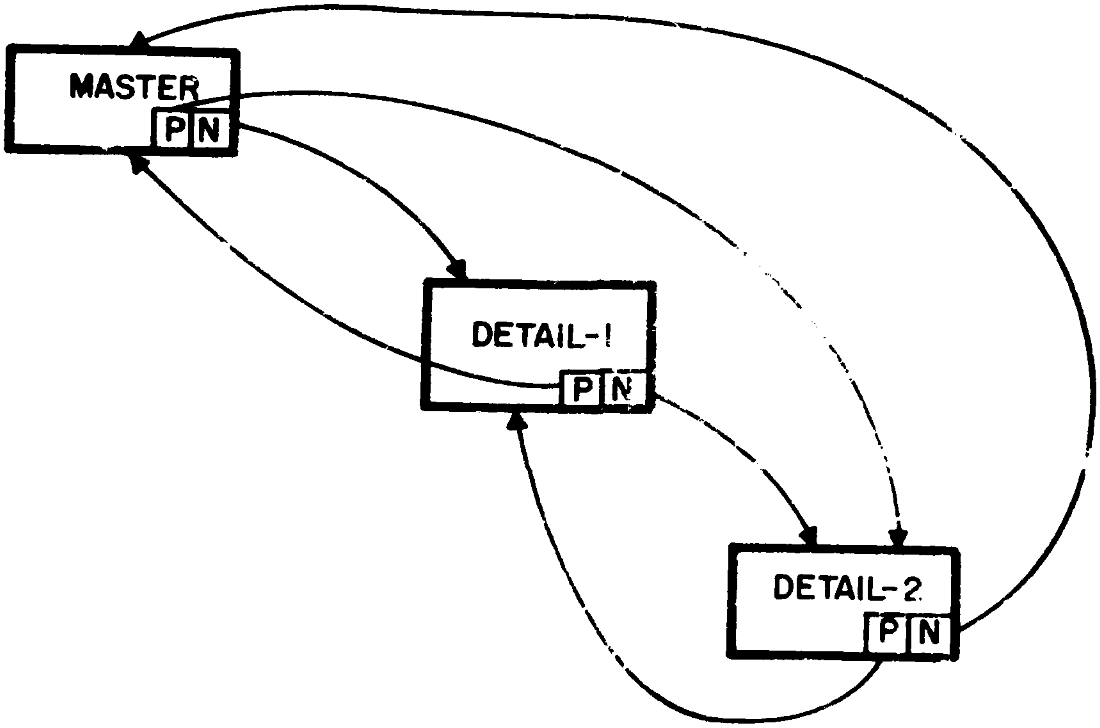


EXHIBIT IDS011-7

CHAIN WITH "NEXT" POINTERS
"PRIOR" POINTERS
"MASTER" POINTERS

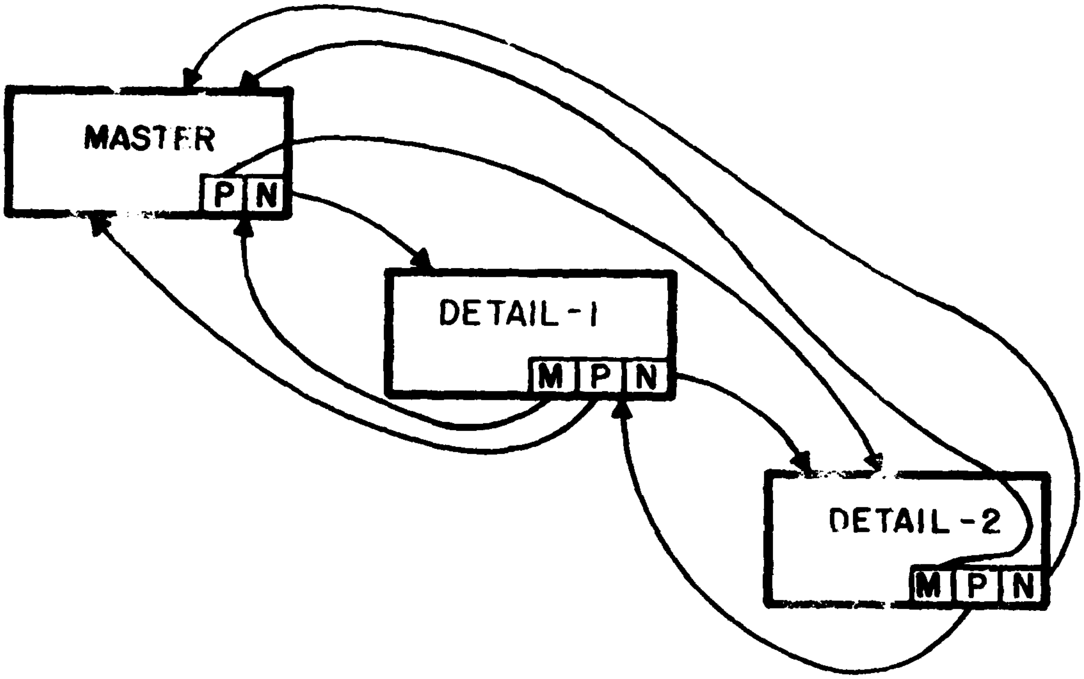
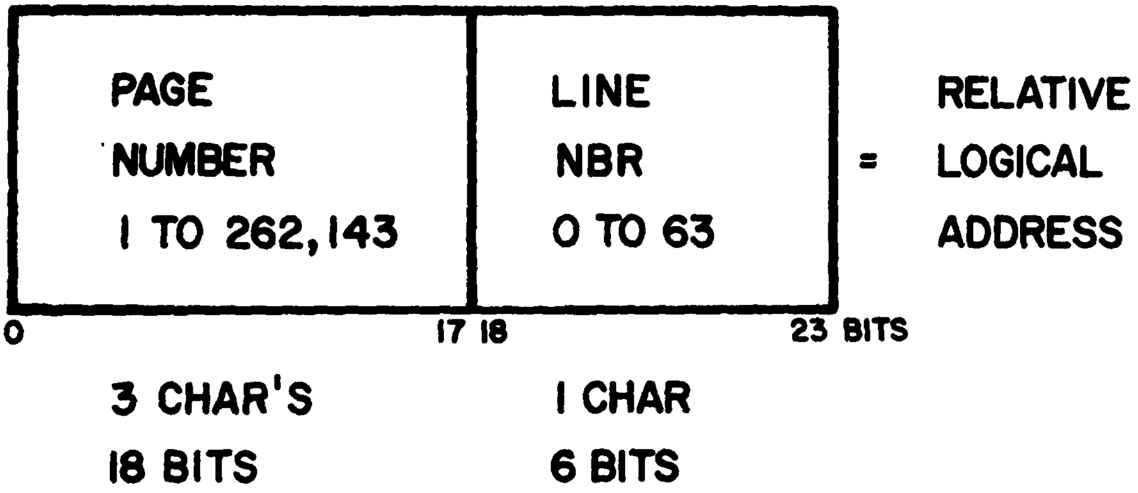


EXHIBIT IDS011-8

REFERENCE CODE
(CHAIN POINTER)



Reference codes are permanently assigned until physical deletion occurs.

EXHIBIT IDS011-9

MASTER/DETAIL

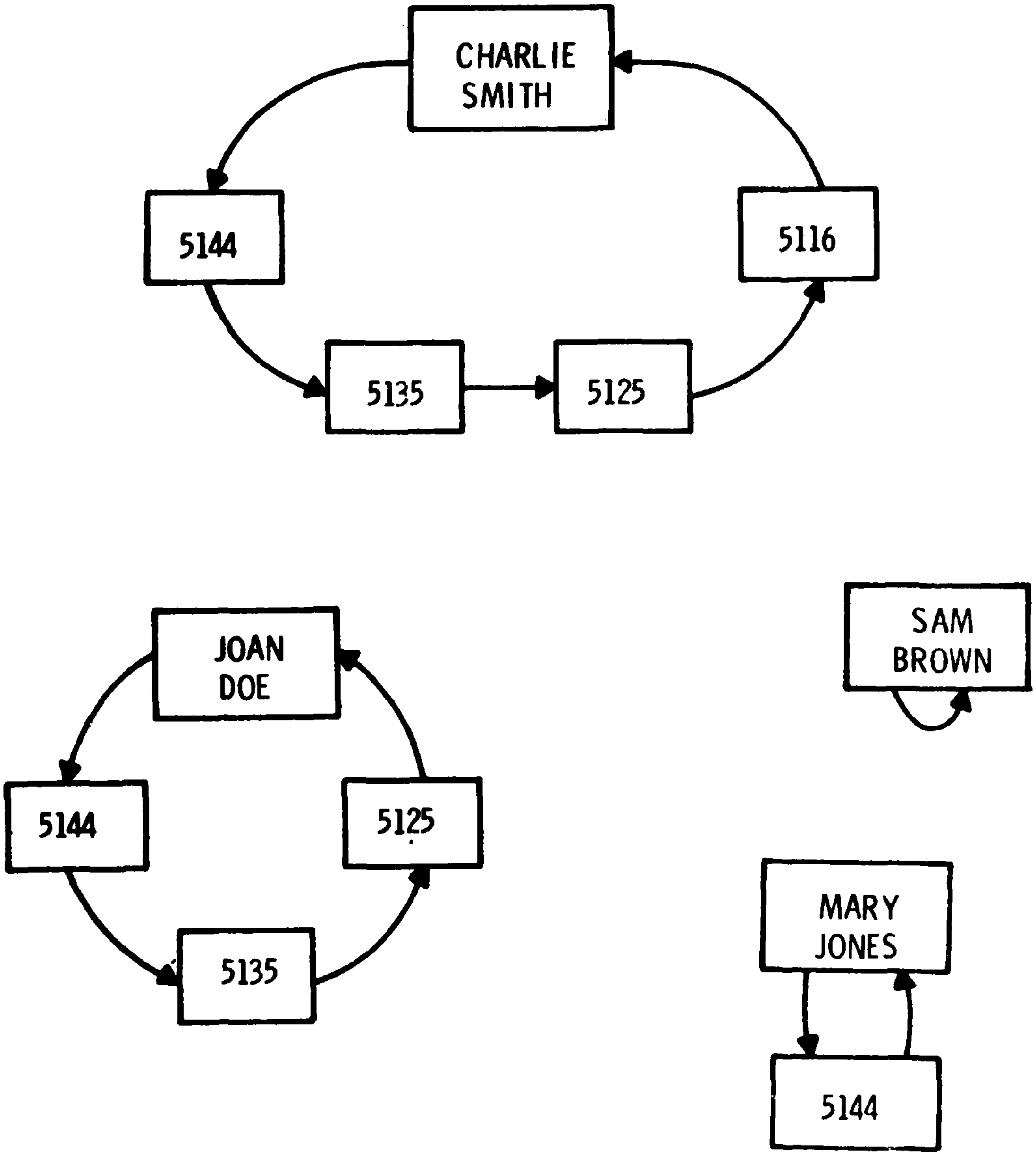


EXHIBIT IDS011-10

IDS CHAINING CHARACTERISTICS

1. Makes the logical relationships between records independent of physical devic*s.

2. Gives various access paths to records:

   A. Once a record has been accessed, it is then possible to access all the related records in the chain by following the pointers.

   B. Access to a record participating in a number of chains makes it possible to enter any of the related chains & retrieve records.

3. Permits chain processing in either direction beginning at any record in a chain.

EXHIBIT IDS011-11

IDS CHAINING ADVANTAGES

1. Permits linking together logically related data.

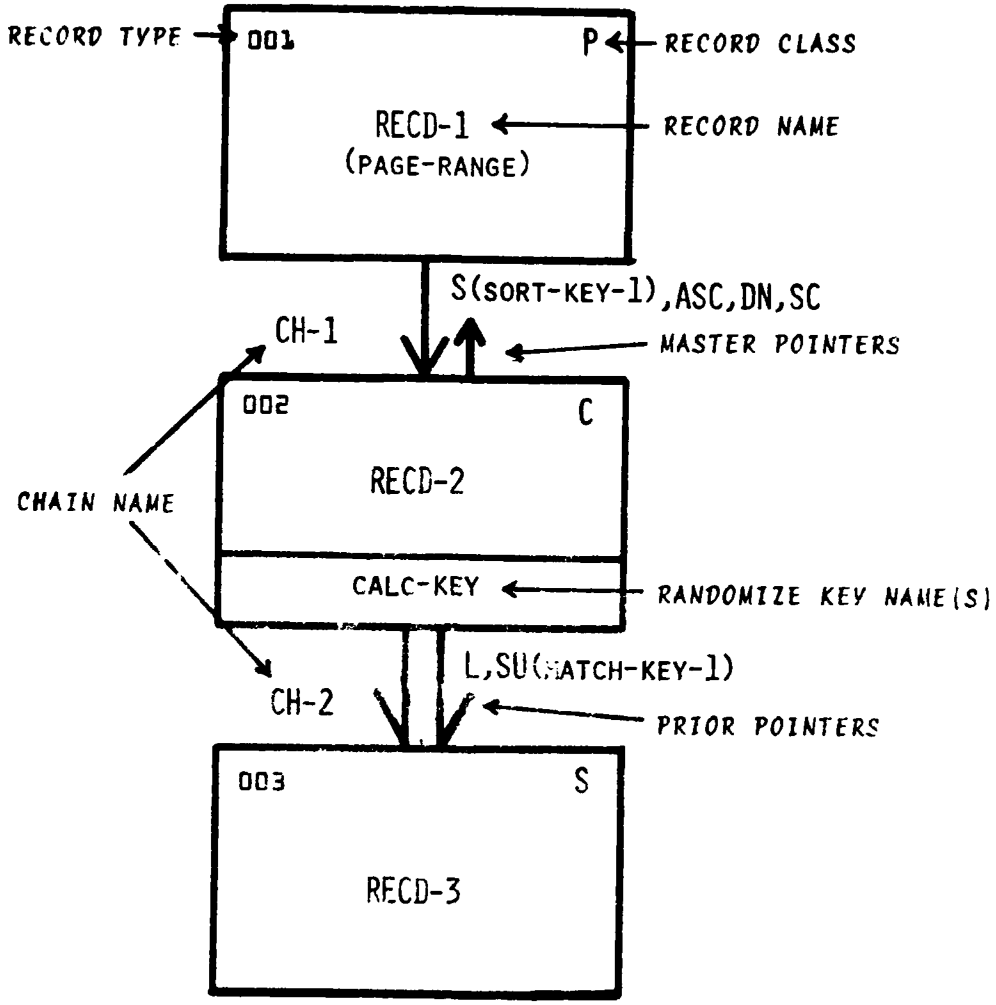
2. Permits the design of a data base in a way to reflect actual organizational relationships.

3. Permits interrelationships between different sets of records by chaining rather than by duplication.

4. One fact provides access to all related facts.

EXHIBIT IDS011-12

STRUCTURE DIAGRAM CONVENTIONS



Chain Orders
First - F
Last - L
Before - B
After - A
Sorted - S (sort-key-1,...)

Sorted within type - T

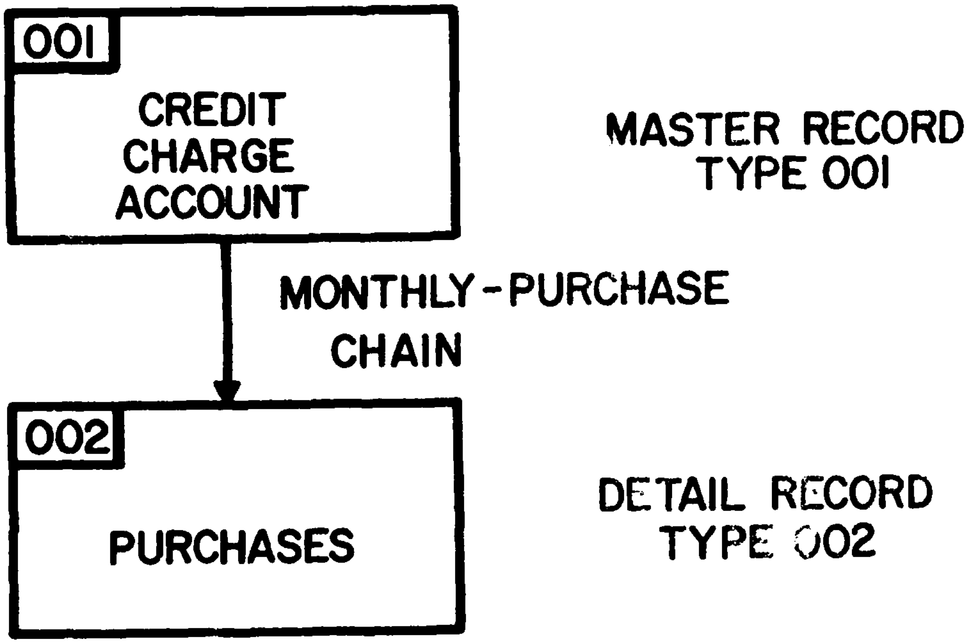
Record Class
Primary - P
Secondary - S
Calculated - C

Master Selection
Select Current - SC
Select Unique - SU (m-k,...)

Sorted Chains
Duplicates First - DF
Duplicates Last - DL
Duplicates N/A - DN
Ascending - ASC
Descending - DSC

EXHIBIT IDS012-1

SAMPLE MASTER/DETAIL RELATIONSHIP



— SHORTHAND —

EXHIBIT IDS012-2

FOUR OCCURRENCES OF PURCHASE CHAIN
-LONGHAND-

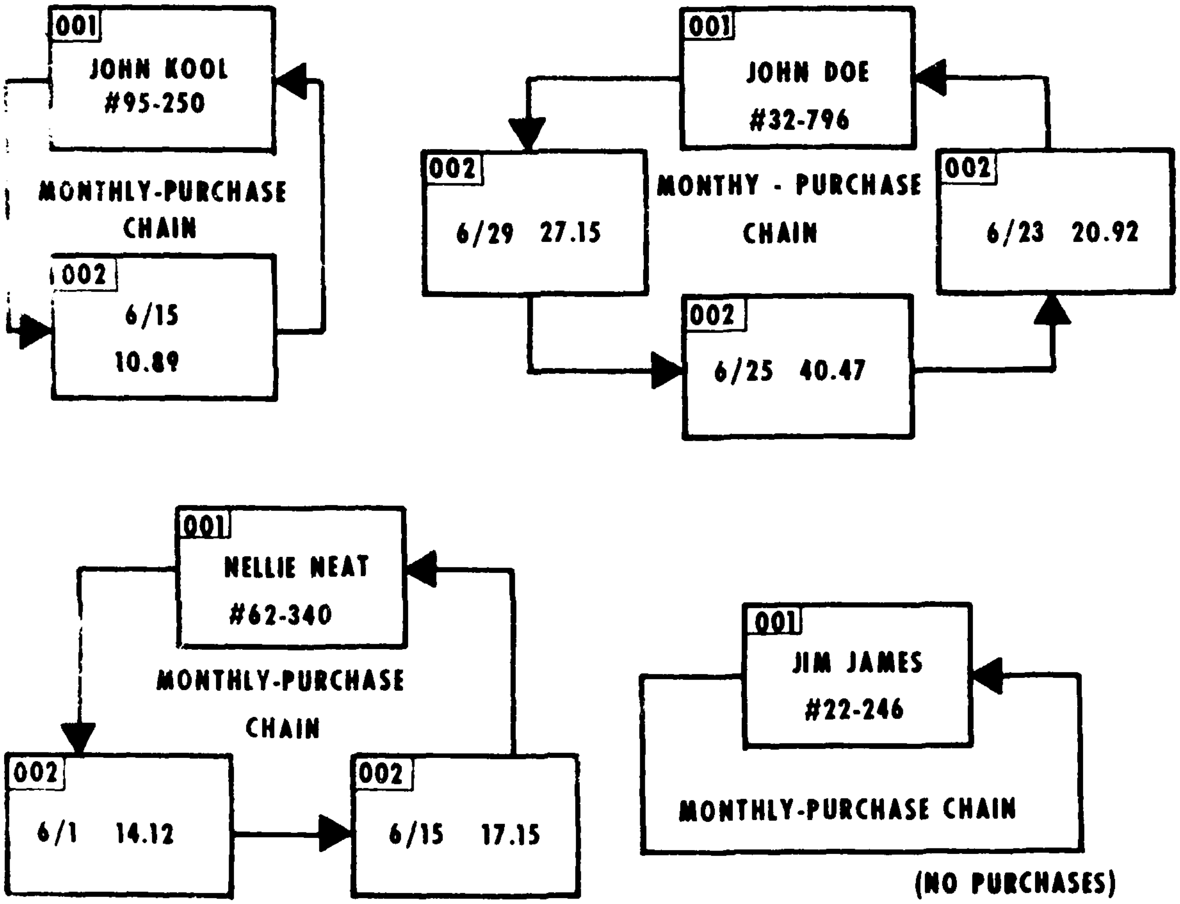
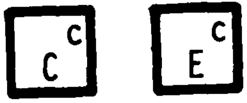


EXHIBIT IDS012-3

RECORD CLASSES

- CALC RECORDS

SHORTHAND



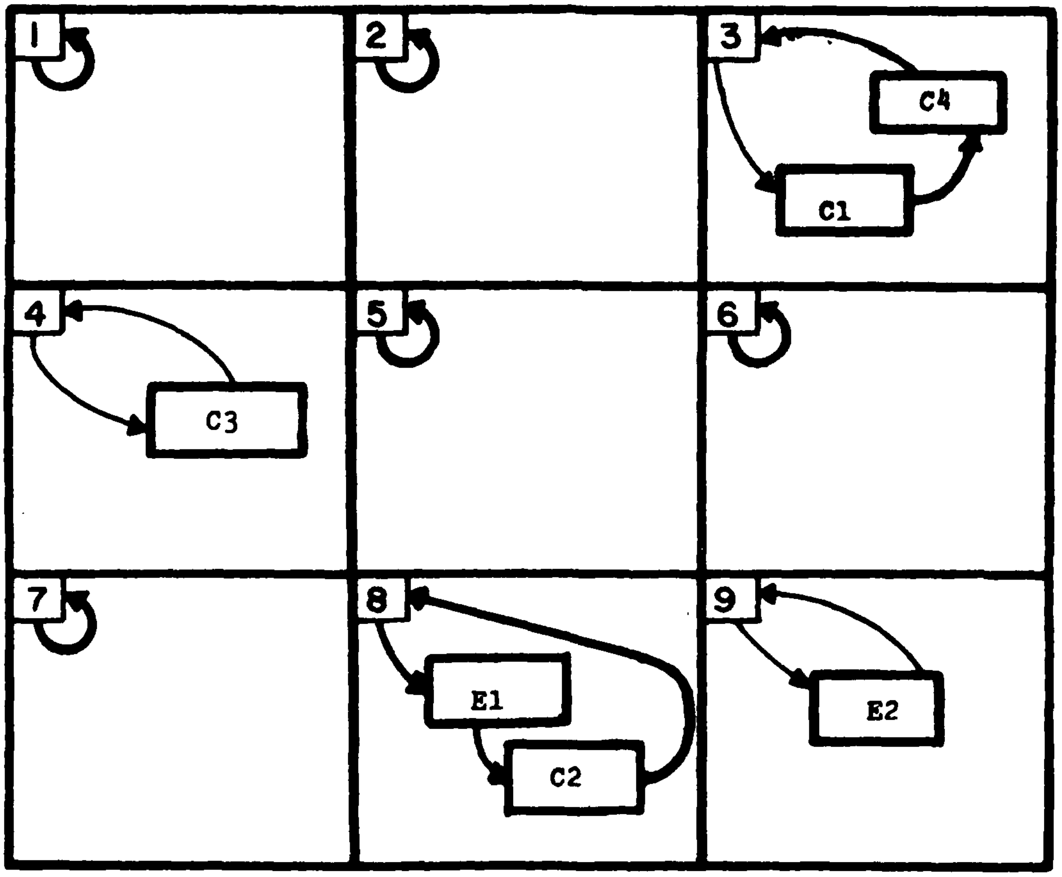
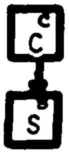


EXHIBIT IDS012-4

RECORD CLASSES

- SECONDARY RECORDS

SHORTHAND



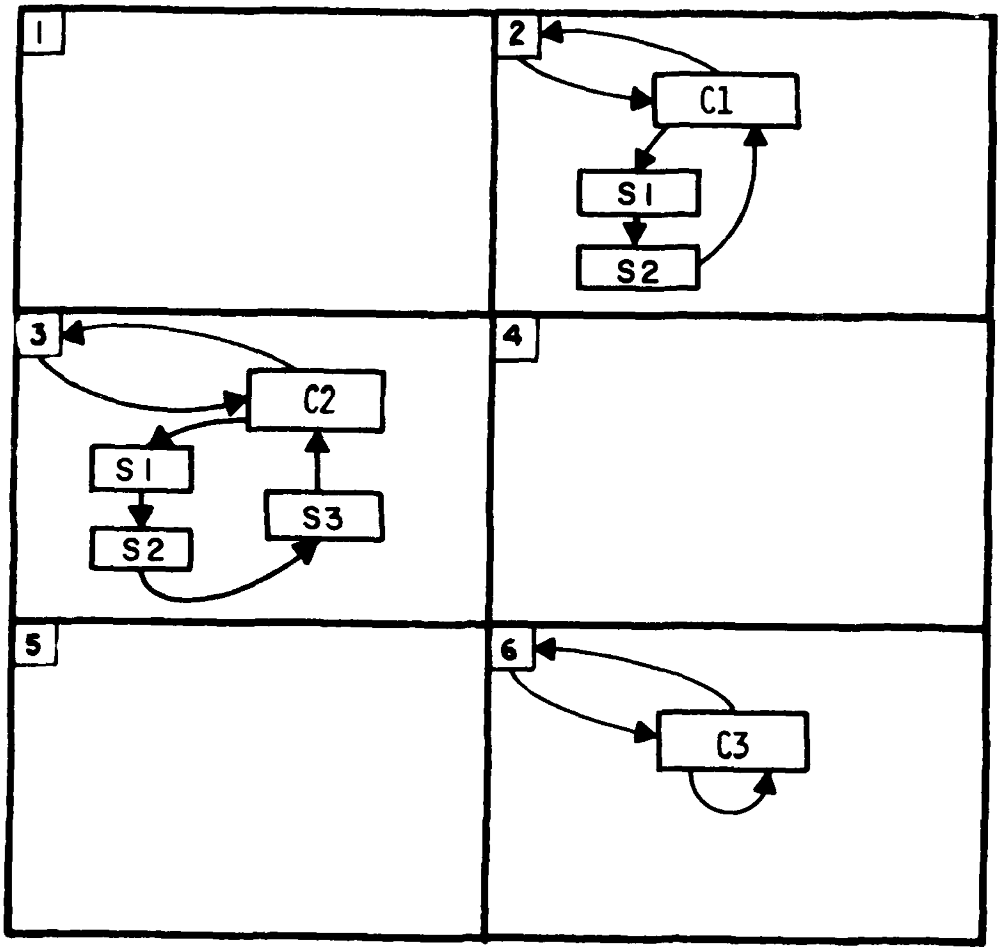


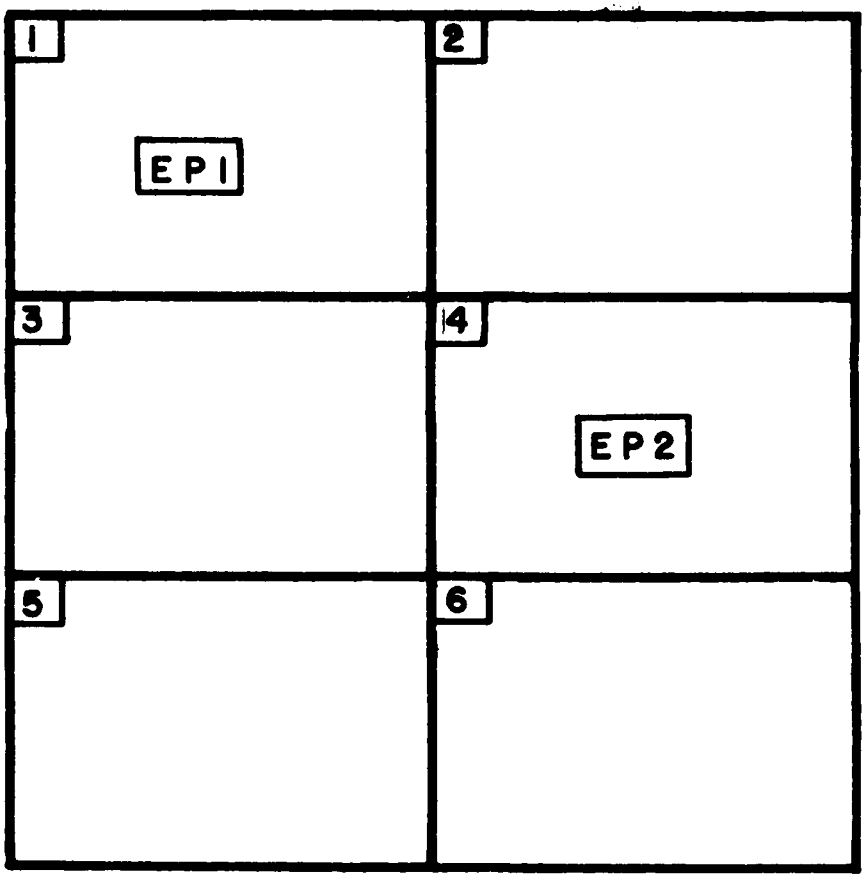
EXHIBIT IDS012-5

SHORTHAND

EP^{P}

RECORD CLASSES

- PRIMARY RECORDS



1

EP1

2

3

4

EP2

5

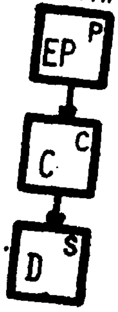
6

EXHIBIT IDS012-6

RECORD CLASSES

- PRIMARY RECORDS
- CALC RECORDS
- SECONDARY RECORDS

SHORTHAND



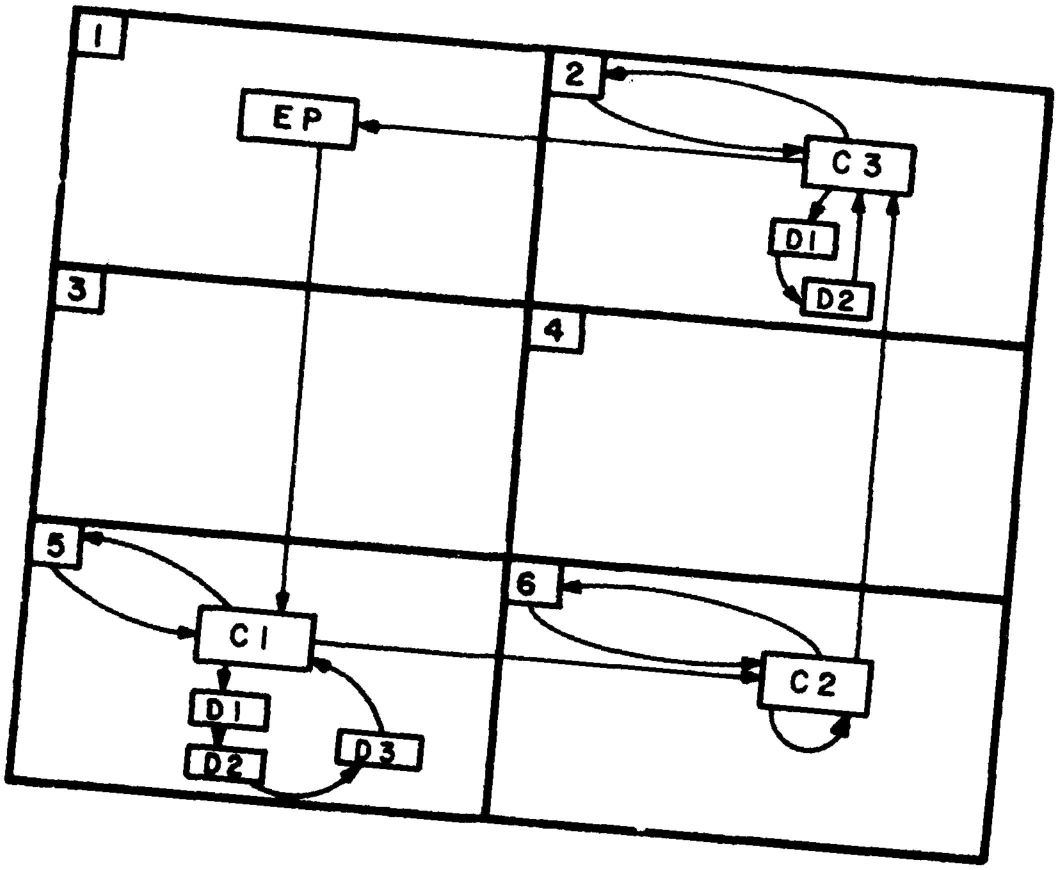
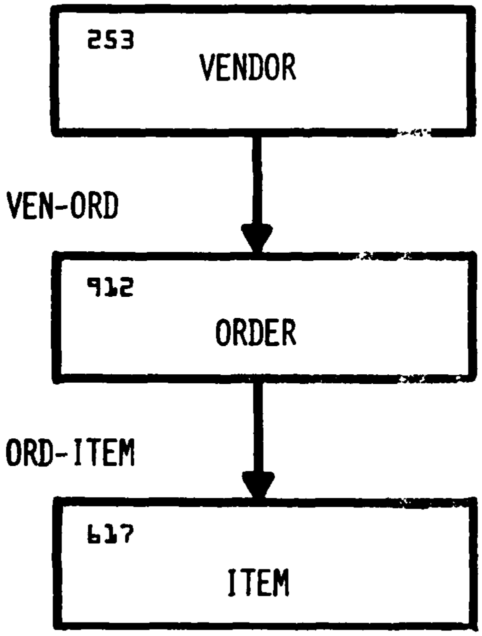


EXHIBIT IDS012-7

HIERARCHICAL STRUCTURE
(SHORTHAND)



Vendor Record - Master of "VEN-ORD" chain

Order Record - Detail of "VEN-ORD" chain/master of "ORD-ITEM" chain

Item Record - Detail of "ORD-ITEM" chain

EXHIBIT IDS013-1

HIERARCHICAL STRUCTURE
- LONGHAND -

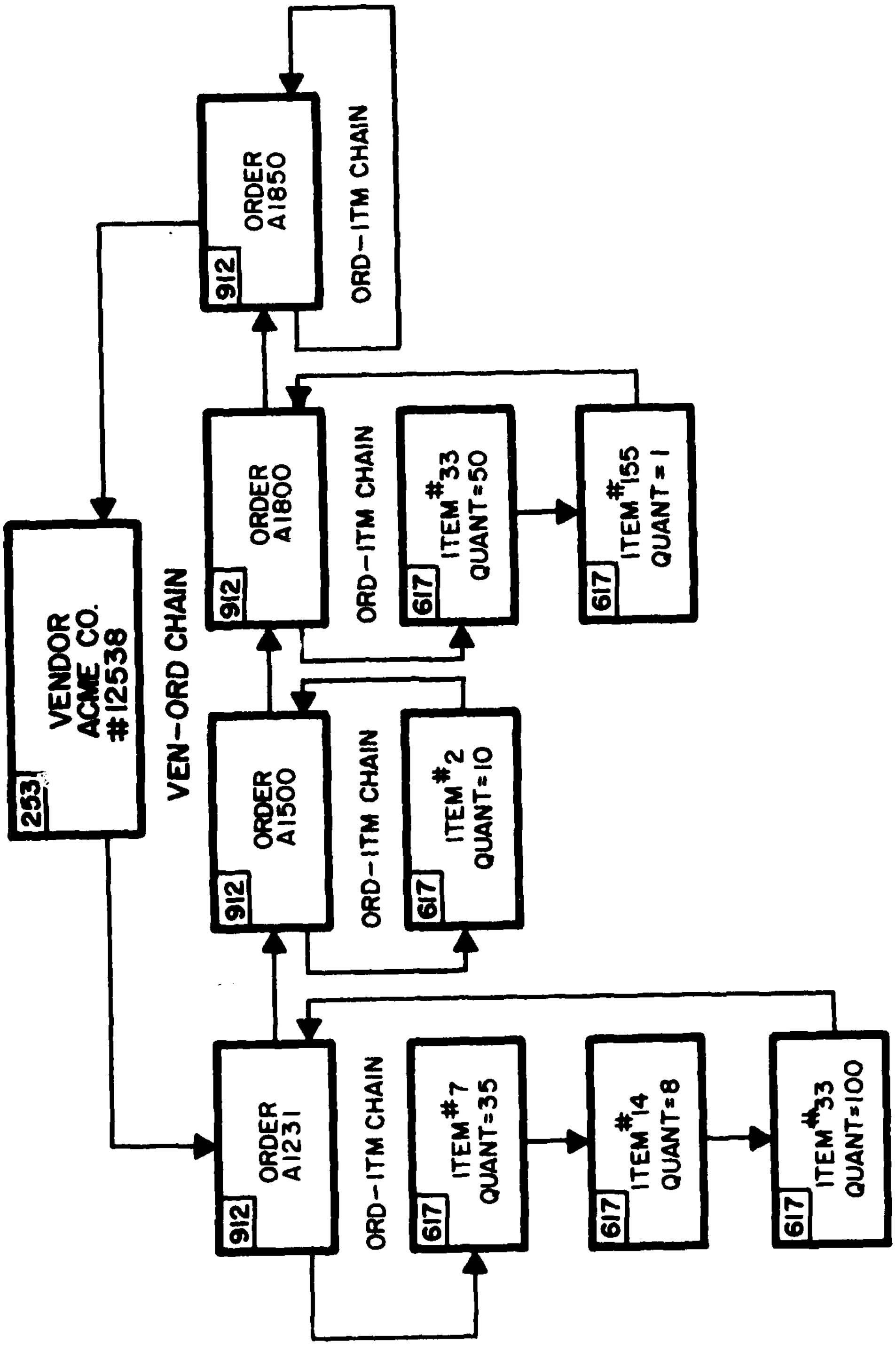
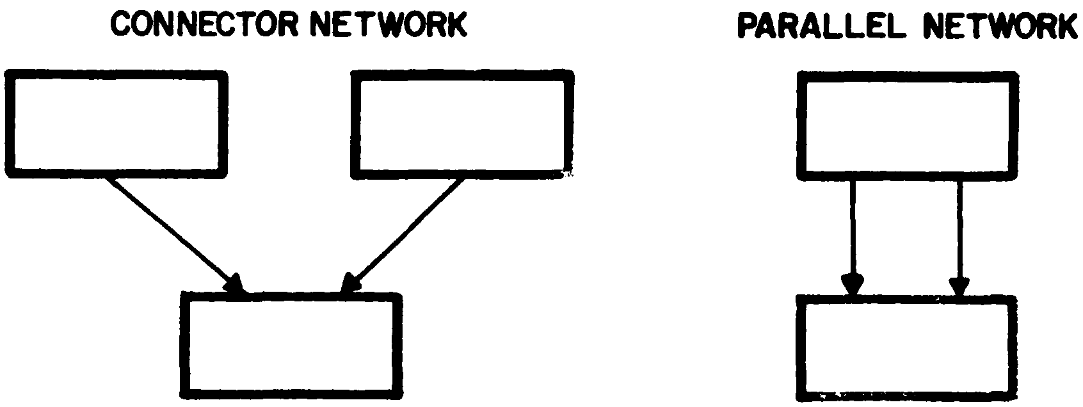


EXHIBIT IDS013-2

NETWORKS



If there are no network relationships in your data base, it is probably not beneficial to use IDS.

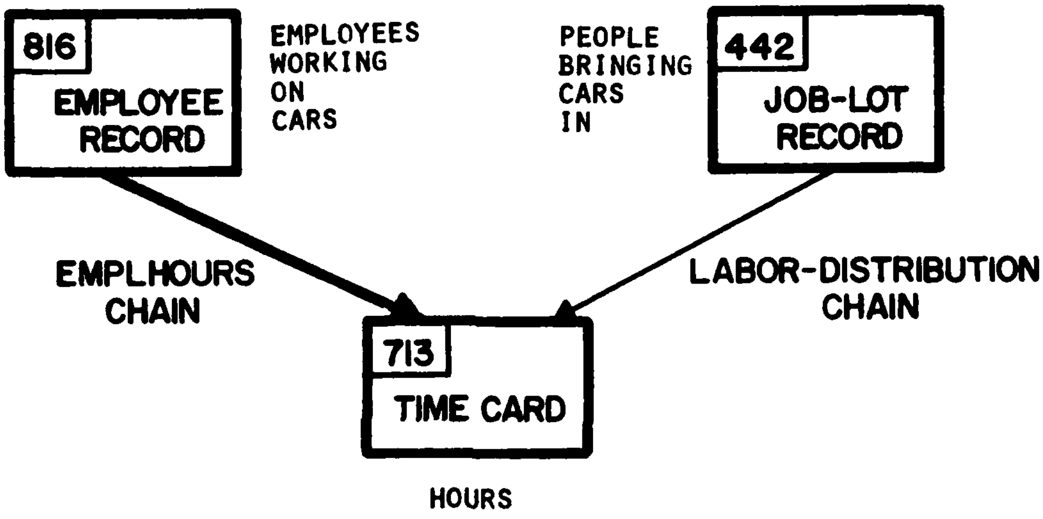
EXHIBIT IDS013-3

SITUATION: Automobile Repair Shop

Different people work on a car//paid by the hour

Different people bring cars in//charged by the hour

DETAIL OF MORE THAN ONE CHAIN
(SHORTHAND)



Heavy line designates the "prime" chain used for clustering purposes.

EXHIBIT IDS013-4

DETAIL OF MORE THAN ONE CHAIN -LONGHAND-

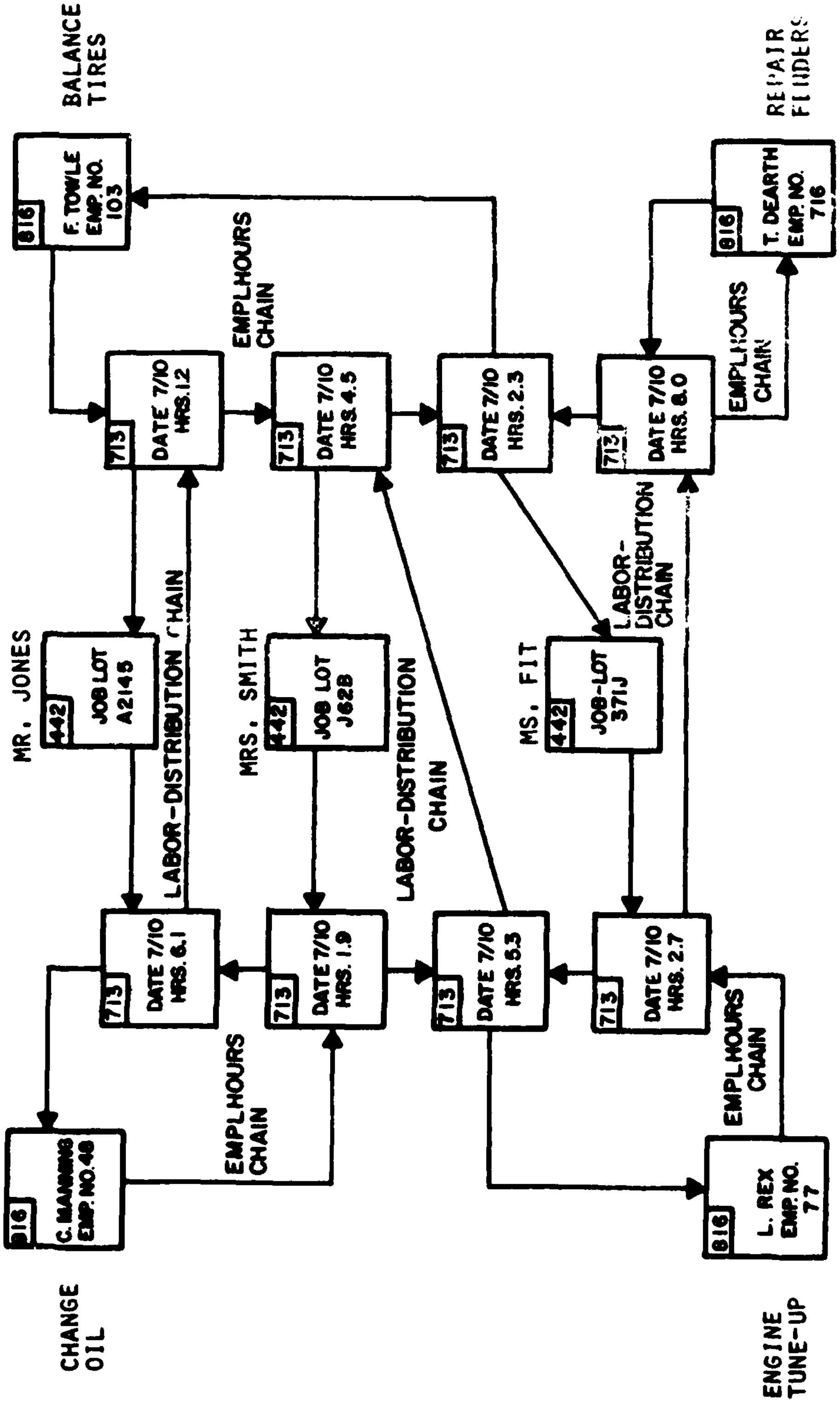


EXHIBIT IDS013-5

MASTER OF MORE THAN ONE CHAIN
(SHORTHAND)

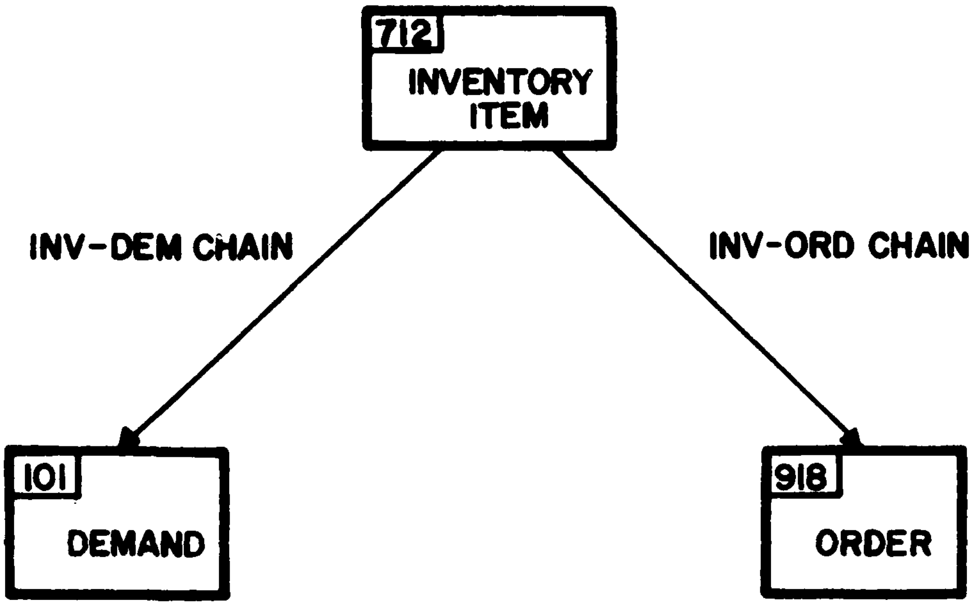


EXHIBIT IDS013-6

MASTER OF MORE THAN ONE CHAIN
(LONGHAND)

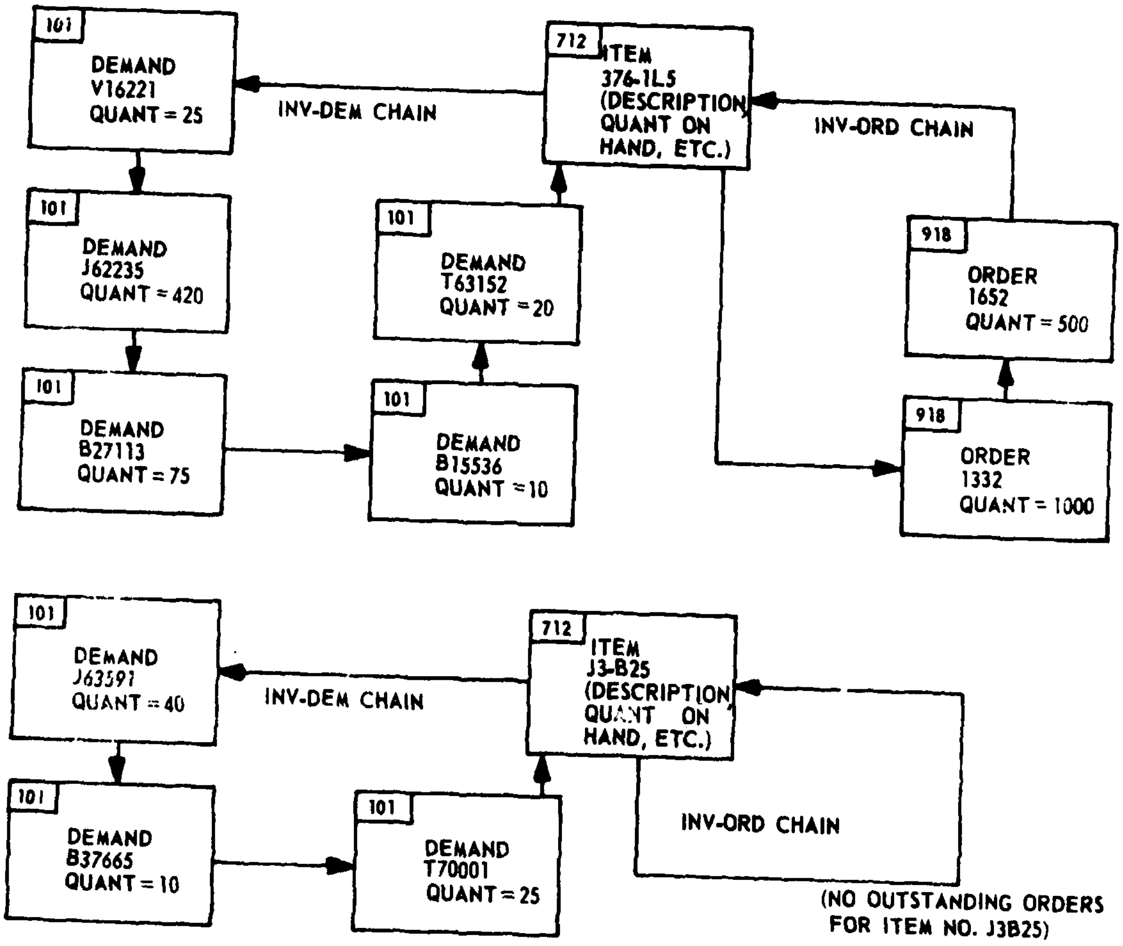


EXHIBIT IDS013-7

MULTIPLE TYPE DETAILS PER CHAIN
(SHORTHAND)

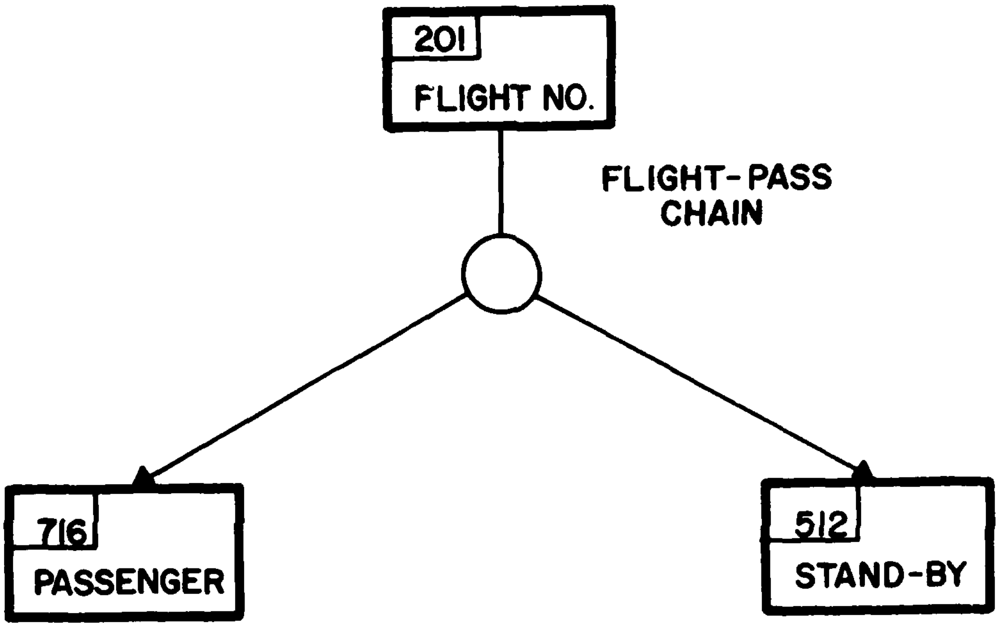


EXHIBIT IDS013-8

MULTIPLE TYPE DETAILS PER CHAIN
(LONGHAND)

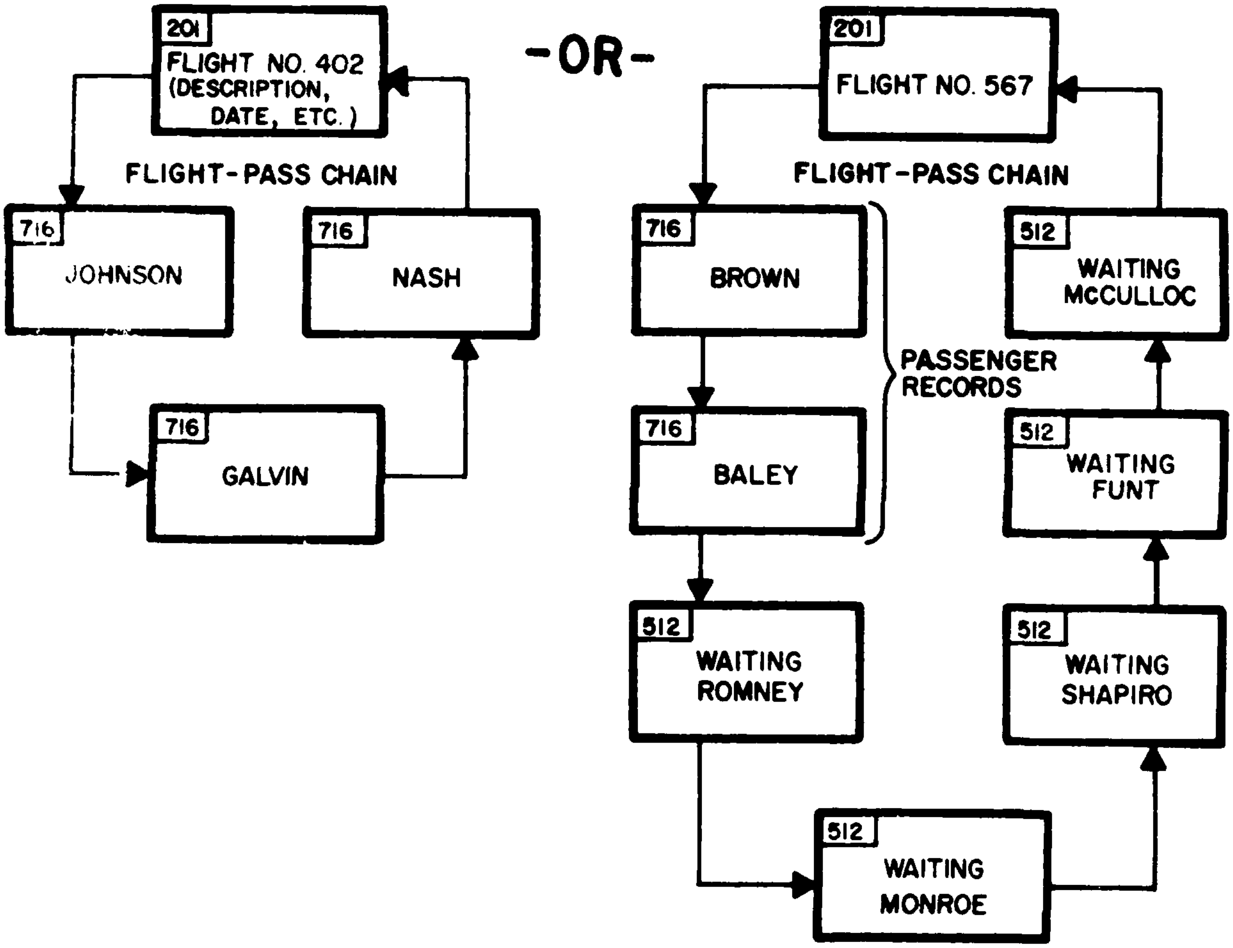


EXHIBIT IDS013-9

MASTER/DETAIL BETWEEN DETAILS

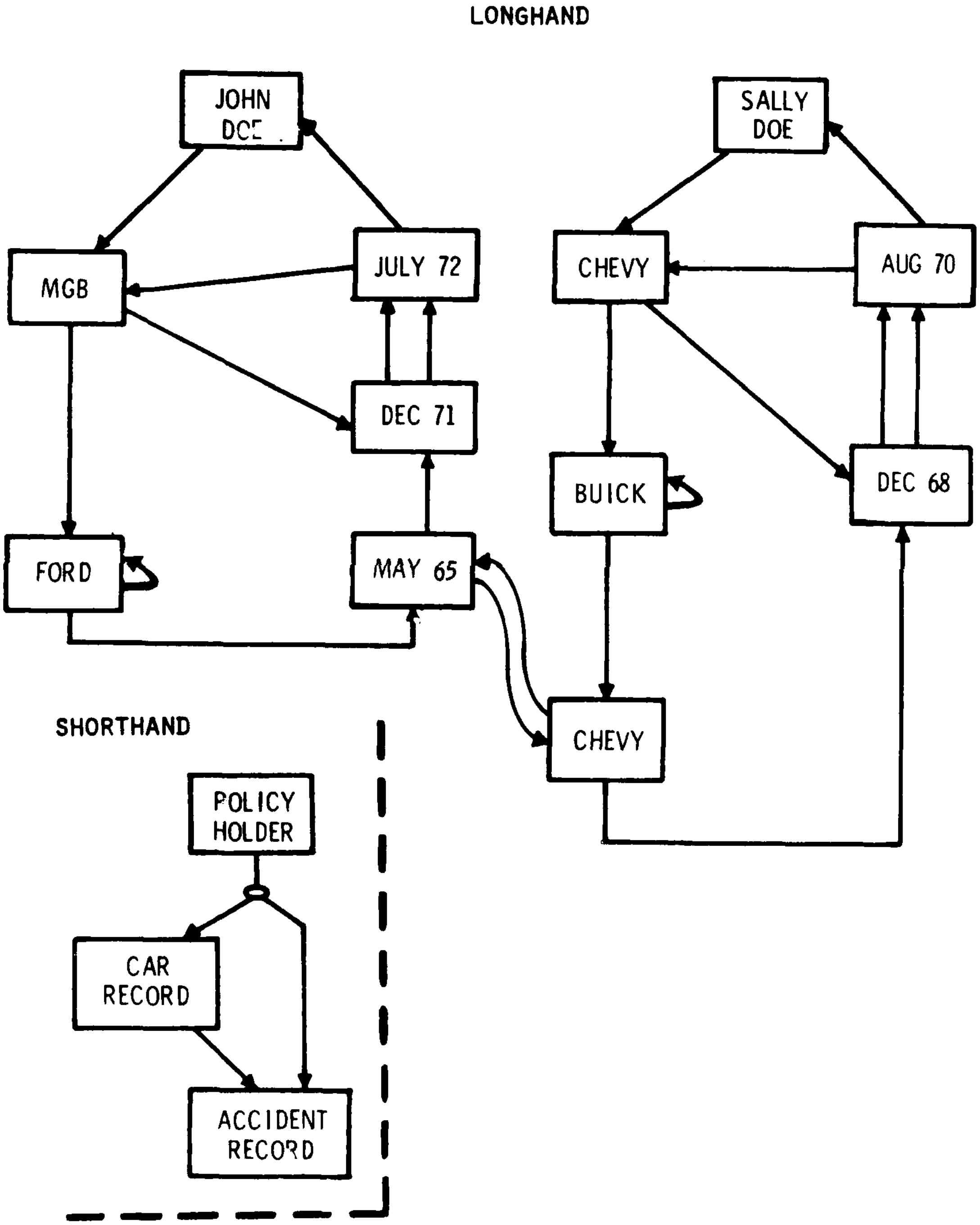
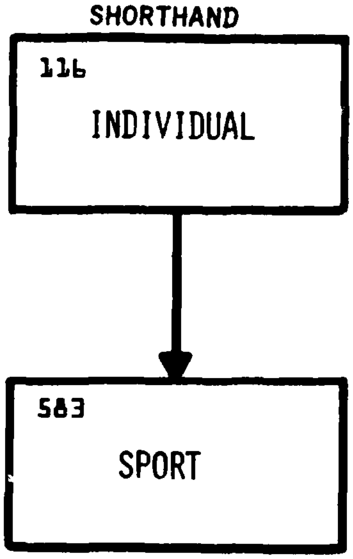


EXHIBIT IDS013-10



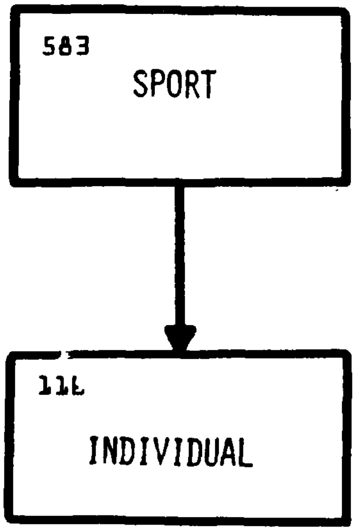
(YES) An individual can exist without participating in a sport.

(YES) An individual can participate in any number of sports.

(NO) A sport can't exist unless someone participates in it.

Additionally, a sport must be stored for each individual who participates in it.

EXHIBIT IDS013-11



(YES) A sport can exist without someone participating in it.

(YES) A sport can have any number of individuals participating in it.

(NO) An individual can't exist unless he participates in a sport.

Additionally, an individual must be stored for as many sports as he/she participates in.

EXHIBIT IDS013-12

DETAILS USED AS CONNECTOR RECORDS
(SHORTHAND)

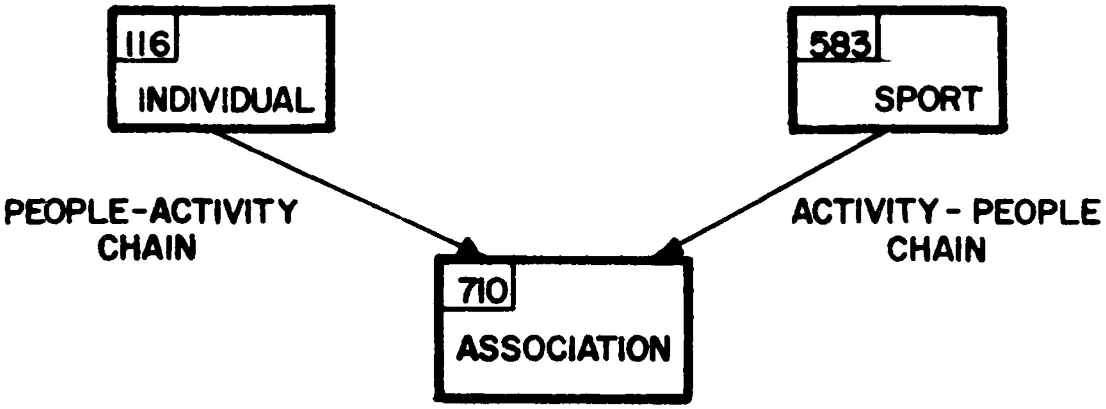
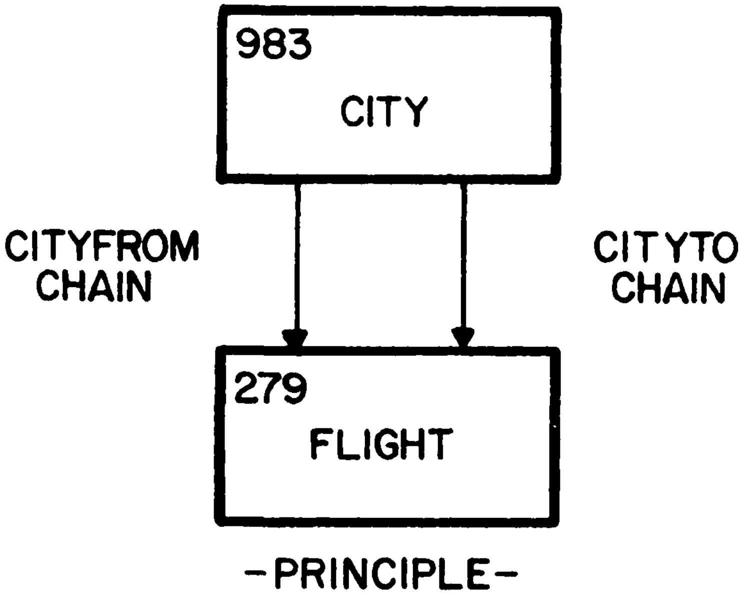


EXHIBIT IDS013-13

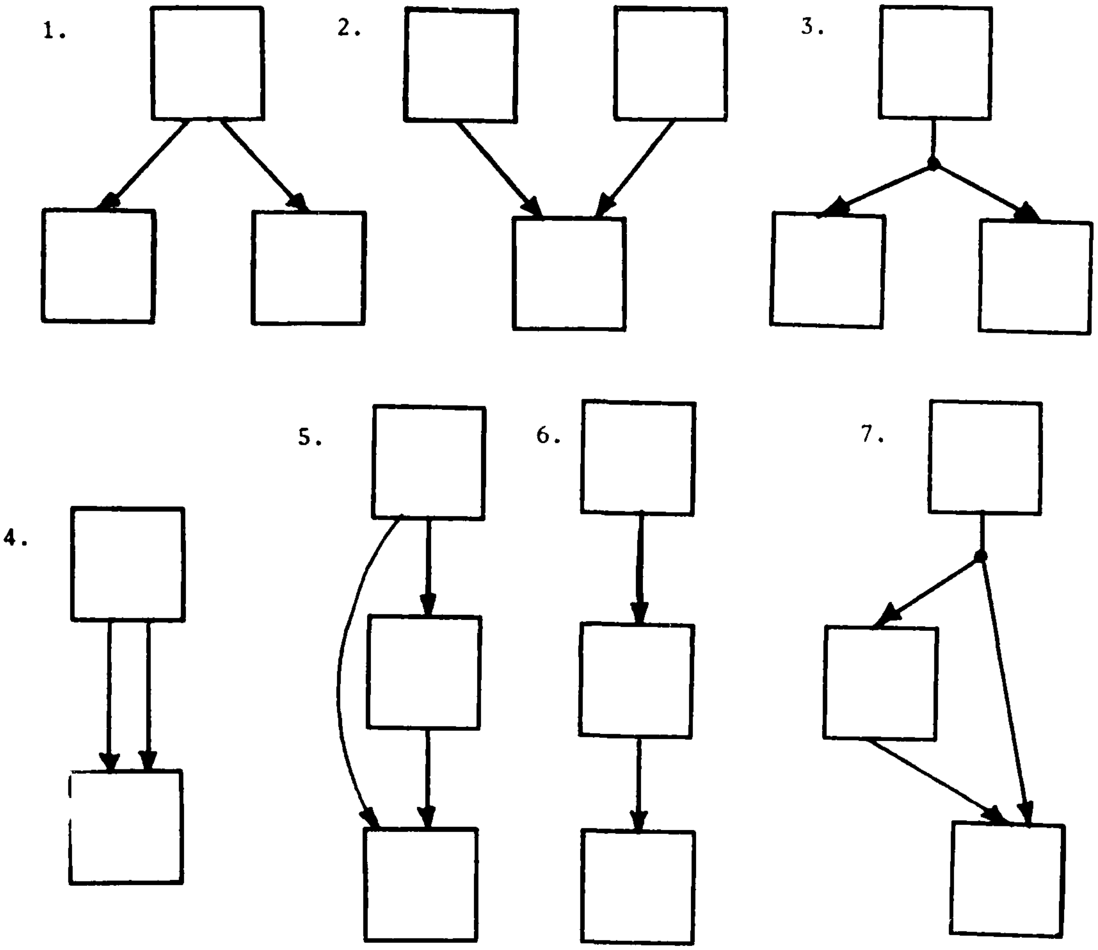
PARALLEL NETWORK RELATIONSHIP
(SHORTHAND)



Associates masters of the same record type.

EXHIBIT IDS013-14

LEGAL IDS DATA STRUCTURES



ILLEGAL

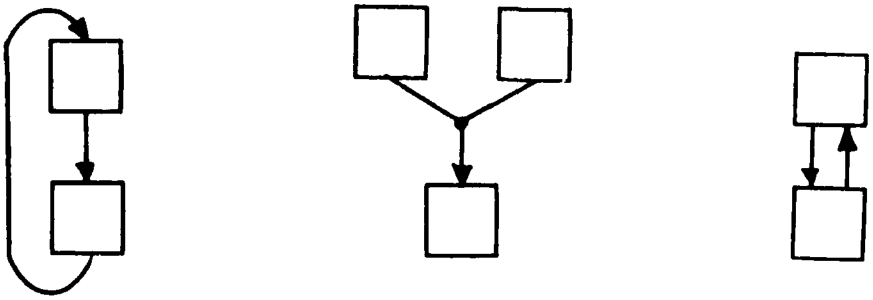


EXHIBIT IDS013-15

IDS CHAIN ORDERS

FIRST
LAST
SORTED

AFTER
BEFORE
SORTED WITHIN TYPE

EXHIBIT IDS020-1

CHAIN-ORDER FIRST

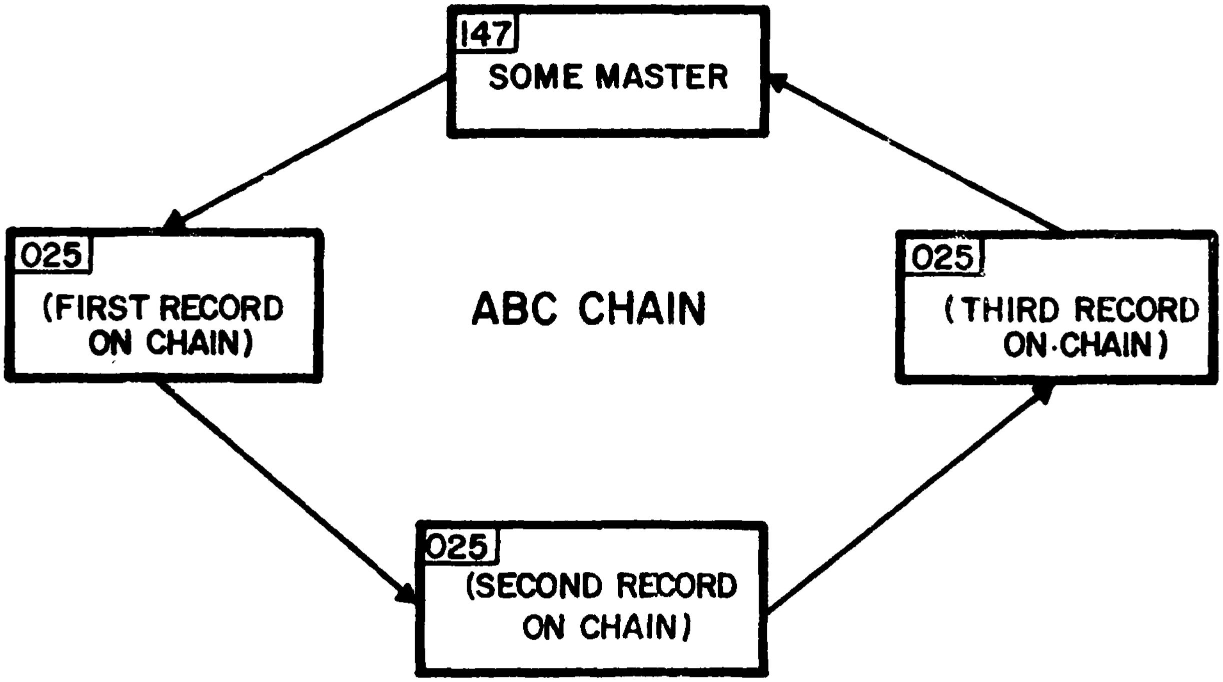
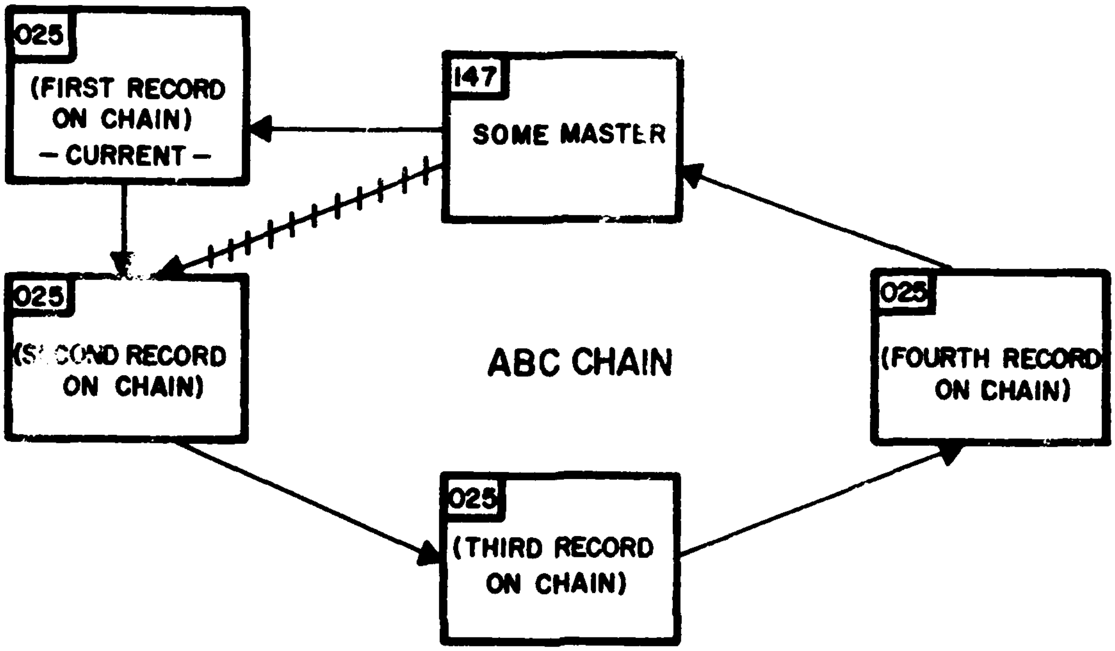


EXHIBIT IDS020-2

CHAIN ORDER FIRST



CHARACTERISTICS

1. Has the effect of a logical last in/first out sequence.

2. Requires only "NEXT" chain pointers.

EXHIBIT IDS020-3

CHAIN-ORDER LAST

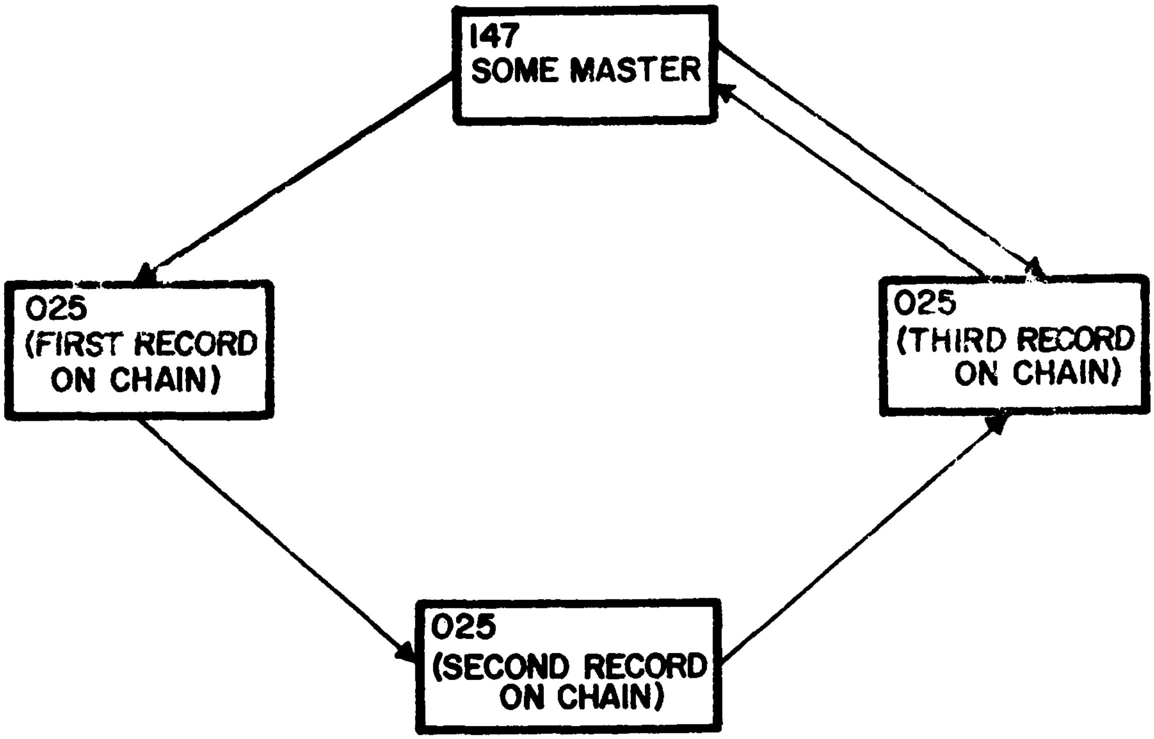
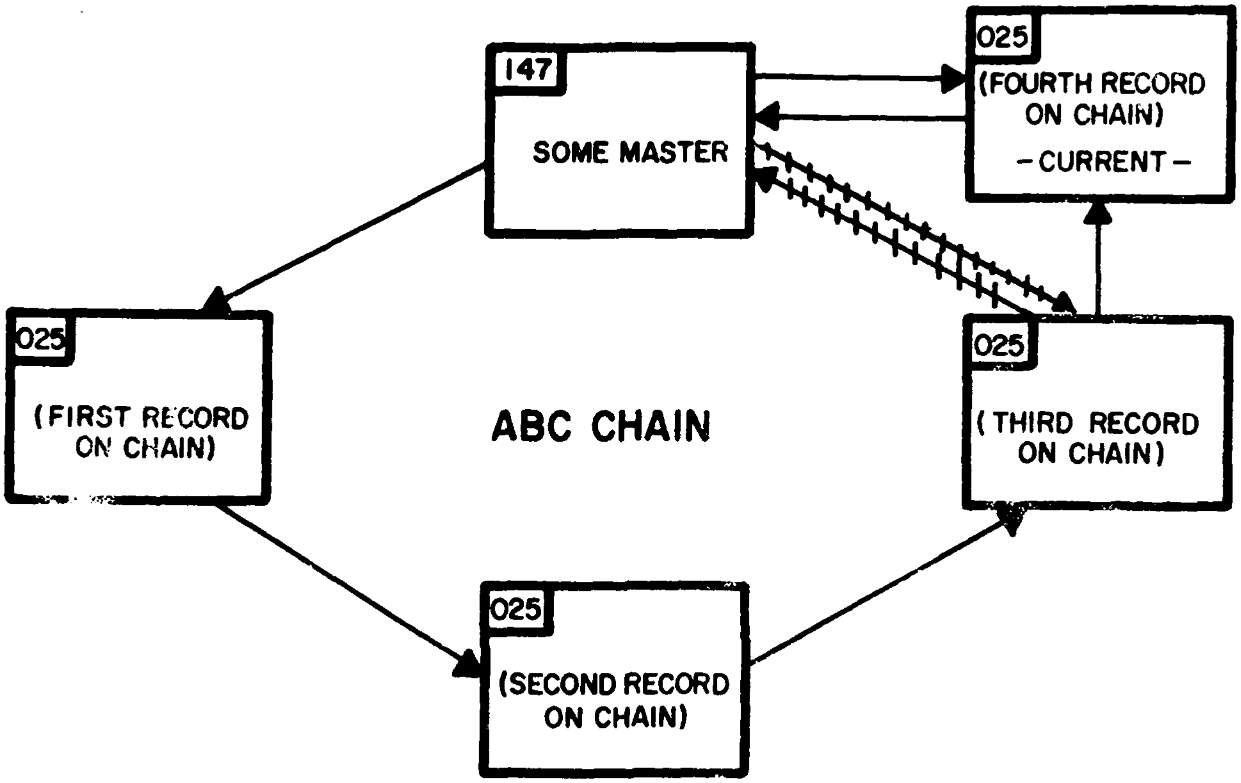


EXHIBIT IDS020-4

CHAIN-ORDER LAST



CHARACTERISTICS

1. Has the effect of first in/first out sequence.

2. Requires the IDS software to automatically include an additional "PRIOR" pointer in the master.

EXHIBIT IDS020-5

CHAIN-ORDER BEFORE

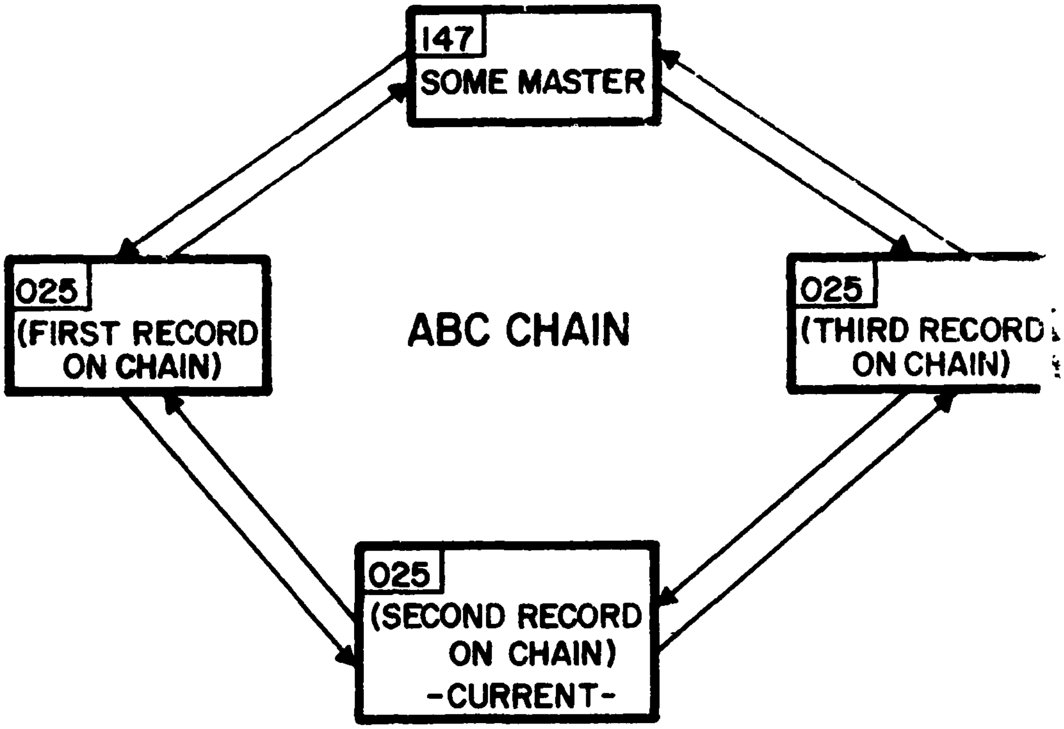
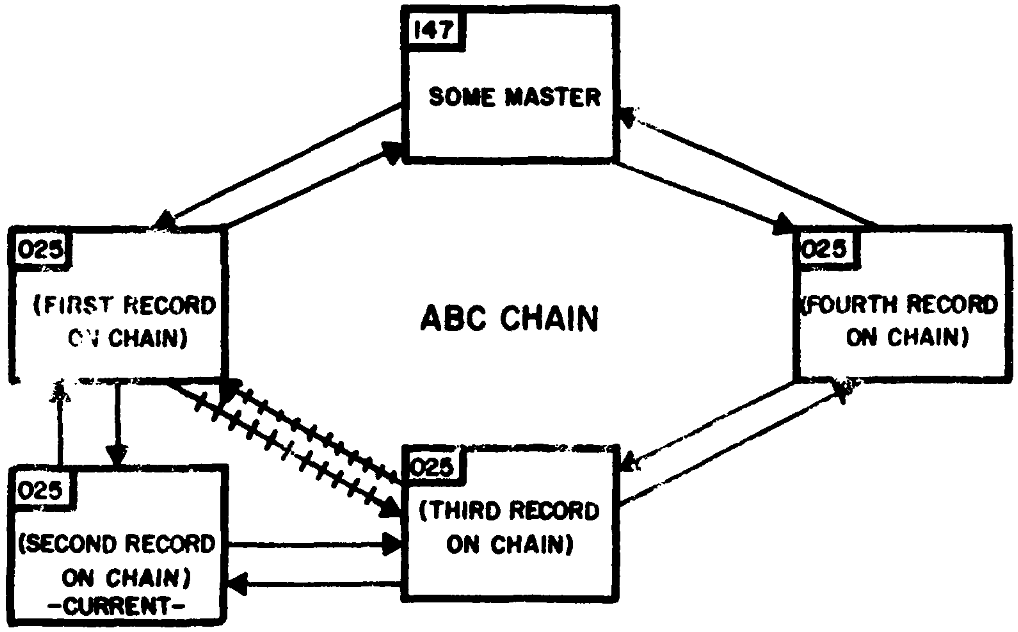


EXHIBIT IDS020-6

CHAIN-ORDER BEFORE



CHARACTERISTICS

1. Allows logical insertion of new record before the "CURRENT" record (master or detail) of the chain being processed.

2. Each record type in the chain includes a "PRIOR" pointer.

EXHIBIT IDS020-7

CHAIN-ORDER AFTER

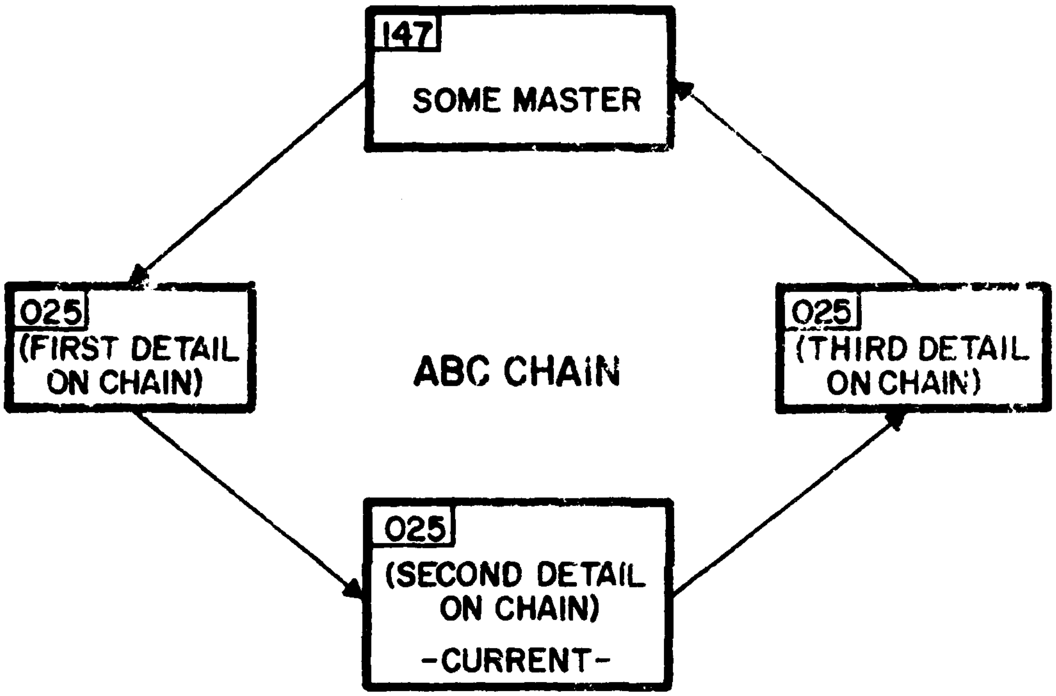
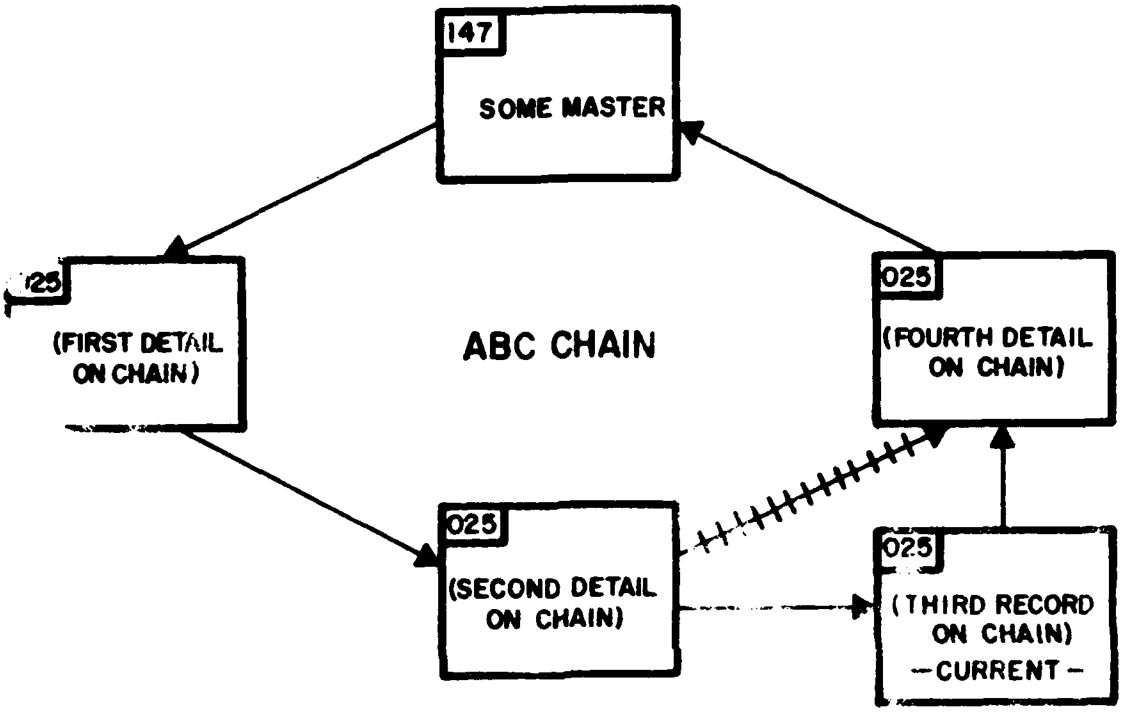


EXHIBIT IDSO?0-8

CHAIN-ORDER AFTER



CHARACTERISTICS

1. Does not require additional pointers other than "NEXT."

2. Allows logical insertion of new record after the "CURRENT" record (master or detail) of the chain being processed.

EXHIBIT IDS020-9

CHAIN-ORDER SORTED

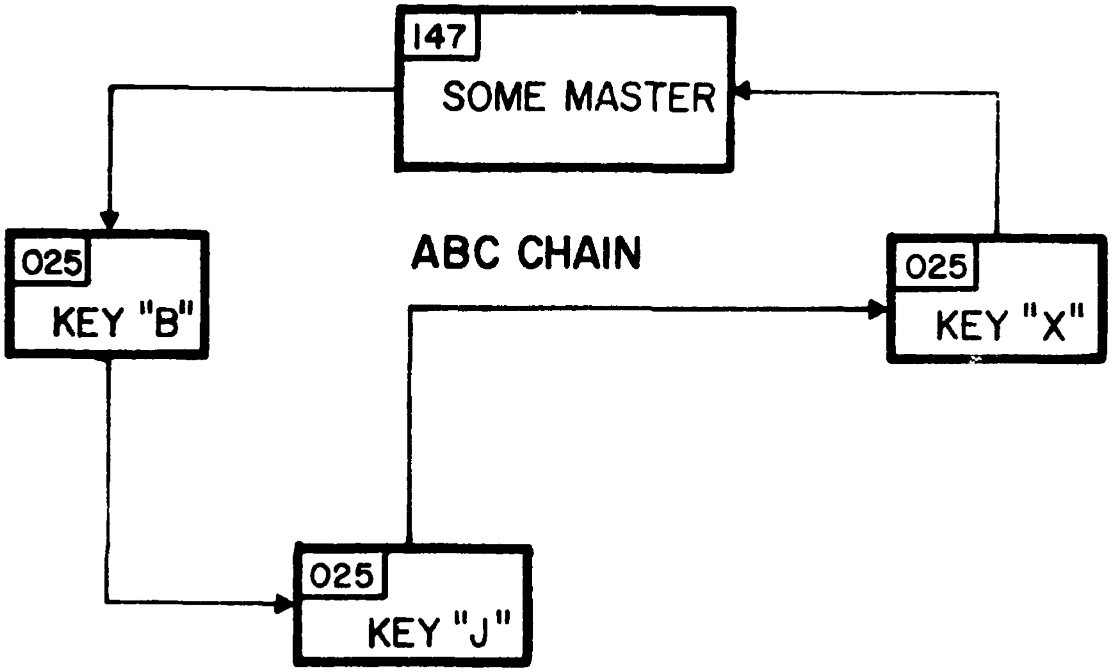
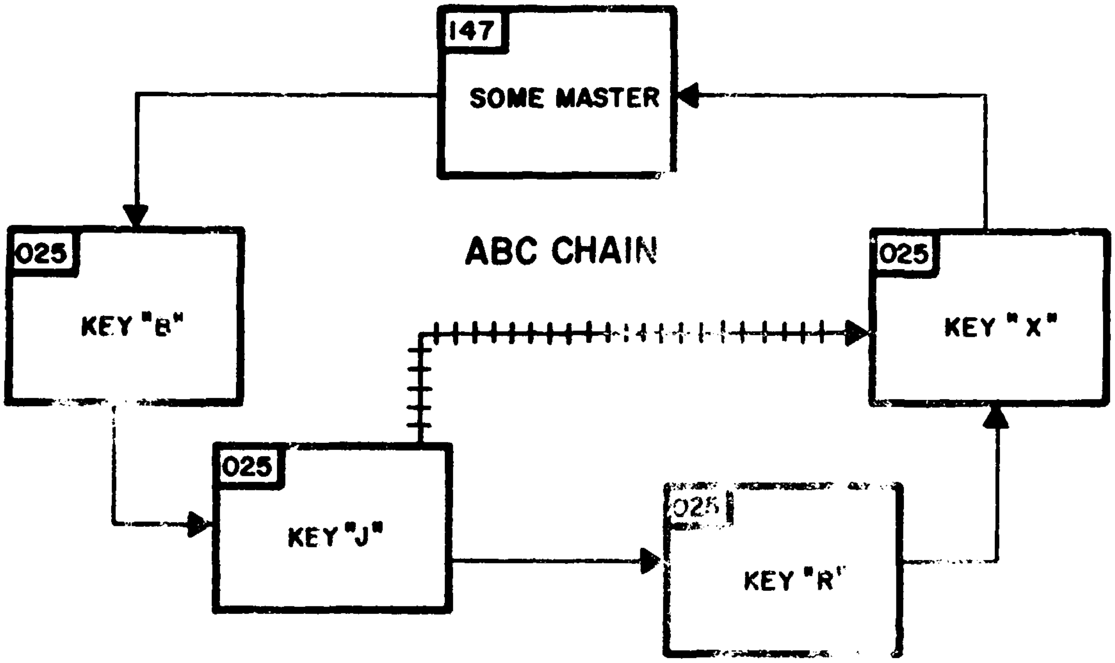


EXHIBIT IDS020-10

CHAIN-ORDER SORTED



CHARACTERISTICS

1. Ascending or descending sequence, one or many sequence keys.

2. The records are not physically, but logically sorted.

3. Can be resequenced on increasing values in one field and decreasing values in another field.

4. A record participating in several chains can be sequenced in many different orders.

EXHIBIT IDS020-11

CHAIN-ORDER SORTED WITHIN TYPE

THINK THIS WAY

1	025 TYPE RECORDS ASCENDING SEQUENCE
2	973 TYPE RECORDS DESCENDING SEQUENCE
3	STORING NEW 025 TYPE

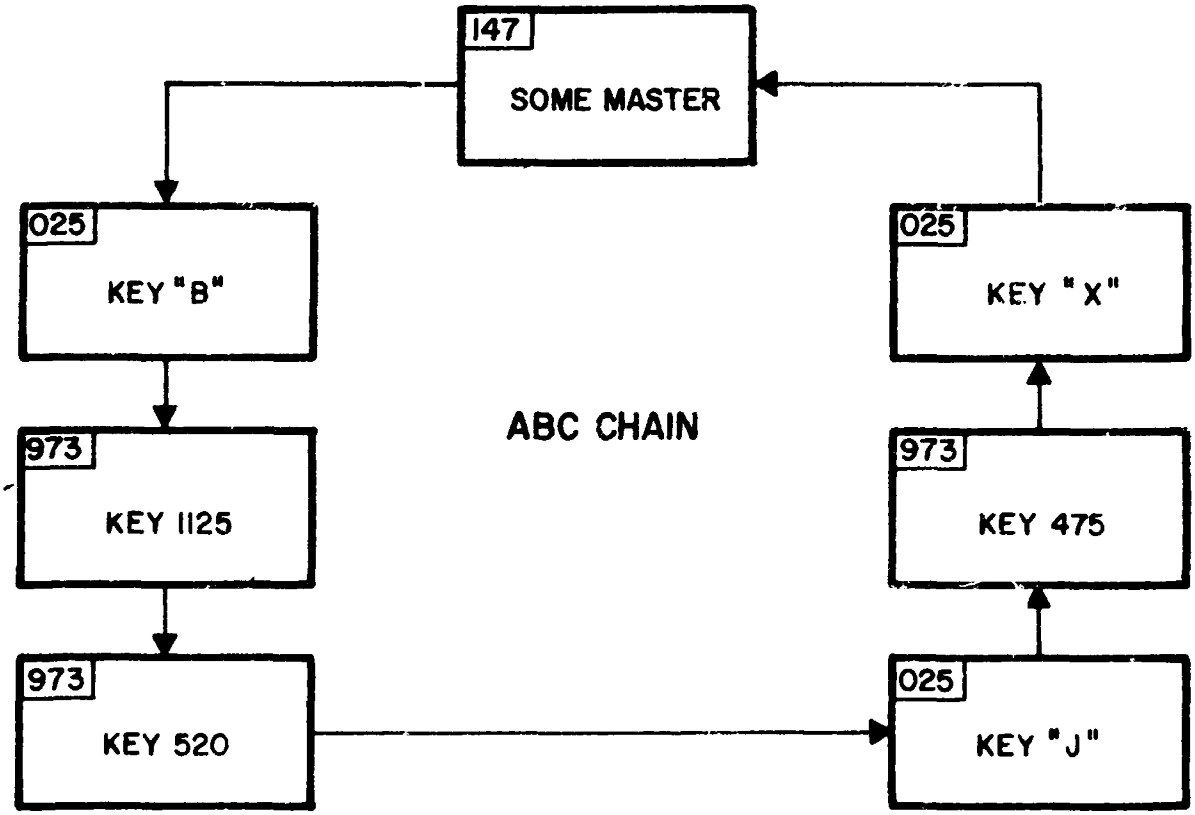
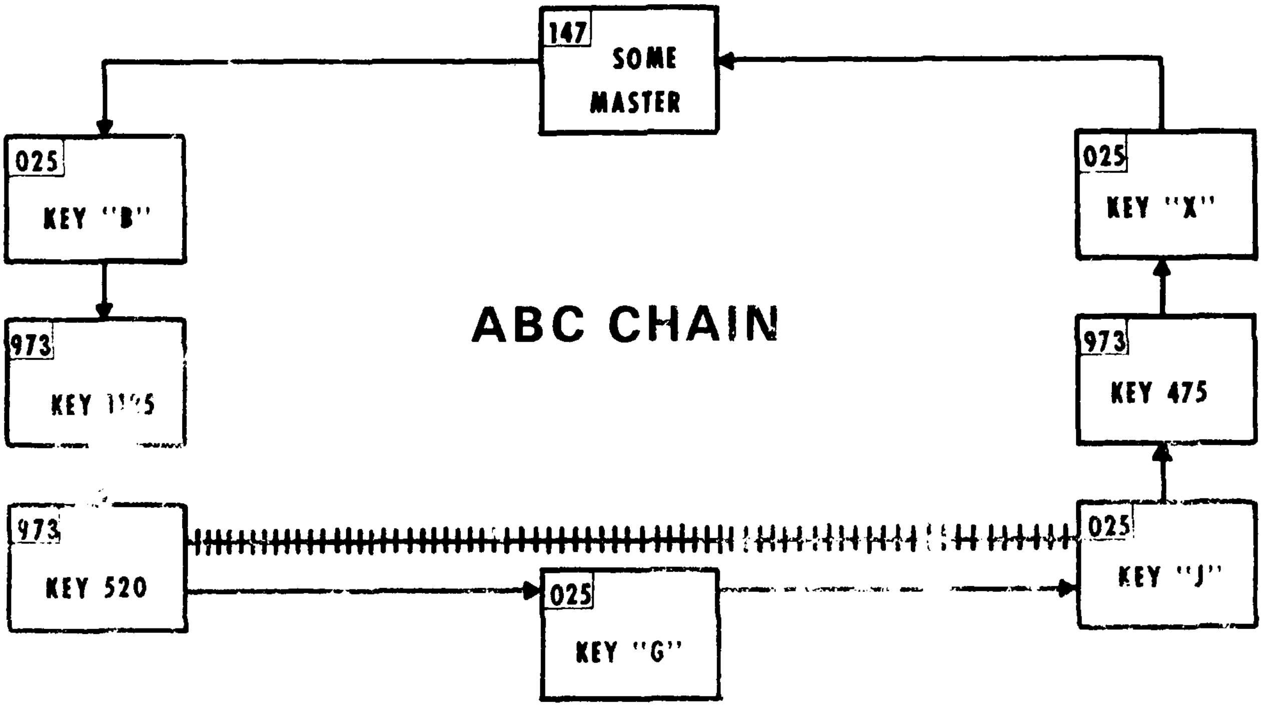


EXHIBIT IDS020-12

CHAIN-ORDER SORTED WITHIN TYPE

THINK THIS WAY

1	025 TYPE RECORDS ASCENDING SEQUENCE
2	973 TYPE RECORDS DESCENDING SEQUENCE
3	STORING NEW 025 TYPE



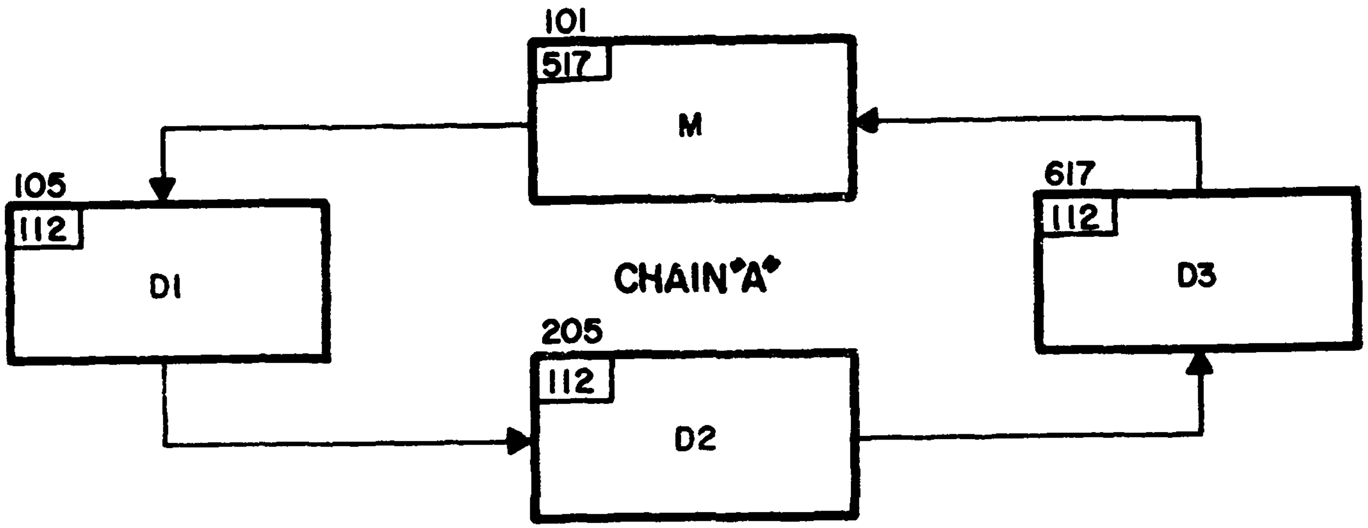
CHARACTERISTICS

1. Records maintained in sequence within type, independent of other types in that chain.

2. Within the chain details of different types are intermixed.

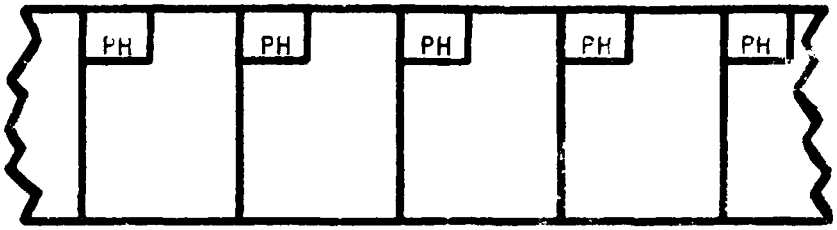
EXHIBIT IDS020-13

CHAIN TABLES

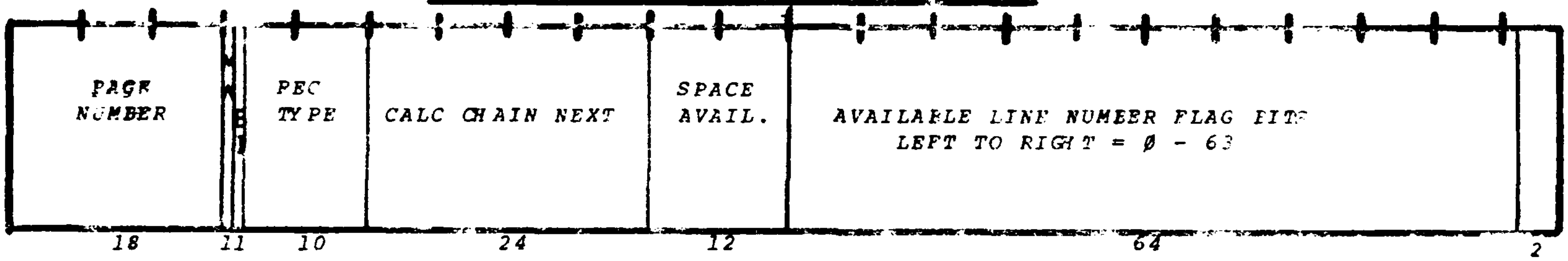


CHAIN*A* TABLE	
MASTER	
PRIOR	
CURRENT	
NEXT	

EXHIBIT IDS020-14



BASE PAGE HEADER FORMAT (22 CHARACTERS)

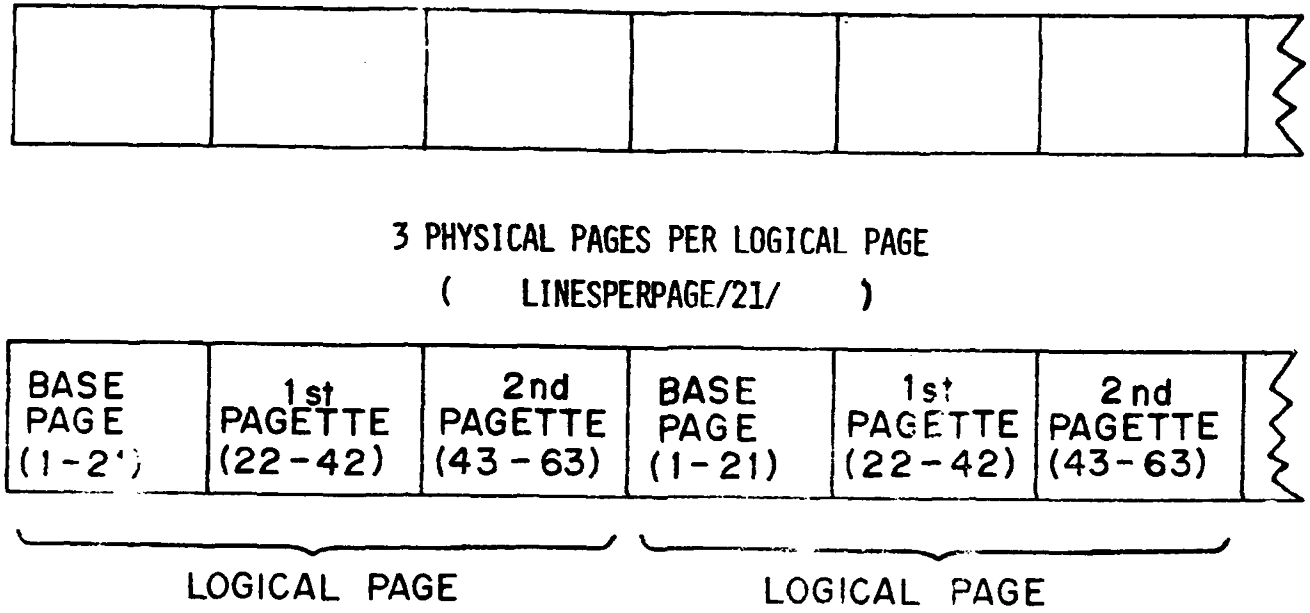


EXAMPLE OF .QUTD SHOWING PAGE HEADER FORMAT

OCTAL - 000003	3750	000003 01	2476	760000000000000000000
DECIMAL - 3	01 1000	pg 3 ln 1	1342	111110 ------------------------------------ 0 012345...

EXHIBIT IDS021-1

PAGES PER PAGE (PAGETTES)



PAGETTE HEADER (22 CHAR)

LOGICAL PAGE NUMBER	MW	BI	REC TYPE 1003 (10)	BEGINNING REF CODE	SPACE AVAILABLE	AVAILABLE LINE NUMBER FLAG BITS	MBZ
18	1	1	10	24	12	64	?

EXHIBIT IDS021-2

SUBFILING

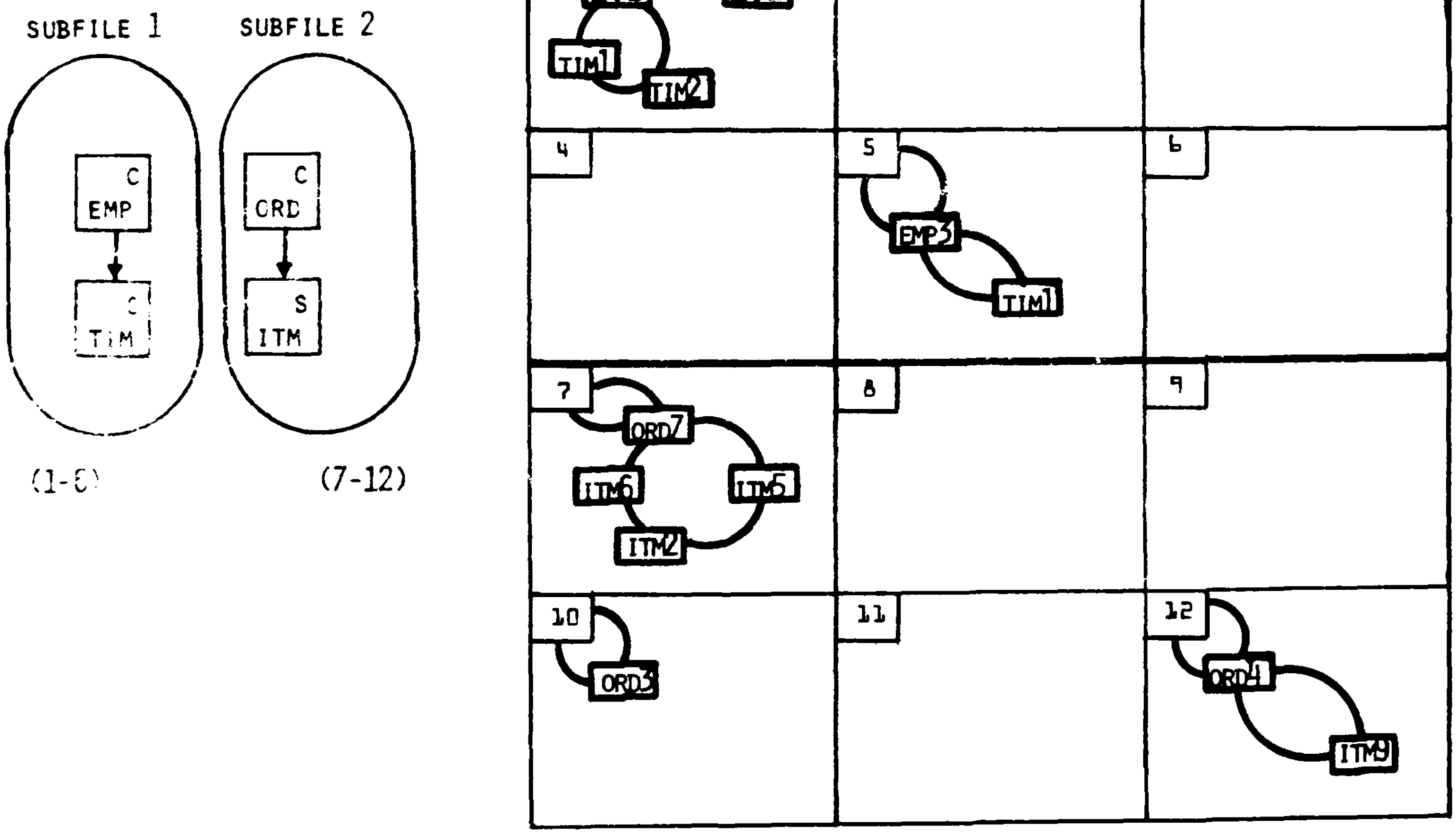


EXHIBIT IDS021-3

IDS SUBFILE/DEVICE RELATIONSHIP

(1) May occupy part or all of a physical device.

(2) Is represented by a JCL card as a permanent or temporary file or both.

(3) May contain any mix of IDS records.

(4) Is not bound by any IDS page ranges.

EXHIBIT IDS021-4

I-D-S FILE/SUBFILE

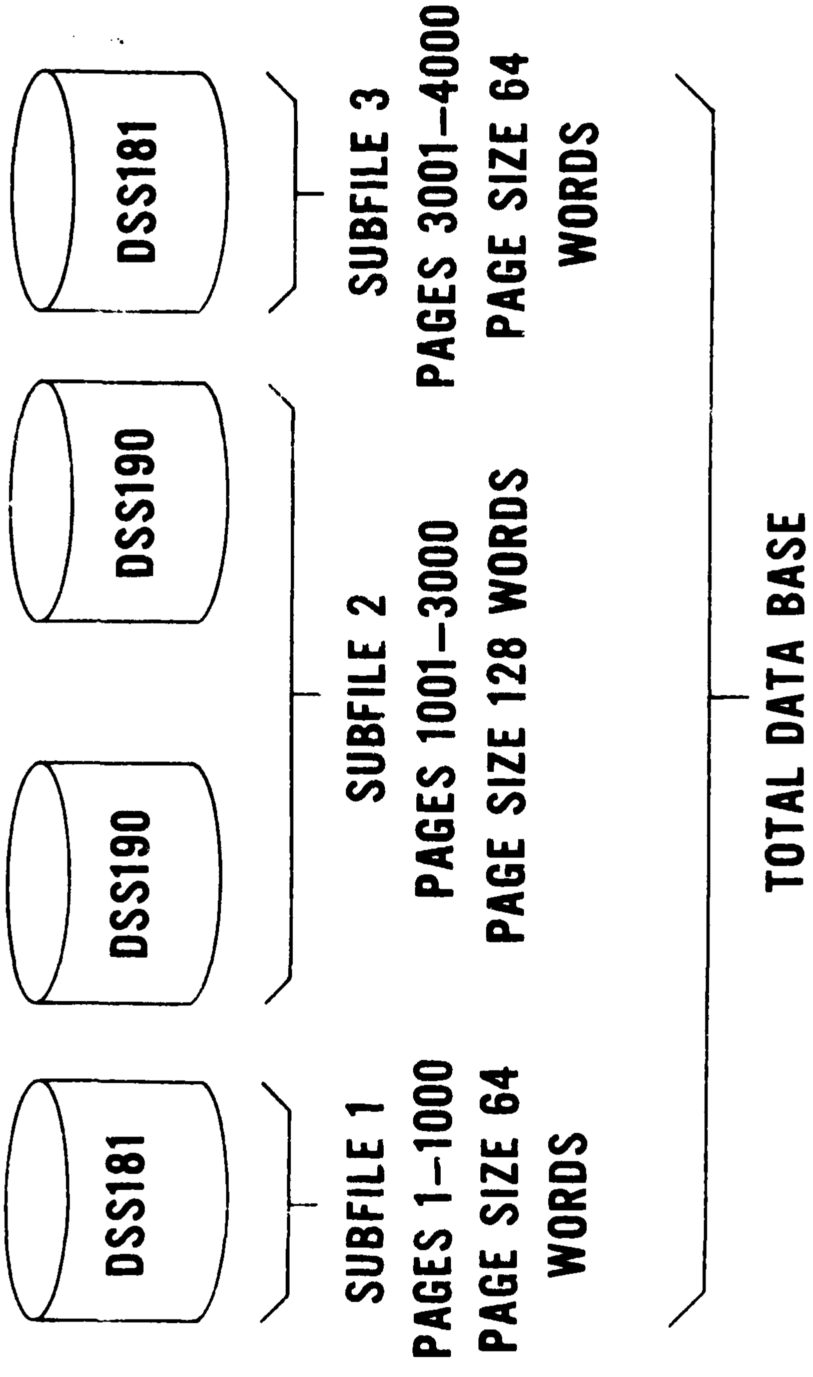
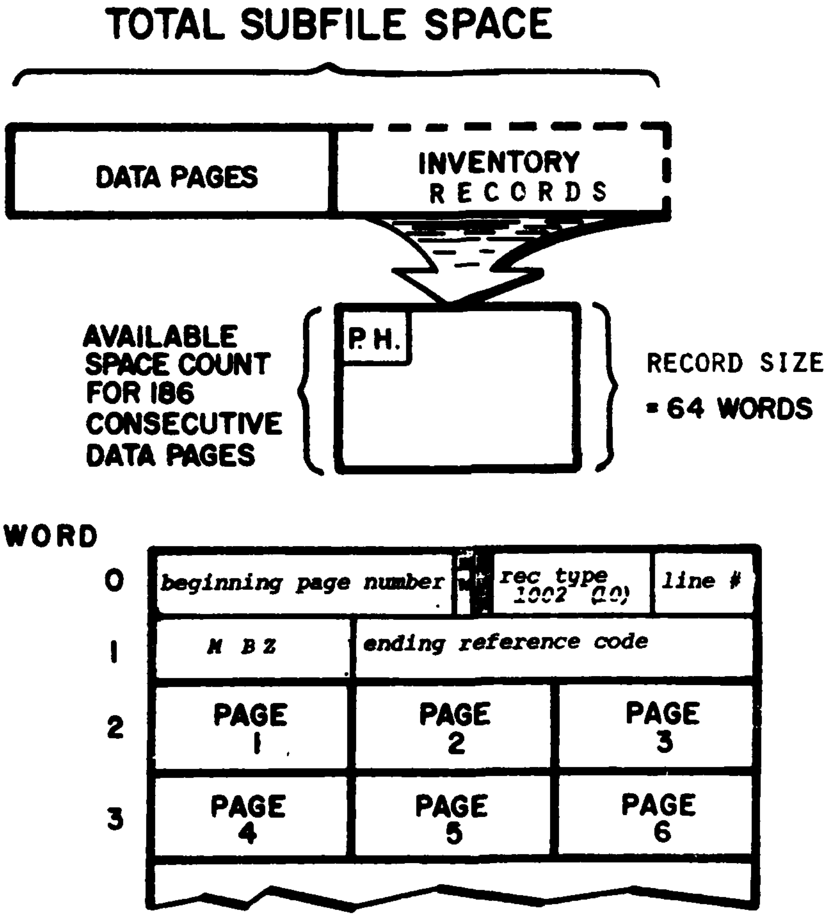


EXHIBIT IDS021-5

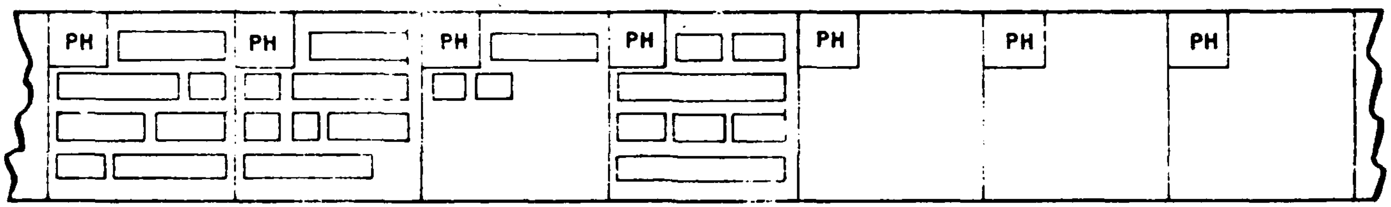
INVENTORY CONTROL



WORD			
0	*beginning page number*	*rec type* *1002 (10)*	*line #*
1	*M B Z*	*ending reference code*	
2	PAGE 1	PAGE 2	PAGE 3
3	PAGE 4	PAGE 5	PAGE 6

EXHIBIT IDS021-6

DATA RECORDS WITHIN A FILE



DATA RECORD FORMATS

CALC	HDR.	CALC NEXT	USER DATA	LOGICAL CHAIN POINTERS
NO. OF CHAR.	5	4	?	@ 4 CHAR. EACH

PRIMARY SECONDARY	HDR.	USER DATA	LOGICAL CHAIN POINTERS
NO. OF CHAR.	5	?	@ 4 CHAR. EACH

DATA RECORD HEADER

	ASSIGNED LINE NUMBER	D S	RECORD TYPE (USER DEFINED)	PHYSICAL RECORD SIZE
NO. OF BITS	6	1	11	12

EXHIBIT IDS021-7

I-D-S PAGING

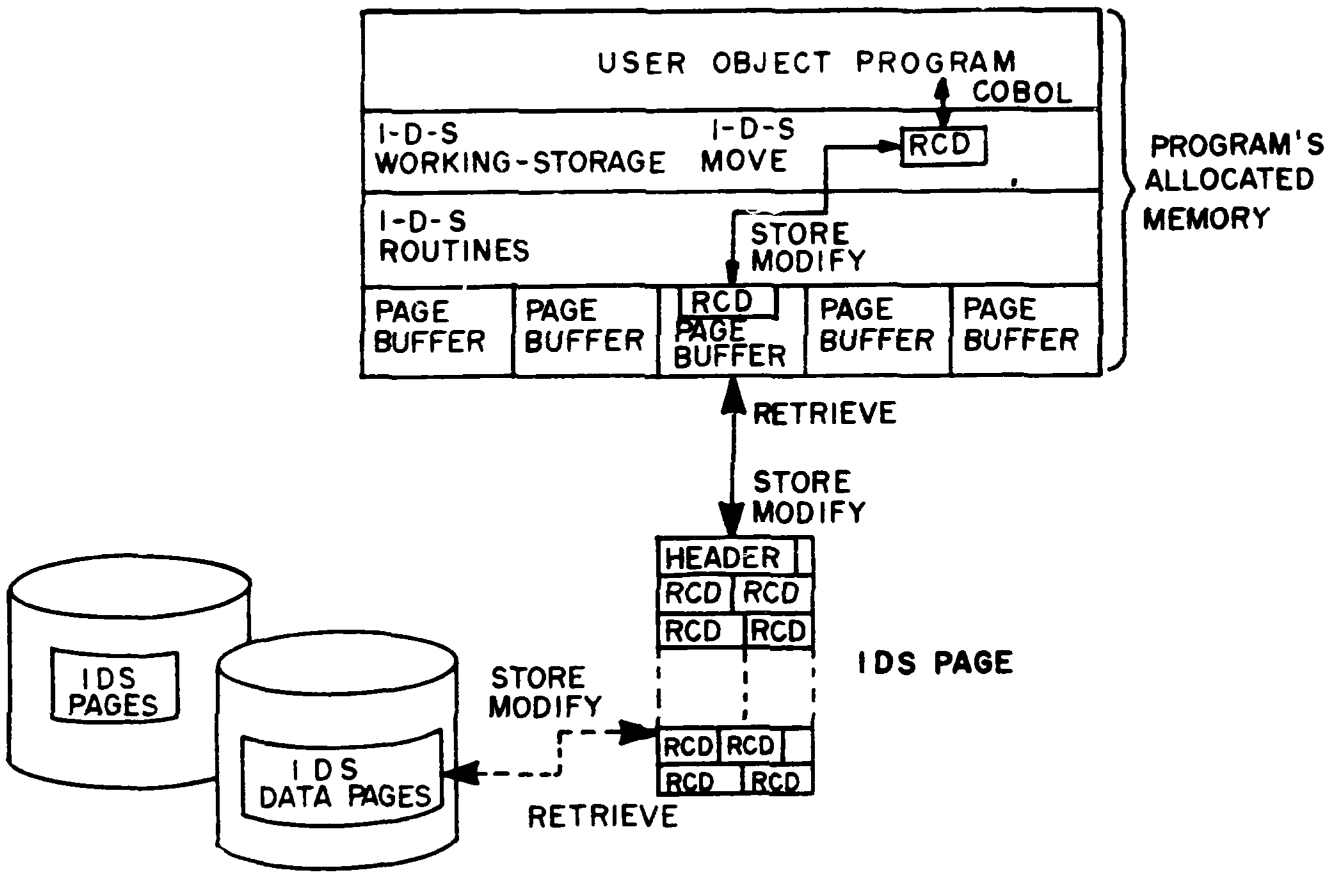


EXHIBIT IDS021-8

IDS PROGRAM SKELETON

```
ENVIRONMENT DIVISION
CONFIGURATION SECTION.
SOURCE-COMPUTER. ...
OBJECT-COMPUTER. ... }           as in COBOL
SPECIAL-NAMES.   ...
IDS-SPECIAL-NAMES.
     .
     .   ... IDS Special Names Syntax
     .
INPUT-OUTPUT SECTION.
FILE-CONTROL.
       SELECT IDS ...}           IDS Syntax
       SELECT ...
           .
           .         }           as in COBOL
           .

I-O-CONTROL
     .
     .
     .
```

EXHIBIT IDS030-1

CCBLOXK - COMMUNICATION AREA

```
IDS SECTION.
    01    CCBLOXK.                                          )
                                                            |
    02    DIRECT-REFERENCE                                  |
          SIZE 10 USAGE COMP-3 SYNC RIGHT.                  |   Not user defined
                                                            |
    02    FIRST-REFERENCE                                   |
          SIZE 10 USAGE COMP-3 SYNC RIGHT.                  |
                                                            |
    02    LAST-REFERENCE                                    |
          SIZE 10 USAGE COMP-3 SYNC RIGHT.                  |
                                                            >   Generated by IDS
    02    RECORD-TYPE                                       |   translator
          SIZE 4 USAGE COMP-1 SYNC RIGHT.                   |
                                                            |
    02    REC-FILE                                          |
          SIZE 6 CLASS IS ALPHANUMERIC                      |   User accessible in
          VALUE IS "ØØØØFC".                                |   IDS and COBOL
                                                            |
    02    ERROR-REFERENCE                                   |
          SIZE 3 CLASS IS ALPHANUMERIC SYNC RIGHT.          )

MD  DATA-BASE.
```

EXHIBIT IDS030-2

SAMPLE RECORD DESCRIPTION

```
CARD COLUMN
 8   12                  32
 01  SALESMAN                       TYPE IS 020
                                    RETRIEVAL VIA CALC CHAIN
                                    PAGE-RANGE IS 100 TO 900.

     02  NAME            PIC X(30).
     02  SSAN            PIC 9(9).
     02  FILLER          PIC X(40).
     02  LOCATION        SIZE 42.
         03  ADDRESS.
             05  NUMBER  PIC 9(5).
             05  STREET  PIC X(20).
         03  CITY        PIC X(10).
         03  STATE       PIC XX.
         03  ZIPCODE     PIC 9(5).

     98  SALESMAN-SALES CHAIN MASTER
         CHAIN-ORDER IS AFTER.

     98  DISTRICT-SALESMAN CHAIN DETAIL
         SELECT CURRENT MASTER
         LINKED TO MASTER.

     98  CALC CHAIN DETAIL
         RANDOMIZE ON NAME
         RANDOMIZE ON SSAN.
```

EXHIBIT IDS030-3

IDS RECORD DESCRIPTION ENTRY

```
01  record-name  TYPE   IS integer-1

    RETRIEVAL  VIA  { field-name FIELD   }
                    { chain-name-1 CHAIN }
                    { CALC CHAIN         }

   [ PAGE-RANGE IS { integer-2 TO integer-3     } ]
   [               {                            } ]
   [               { data-name-1 TO data-name-2 } ]

   [PLACE NEAR chain-name-2 CHAIN] [INTERVAL IS integer-4 PAGES]

   [AUTHORITY IS integer-5] [ WITHIN AREA  { integer-6   } ] .
                            [              {             } ]
                            [              { data-name-3 } ]
```

EXHIBIT IDS031-1

PAGE-RANGE

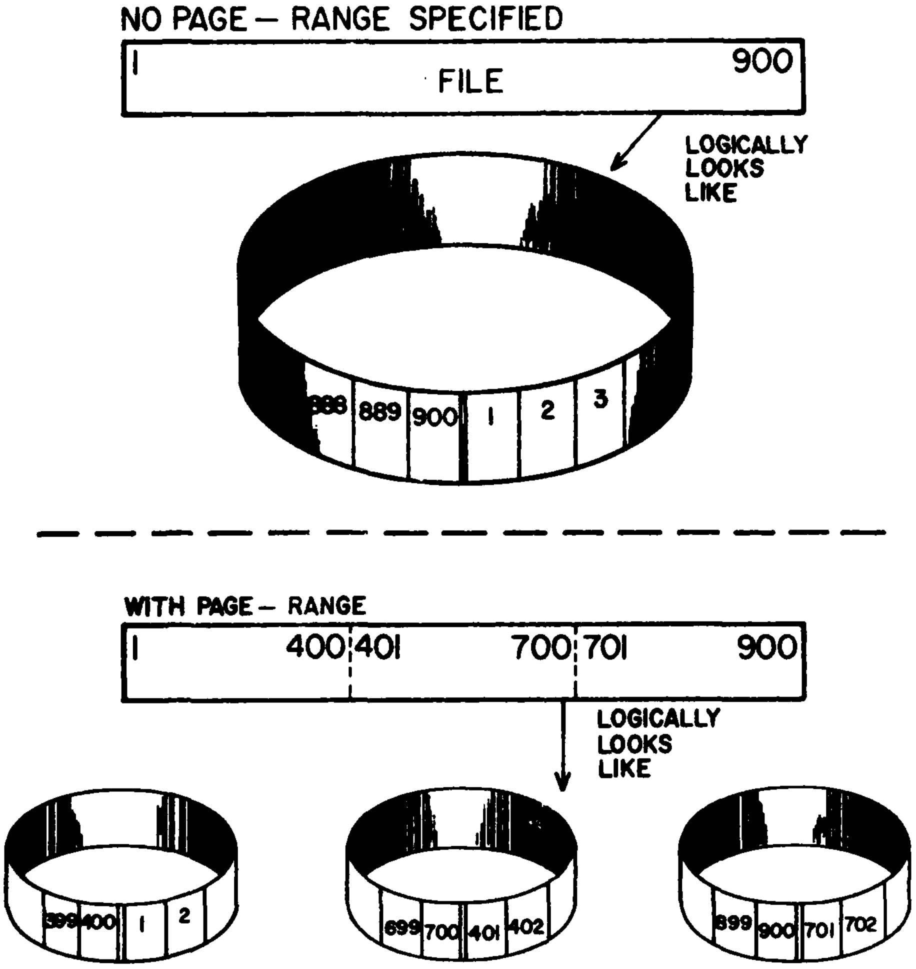


EXHIBIT IDS031-2

IDS RECORD DESCRIPTION ENTRY
(PRIMARY RECORD)

$$\text{01 record-name } \underline{\text{TYPE}} \text{ IS integer-1}$$

$$\underline{\text{RETRIEVAL}} \text{ VIA field-name } \underline{\text{FIELD}}$$

$$\left[\underline{\text{PAGE}}\text{-}\underline{\text{RANGE}} \text{ IS } \left\{\begin{matrix}\text{integer-2 TO integer-3}\\ \text{data-name-1 TO data-name-2}\end{matrix}\right\}\right]$$

$$[\underline{\text{PLACE}} \text{ NEAR chain-name-2 CHAIN}]$$

$$[\underline{\text{INTERVAL}} \text{ IS integer-4 PAGES}]$$

$$[\underline{\text{AUTHORITY}} \text{ IS integer-5}]$$

$$\left[\underline{\text{WITHIN}} \text{ AREA } \left\{\begin{matrix}\text{integer-6}\\ \text{data-name-3}\end{matrix}\right\}\right].$$

NOTE: Field-name must be described as an 02 level in this record with a description of 9(8) (or 9(10) for multi-area applications).

EXHIBIT IDS031-3

IDS RECORD DESCRIPTION ENTRY
(CALCULATED RECORD)

```
01  record-name TYPE IS integer-1

    RETRIEVAL VIA CALC CHAIN

   [                  { integer-2 TO integer-3     } ]
   [ PAGE-RANGE IS    {                            } ]
   [                  { data-name-1 TO data-name-2 } ]

   [AUTHORITY IS integer-5]

   [                      { integer-6   } ]
   [ WITHIN   AREA        {             } ] .
   [                      { data-name-3 } ]
```

EXHIBIT IDS031-4

IDS RECORD DESCRIPTION ENTRY
SECONDARY RECORD

```
01  record-name TYPE IS integer-1

    RETRIEVAL VIA chain-name-1 CHAIN

                      integer-2 TO integer-3
    PAGE-RANGE IS
                      data-name-1 TO data-name-2

    [PLACE NEAR chain-name-2 CHAIN]

    [INTERVAL IS integer-4 PAGES]

    [AUTHORITY IS integer-5]

   [                 { integer-6   } ]
   [ WITHIN AREA     {             } ] .
   [                 { data-name-3 } ]
```

NOTE: Chain-name-1 must be a chain this record is a detail on.

EXHIBIT IDS031-5

IDS 02 LEVEL CODING

```
01  NAME-REC  TYPE IS ...
              RETRIEVAL VIA ...
    02  RANDOMIZE-FLD       PIC X(10).
    02  SORT-KEY-FLD        PIC X(8).
    02  SORT-KEY-FLD-2      PIC 9(7).
    02  SSAN                SIZE 9.
        03  SS-1            PIC XXX.
        03  SS-1            PIC XX.
        03  SS-3            PIC X(4).
    02  FILLER              PIC X(7).
    02  OTHER-DATA-3        SIZE 30.
        03  COBOL-FIELD     PIC 9(15).
        03  COUNTER-1       PIC 9(5).
        03  INDEX-3         PIC 9(10).
                .
                .
                .
```

EXHIBIT IDS032-1

CHAIN DESCRIPTION ENTRY FOR A MASTER OF A USER DEFINED CHAIN

```
98  chain-name  CHAIN  MASTER

                       ( FIRST               )
                       | LAST                |
         CHAIN-ORDER IS< BEFORE              >
                       | AFTER               |
                       | SORTED              |
                       ( SORTED WITHIN TYPE  )

    [LINKED TO PRIOR].
```

EXHIBIT IDS032-2

CHAIN DESCRIPTION ENTRY FOR A DETAIL
OF THE CALCULATION CHAIN

```
98  CALC CHAIN DETAIL

    RANDOMIZE ON field-name-1

    [RANDOMIZE . . .]

    [RANDOMIZE . . .] .
```

EXHIBIT IDS032-3

CHAIN DESCRIPTION ENTRY FOR A DETAIL
OF A USER CHAIN

```
                                               ┌ ARE FIRST   ┐
98  chain-name-2  CHAIN DETAIL  [ DUPLICATES  {  ARE LAST     } ]
                                               └ NOT ALLOWED ┘

{ ASCENDING  }                    [ { ASCENDING   ... }       ]  *
{ DESCENDING }   KEY IS field-name-2 { DESCENDING      }  ...  ]

      [ASCENDING RANGE KEY IS field-name-3] ]

{ SELECT CURRENT MASTER **                          }
{ SELECT UNIQUE MASTER MATCH-KEY IS field-name-4    }

      [MATCH-KEY ...] ...

      [MATCH-KEY IS  [ field-name-5   { SYNONYM } ]  field-name-6]
                                      { SYN     }

[LINKED TO MASTER].
```

*Must be used with SORTED CHAINS.
**Must be used with chain-orders "BEFORE" and "AFTER".

EXHIBIT IDS032-4

SELECT CURRENT

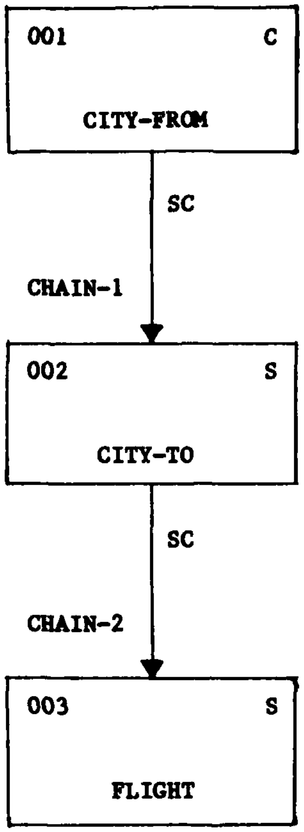


EXHIBIT IDS032-5

INITIAL FILE LOAD

1.	STORE "ATLANTA"	(001)
2.	STORE "BOSTON"	(002)
3.	STORE "176"	(003)
4.	STORE "202"	(003)
5.	STORE "CHICAGO"	(002)
6.	STORE "206"	(003)
7.	STORE "143"	(003)

	CURRENT OF TYPE		
	001	002	003
1	ATLANTA	0	0
2	ATLANTA	BOSTON	0
3	ATLANTA	BOSTON	176
4	ATLANTA	BOSTON	202
5	ATLANTA	CHICAGO	202
6	ATLANTA	CHICAGO	206
7	ATLANTA	CHICAGO	143

SELECT CURRENT
(LONGHAND)

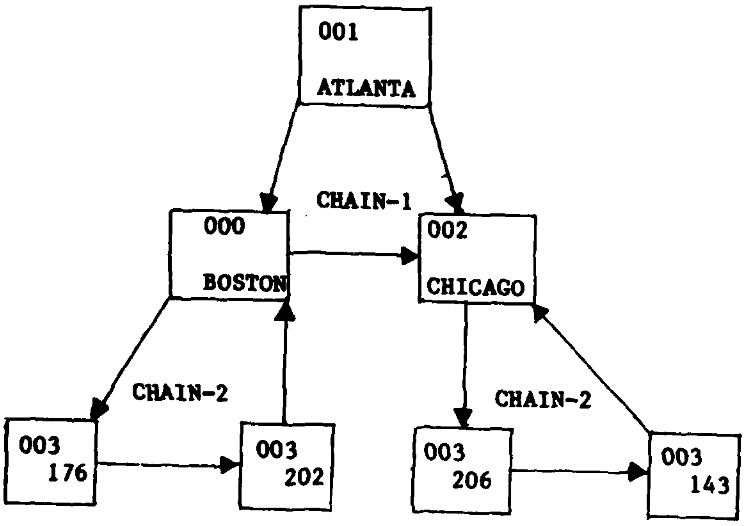


EXHIBIT IDS032-6

ENTER IDS

```
ENTER IDS.  IDS statement-1
     [IDS statement-2] ...
     [IDS statement-n].
```

EXHIBIT 040-1

DATA BASE MAINTENANCE AND RECOVERY

1. OPEN
2. CLOSE
3. SET
4. CHECKPOINT
5. ROLLBACK

EXHIBIT IDS040-2

OPEN

ENTER IDS.

```
     ⎡      ⎧RETRIEVAL⎫⎤
OPEN ⎢ FOR  ⎨         ⎬⎥
     ⎣      ⎩UPDATE   ⎭⎦
            [WITH AUTHORITY-KEY integer-1].
```

EXHIBIT IDS040-3

CLOSE

ENTER IDS.

```
    CLOSE [WITH LOCK]
```

EXHIBIT IDS040-4

DATABASE PROTECTION FEATURES

ENTER IDS.

```
⎧CHECKPOINT FILES [AND PROGRAM]⎫
⎩ROLLBACK FILES [AND PROGRAM]  ⎭ .
```

EXHIBIT IDS040-5

SET

ENTER IDS.

SET AREA TO field-name-1.

EXHIBIT IDS040-6

DATABASE UPDATES

1. STORE
2. MODIFY
3. DELETE
4. SORT
5. RETURN

EXHIBIT IDS040-7

STORE

ENTER IDS.

STORE record-name-1 RECORD.

EXHIBIT IDS040-8

STORE EXAMPLE

```
DATA DIVISION.
IDS SECTION.
01  RECORD-A                          TYPE IS 010
                                      RETRIEVAL VIA CALC CHAIN ...

    02  CALC-KEY                      PIC 9(9).
    02  DATA-FLD-1                    PIC 9(4).
    02  DATA-FLD-2                    SIZE 13.
        03  SUB-FIELD-1               PIC X(8).
        03  SUB-FIELD-2               PIC 9(5).

    98  CALC CHAIN DETAIL
        RANDOMIZE ON CALC-KEY.
                .
                .
                .
PROCEDURE DIVISION.
    ENTER IDS.  OPEN FOR UPDATE.
                .
                .
                .
    MOVE INPUT-CALC-KEY TO  CALC-KEY.
    MOVE INPUT-DATA-1   TO  DATA-FLD-1.
    MOVE INPUT-SUB-1    TO  SUB-FIELD-1.
    MOVE INPUT-SUB-2    TO  SUB-FIELD-2.

    ENTER IDS.
        STORE RECORD-A RECORD IF ERROR ...
               .
               .
               .
    ENTER IDS.
        CLOSE.
```

EXHIBIT IDS040-9

MODIFY

```
ENTER IDS.

     MODIFY field-name-1  [field-name-2...].  OPTION 1

ENTER IDS.

     MODIFY CURRENT record-name

     [field-name-1  [field-name-2 ...]].      OPTION 2
```

EXHIBIT IDS040-10

SUGGESTED SEQUENCE FOR MODIFYING A RECORD

1. Retrieve record to be modified.
2. Move retrieved record's field(s) to WORKING-STORAGE.
3. COBOL MOVE the new information into the field(s) to be modified.
4. PERFORM the MODIFY verb.

EXHIBIT IDS040-11

MODIFY EXAMPLE

```
DATA DIVISION.
IDS SECTION.
01  RECORD-B                    TYPE IS 020
                                RETRIEVAL VIA CALC CHAIN...

    02  CALC-KEY                PIC X(5).

    98  CALC CHAIN DETAIL
        RANDOMIZE ON CALC-KEY
                .
                .
                .
PROCEDURE DIVISION.
    ENTER IDS.  OPEN FOR UPDATE.
                    .
                    .
                    .
    MOVE OLD-KEY-VAL TO CALC-KEY.

    ENTER IDS.
        (1) RETRIEVE RECORD-B RECORD IF ERROR...
                ELSE (2) MOVE TO WORKING-STORAGE.
    (3)  MOVE NEW-KEY-VAL TO CALC-KEY.

    ENTER IDS.
        (4) MODIFY CURRENT RECORD-B CALC-KEY IF ERROR ...
                        .
                        .
                        .

    ENTER IDS.  CLOSE.
```

EXHIBIT IDS040-12

DELETE

ENTER IDS.

```
    DELETE  ┌ CURRENT record-name-1 RECORDS
            └ ON record-name-2 DETAIL

                [MOVE TO WORKING-STORAGE]

               ┌[HEAD chain-name-1 CHAIN [HEAD]]┐
               │[PERFORM procedure-name-1]      │
               └[GO TO procedure-name-2]        ┘

        [ { OTHERWISE }  ON record-name-3 DETAIL ...] ...] .
          { ELSE      }
```

EXHIBIT IDS040-13

SUGGESTED SEQUENCE FOR DELETING A RECORD

1. Retrieve the record to be deleted.

2. PERFORM the DELETE verb.

EXHIBIT IDS040-14

LOGICAL DELETION

1. Logical deletion occurs whenever

   a. a DELETE verb is executed.

   b. an IDS error occurs on an IDS STORE.

2. Delete switch indicates to IDS that at least one chain pointer contains a value.

3. A record is delinked from a chain anytime the following conditions are met:

   a. Delete switch is set.

   b. Data base has been opened for update.

   c. IDS has PRIOR pointer in the chain table.

EXHIBIT IDS040-15

SORT

```
ENTER IDS.

      SORT sort-field-1 ON  { ASCENDING  }  KEY
                            { DESCENDING }

          field-name-1 [,field-name-2]...
         [ ON  { ASCENDING  }  KEY ... ]...
         [     { DESCENDING }          ]

        { INPUT PROCEDURE IS section-name-1 }
        {      [THRU section-name-2]        }
        { USING file-name-2                 }

        { OUTPUT PROCEDURE IS section-name-3 }
        {      [THRU section-name-4]         }
        { GIVING chain-name-1 CHAIN          }.
```

EXHIBIT IDS041-1

RETURN

```
ENTER IDS.

              RETURN chain-name-1 CHAIN

              AT END GO TO procedure-name-1.
```

EXHIBIT IDS041-2

DATA BASE RETRIEVAL

1. RETRIEVE

2. MOVE

3. HEAD

EXHIBIT IDS041-3

RETRIEVE

ENTER IDS.

$$\underline{\text{RETRIEVE}} \left\{ \begin{array}{l} \left\{\begin{array}{l}\text{record-name-1}\\ \underline{\text{CURRENT}}\ \text{record-name-1}\end{array}\right\} \text{ RECORD} \\ \\ \left\{\begin{array}{l}\underline{\text{NEXT}}\\ \underline{\text{PRIOR}}\end{array}\right\} \text{ RECORD OF} \left\{\begin{array}{l}\text{chain-name-2}\\ \underline{\text{CALC}}\end{array}\right\} \text{CHAIN} \\ \\ \underline{\text{MASTER}}\ \text{RECORD OF chain-name-2 CHAIN} \\ \underline{\text{EACH}}\ \text{AT}\ \underline{\text{END}}\ \underline{\text{GO}}\ \underline{\text{TO}}\ \text{procedure-1} \\ \underline{\text{DIRECT}} \end{array} \right\}$$

EXHIBIT IDS041-4

RETRIEVE EXAMPLE

```
DATA DIVISION.
IDS SECTION.
01  RECORD-D                      TYPE IS 040
                                  RETRIEVAL VIA CALC CHAIN ...

    02  CALC-KEY                  PIC X(33)
    98  A-B CHAIN MASTER ...
    98  CALC CHAIN DETAIL...
              .
              .
              .
PROCEDURE DIVISION.
    ENTER IDS.  OPEN FOR RETRIEVAL.
                     .
                     .
                     .
    MOVE KEY-VAL TO CALC-KEY.
    ENTER IDS.
        RETRIEVE RECORD-D IF ERROR...
        ELSE MOVE TO WORKING-STORAGE.
    PERFORM MASTER-INFO-COLLECTION.
CHAIN-WALK.
    ENTER IDS.
        RETRIEVE NEXT OF A-B CHAIN IF ERROR...ELSE
        IF RECORD-D RECORD GO TO END-CHAIN-WALK ELSE MOVE.
    PERFORM DETAIL-COLLECTION.
    GO TO CHAIN-WALK.
END-CHAIN-WALK.
    PERFORM WRITE-MASTER-DETAIL-INFO.
    ENTER IDS.  CLOSE.
```

EXHIBIT IDS041-5

MOVE

ENTER IDS.

$\underline{\text{MOVE}}$ TO WORKING-STORAGE $\left[\text{field-name-1 [,field-name-2]...}\right]$.

ENTER IDS.

$$\underline{\text{MOVE}}\ \text{chain-name-1}\left\{\begin{array}{l}\underline{\text{CHAIN}}\ \underline{\text{TABLE}}\\ \underline{\text{MASTER}}\\ \underline{\text{PRIOR}}\\ \underline{\text{CURRENT}}\\ \underline{\text{NEXT}}\end{array}\right\}\ \text{TO field-name.}$$

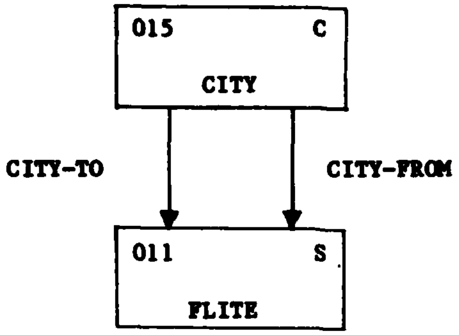
EXHIBIT IDS041-6

HEAD

ENTER IDS.

$\underline{\text{HEAD}}$ chain-name-1 CHAIN [$\underline{\text{HEAD}}$ chain-name-2...]... .

EXHIBIT IDS041-7



After retrieving a "FLITE" record thru the "TO" chain, you could;

a. "RETRIEVE" the "CITY" master record of "CITY-FROM" chain, which would update both the "CITY-TO" and "CITY-FROM" chain tables.

   This would cause the same "CITY" master to be the same master entry in both chain tables, thus losing the "CITY-TO" chain master first found.

   OR

b. "HEAD" the "CITY-FROM" chain, which would result in only the "CITY-FROM" chain table being updated. The "CITY-TO" chain table is not affected and still has the first found city as its master entry.

EXHIBIT 041-8

MISCELLANEOUS

1. GO TO

2. PERFORM

3. IF

4. DEBUG

5. USE

EXHIBIT IDS041-9

TRANSFER OF CONTROL

```
ENTER IDS.

          GO TO procedure-name-1.

ENTER IDS.

          PERFORM procedure-name-2

             [THRU procedure-name 3].
```

EXHIBIT IDS041-10

DETECT RECORD TYPE

```
ENTER IDS.

     IF record-name RECORD statement-1 [,statement-2]...

    ┌ ┌ ELSE      ┐                                  ┐
    │ │           │  statement-3 [statement-4] ...   │ .
    └ └ OTHERWISE ┘                                  ┘

                      ┌ MOVE ...    ┐
                      │ MODIFY ...  │
statement-1,-2,-3,-4: │ DELETE ...  │
                      │ HEAD ...    │
                      │ PERFORM     │
                      └ GO TO       ┘

ALSO statement-3:  IF ...
```

EXHIBIT IDS041-11

DETECT DATA DEPENDENT ERRORS

```
ENTER IDS.

          IF ERROR  { GO TO procedure-name-1   }
                    { PERFORM procedure-name-1 }

        [ { ELSE      }  statement-1 [-statement-2] ... ] .
        [ { OTHERWISE }                                 ]

                   { GO TO ...                      }
                   { PERFORM ...                    }
statement-1,-2:    { IF record-name-1 ...           }
                   { database manipulation verb     }
```

EXHIBIT IDS041-12

DATA MANIPULATION EXAMPLE

NOTE: When a data manipulation verb is used as statement-1 or -2 of option 2, it cannot be followed by a second "IF ERROR" within that statement. The second error check statement must be a separate statement.

EXAMPLE: ENTER IDS.

```
ILLEGAL:  RETRIEVE SALESMAN RECORD IF ERROR GO TO ... ELSE RETRIEVE NEXT OF
          SM-SAL CHAIN IF ERROR GO TO ... ELSE ...

LEGAL:    ENTER IDS.
          RETRIEVE SALESMAN RECORD IF ERROR GO TO ... ELSE RETRIEVE NEXT OF
          SM-SAL CHAIN.

          ENTER IDS.
          IF ERROR GO TO ... ELSE ...
```

EXHIBIT IDS041-13

IDS DATA DEPENDENT ERRORS

1. IDS returns reason code in ERROR-REFERENCE.

2. IDS restores linkages to former values, if possible.

3. IDS deletes record if error was detected during store.

4. Returns control to user.

EXHIBIT IDS041-14

DEBUG

ENTER IDS.

$$\underline{\text{DEBUG}}\ \text{CURRENT}\left[\left\{\begin{array}{l}\underline{\text{BUFFER}}\\ \underline{\text{RECORD}}\\ \underline{\text{CCBLOCK}}\end{array}\right\}\ ;\left[\left\{\begin{array}{l}\underline{\text{BUFFER}}\\ \underline{\text{RECORD}}\\ \underline{\text{CCBLOC}}\end{array}\right\}\right]\right]$$

[chain-name-2 CHAIN]

[$\underline{\text{TRACE}}$ chain-name-3 CHAIN]

EXHIBIT IDS041-15

USE

```
ENTER IDS.

     USE procedure-name-1 [THRU procedure-name-2]

         [WITH TRACE]
            ⎧ error-code-1  [error-code-2 ...] ⎫
         ON ⎨                                  ⎬ .
            ⎩ ANY ABORT                        ⎭
```

EXHIBIT IDS041-16

TRANSACTION TYPES

1. STORE
2. RETRIEVE
3. MODIFY
4. DELETE

EXHIBIT IDS042-1

PARTS OF A TYPICAL TRANSACTION RECORD

A. Type of transaction

B. Key fields

(1) Match Keys
or
Keys to set current of type

(2) RANDOMIZE keys

(3) SORT keys

C. Other data

KEYWORD: LANGLEY AFB

EXHIBIT IDS042-2

IDS STORE TRANSACTION

1. Determine the physical location of the record.

2. Determine the logical location (splice point) of the record.

EXHIBIT IDS042-3

DETERMINING THE LOGICAL SPLICE POINT
--CASE 1--

CHAIN ORDER	When SELECT UNIQUE IS specified, the user must:
FIRST LAST	Initialize match-keys within the master records.
SORTED SORTED WITHIN TYPE	Initialize SORT keys.

EXHIBIT 042-4

DETERMINING THE LOGICAL SPLICE POINT
--CASE 2--

CHAIN ORDER	When SELECT CURRENT is specified, the user must:
FIRST LAST	Initialize CURRENT OF TYPE for master records.
SORTED SORTED WITHIN TYPE	Initialize CURRENT of chain to indicate the splice point.
BEFORE AFTER	Initialize CURRENT of chain to indicate the splice point

EXHIBIT IDS042-5

STORING A DETAIL OF A CHAIN ORDERED <u>FIRST</u>

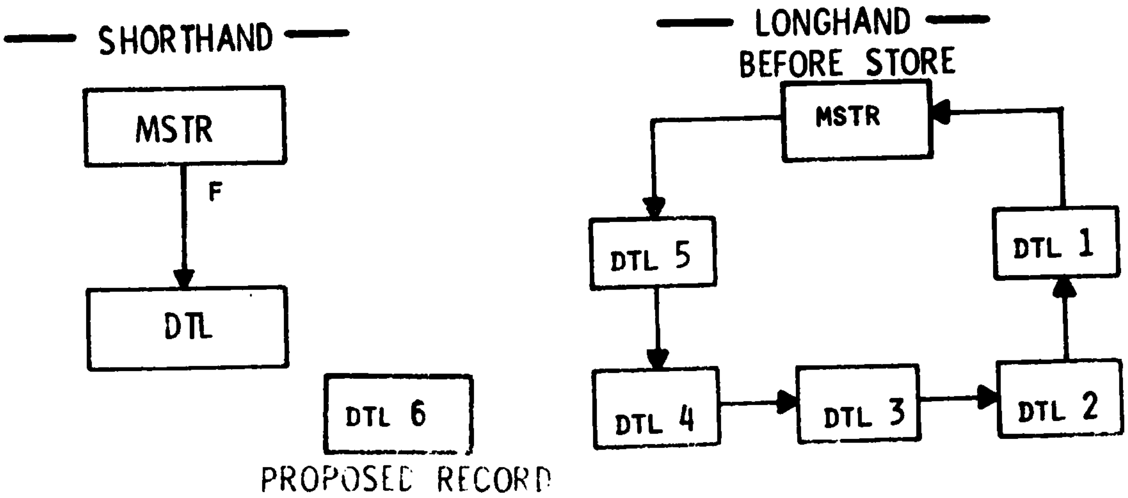


EXHIBIT IDS042-6

STORING A DETAIL OF A CHAIN ORDERED <u>FIRST</u>

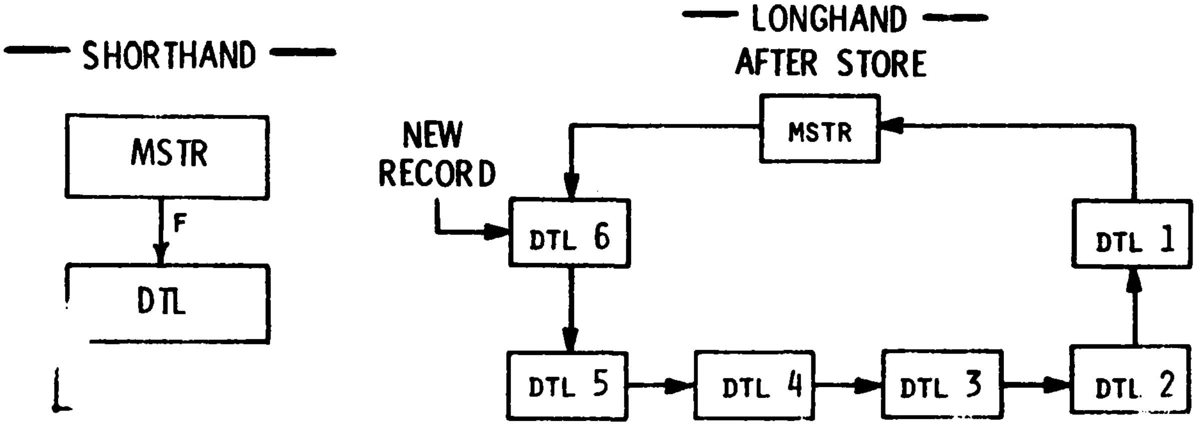


EXHIBIT IDS042-7

STORING A DETAIL OF A CHAIN ORDERED LAST

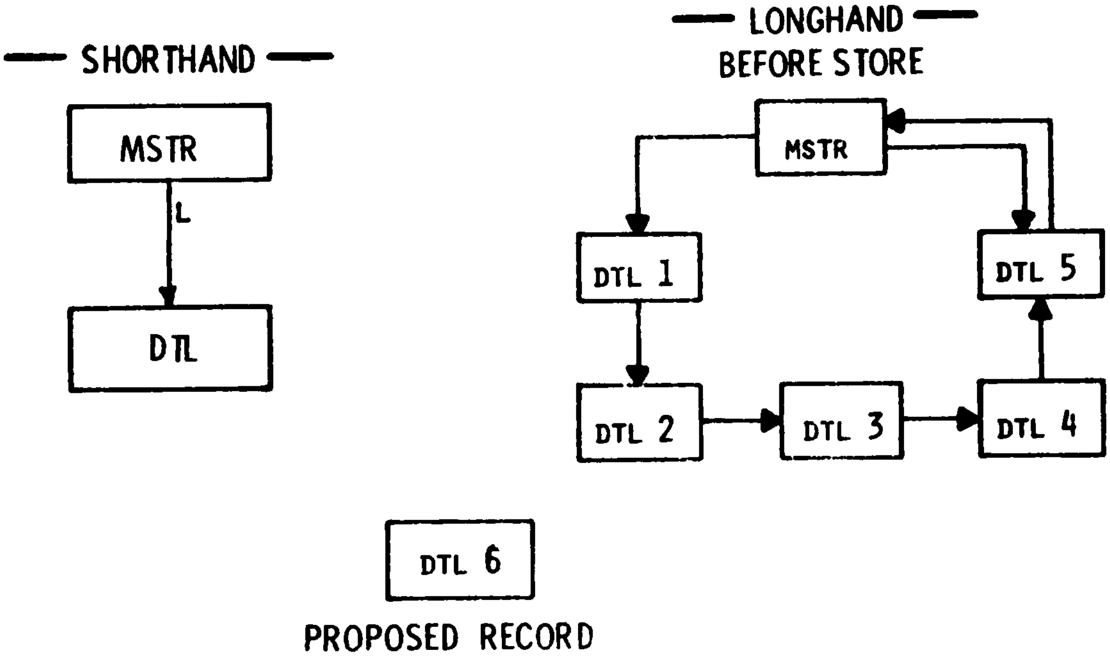


EXHIBIT IDS042-8

STORING A DETAIL OF A CHAIN ORDERED LAST

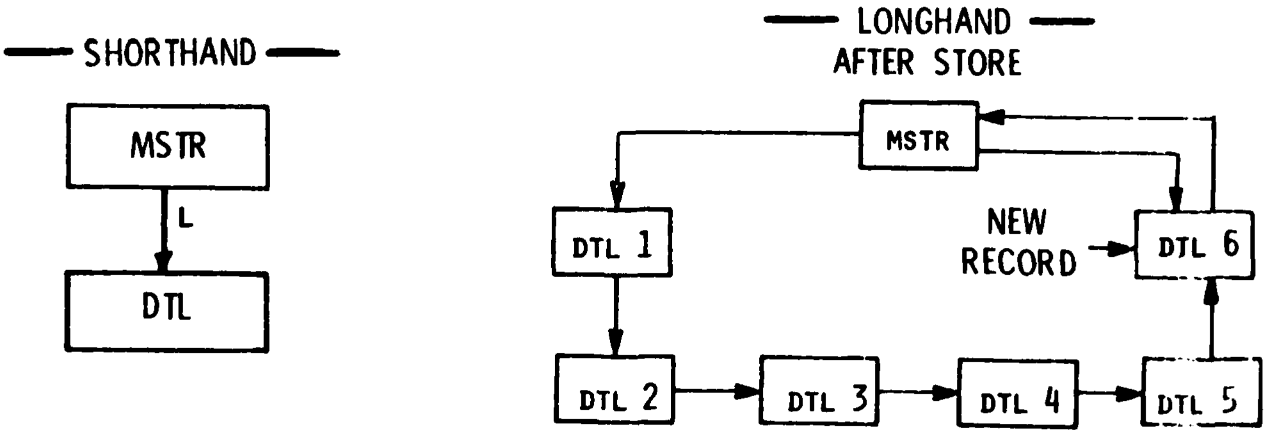


EXHIBIT IDS042-9

STORING A DETAIL OF A CHAIN ORDERED **BEFORE**

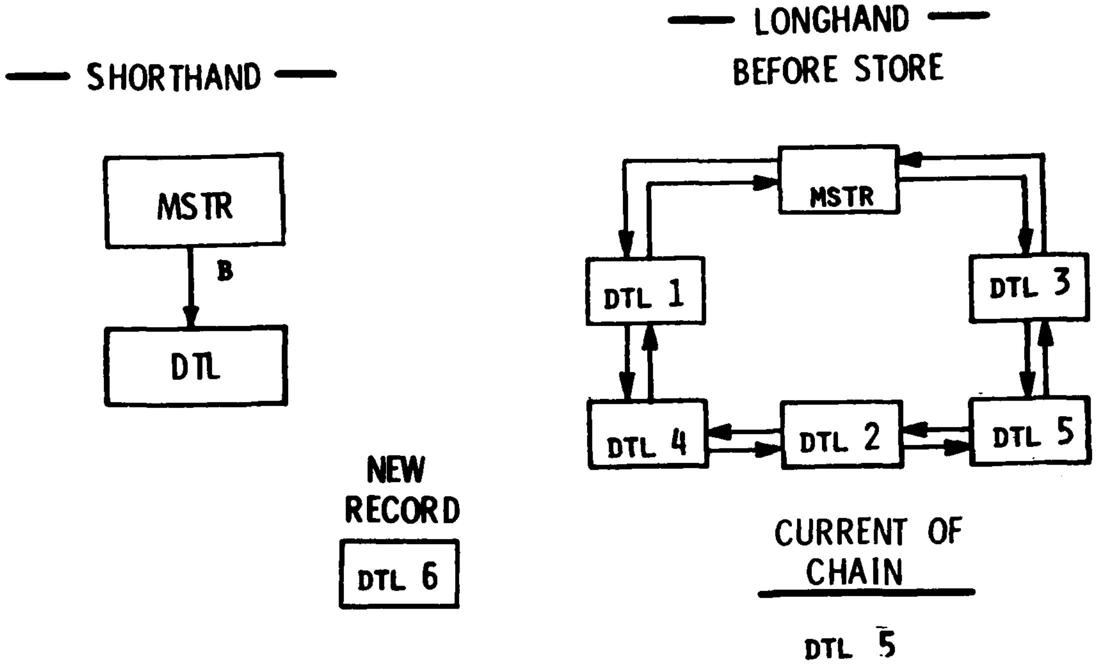


EXHIBIT IDS042-10

STORING A DETAIL OF A CHAIN ORDERED **BEFORE**

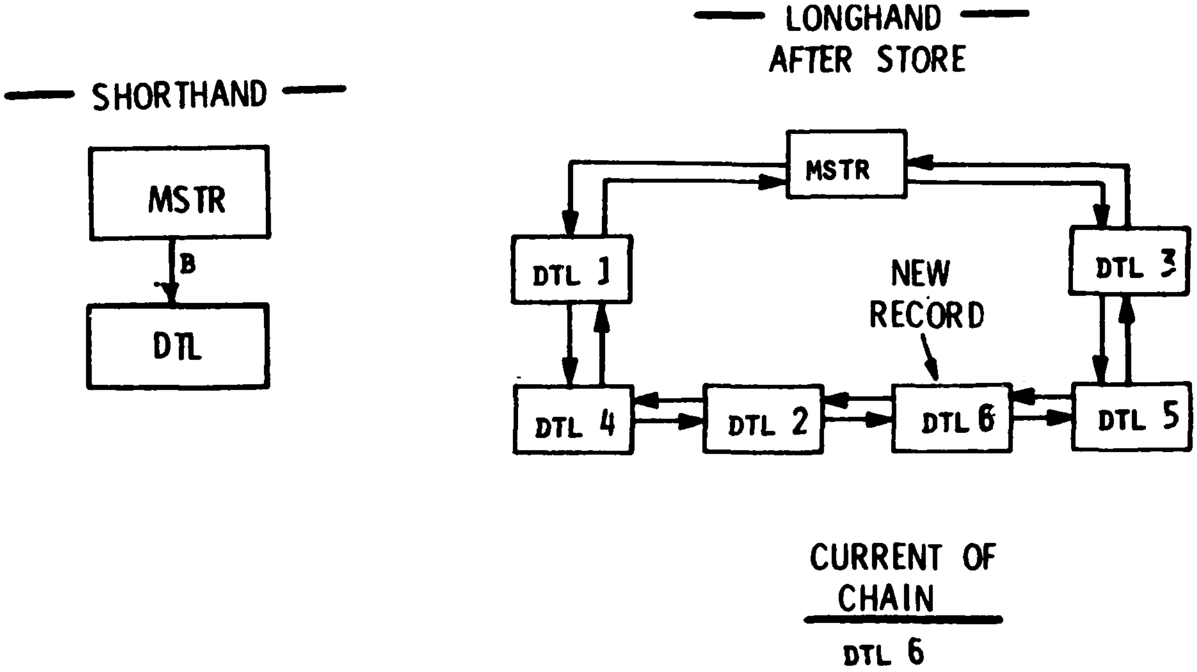


EXHIBIT IDS042-11

STORING A DETAIL OF A CHAIN ORDERED <u>AFTER</u>

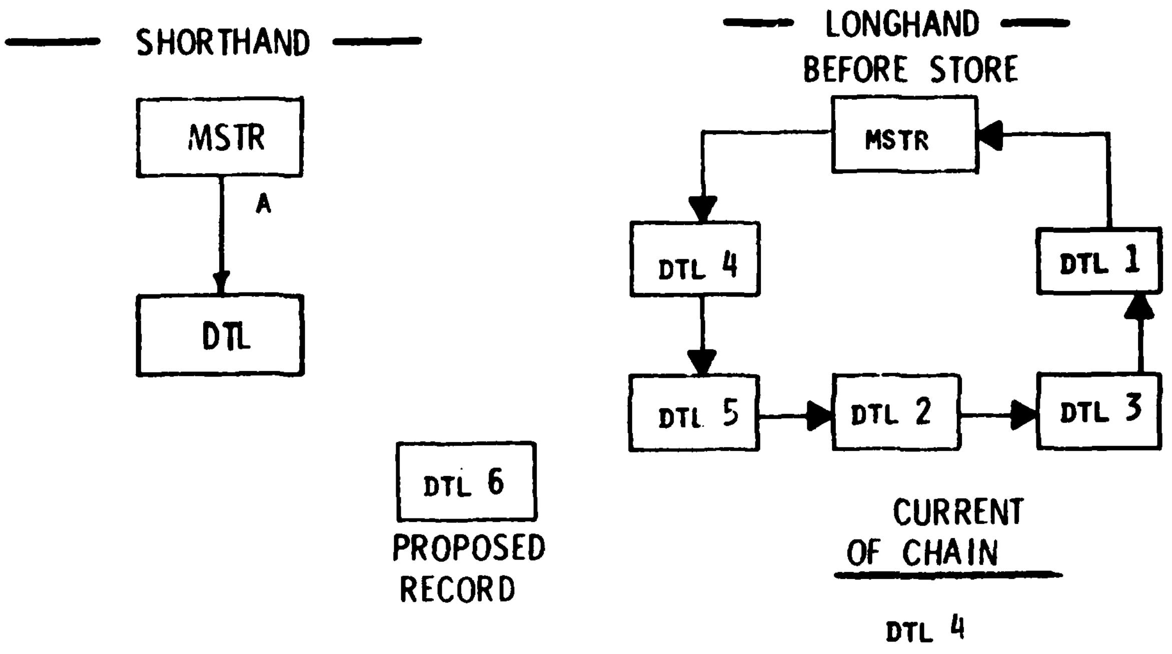


EXHIBIT IDS042-12

STORING A DETAIL OF A CHAIN ORDERED <u>AFTER</u>

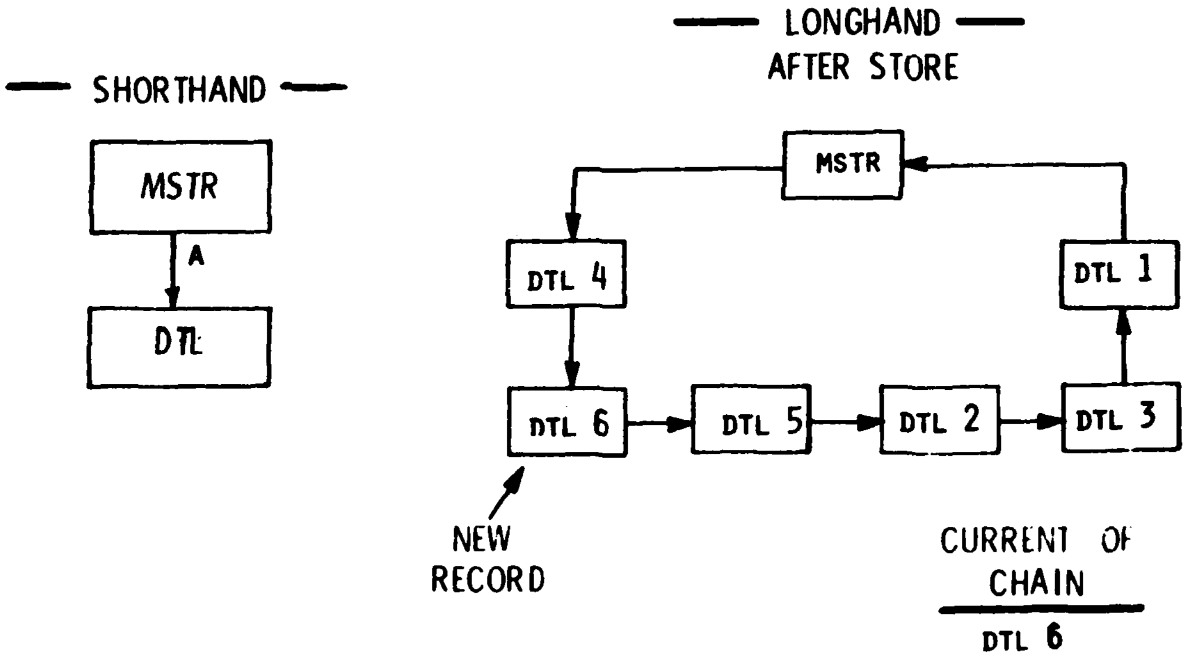


EXHIBIT IDS042-13

STORING A DETAIL OF A CHAIN ORDERED SORTED

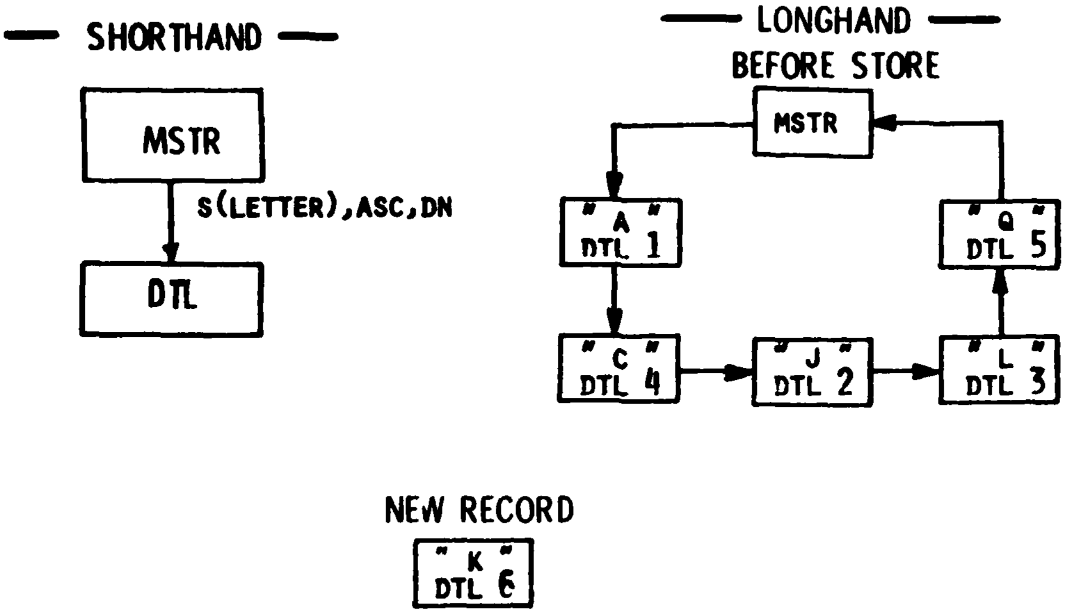


EXHIBIT IDS042-14

STORING A DETAIL OF A CHAIN ORDERED SORTED

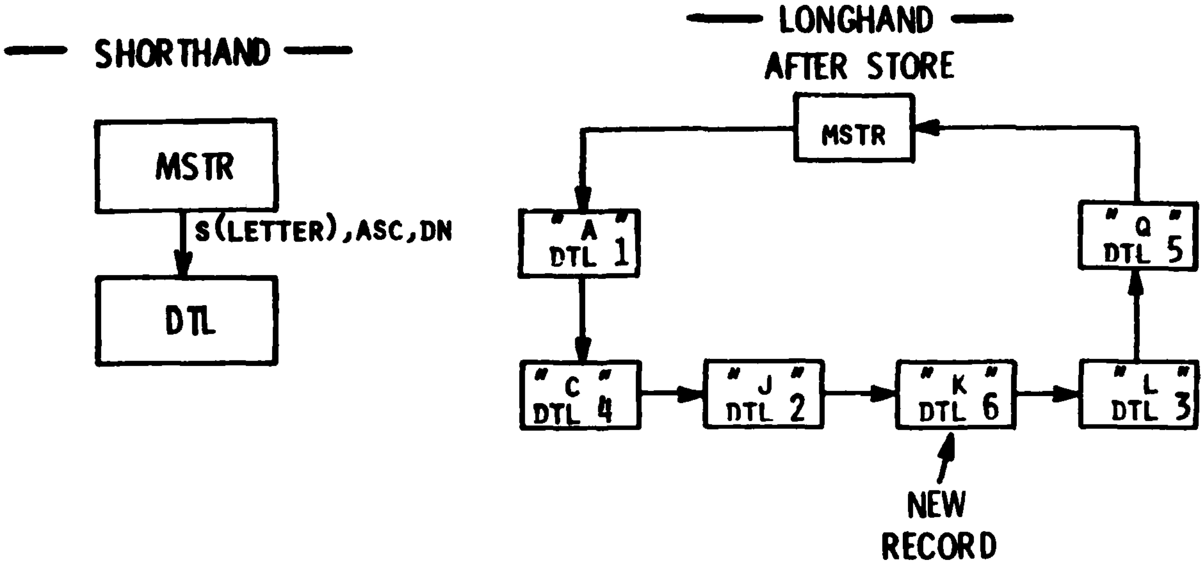


EXHIBIT IDS042-15

STORING A DETAIL WHEN CHAIN-ORDER IS SORTED WITHIN TYPE

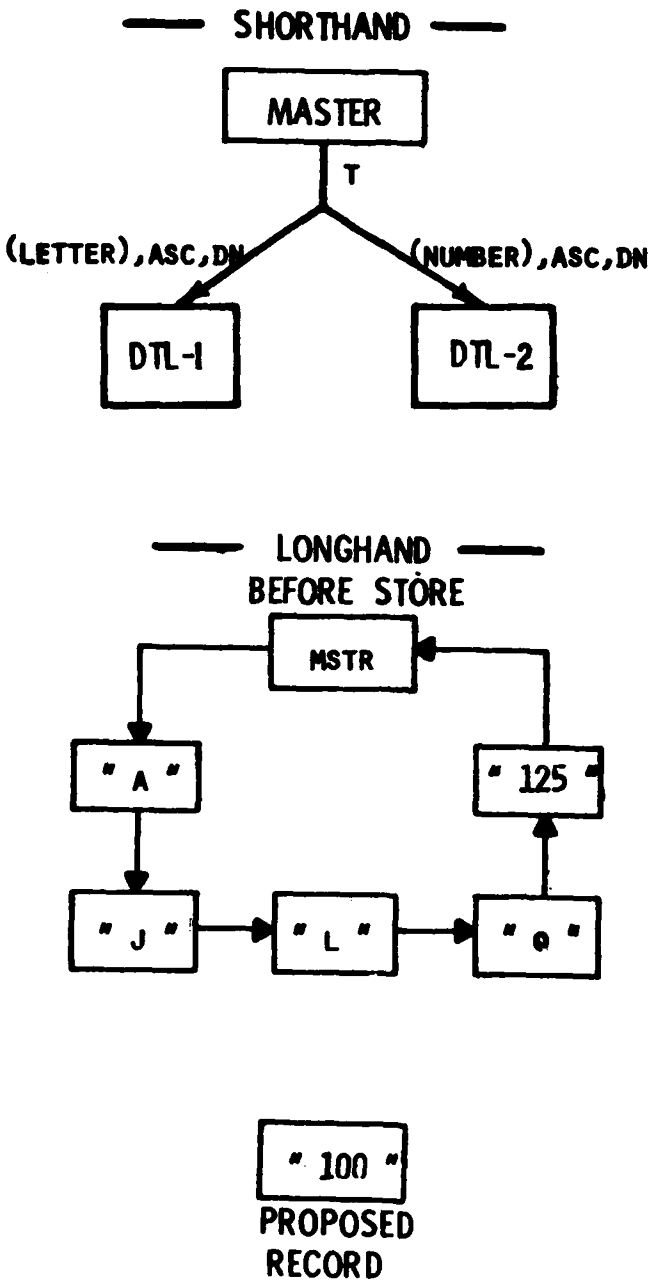
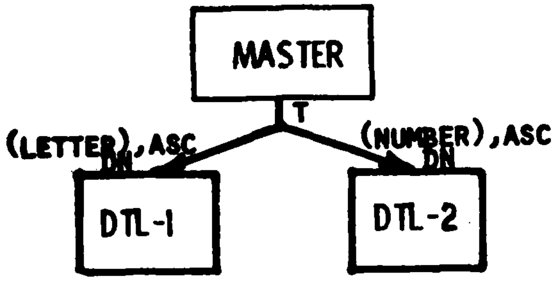


EXHIBIT IDS042-16

STORING A DETAIL WHEN CHAIN-ORDER IS SORTED WITHIN TYPE

— SHORTHAND —



— LONGHAND —

AFTER STORE

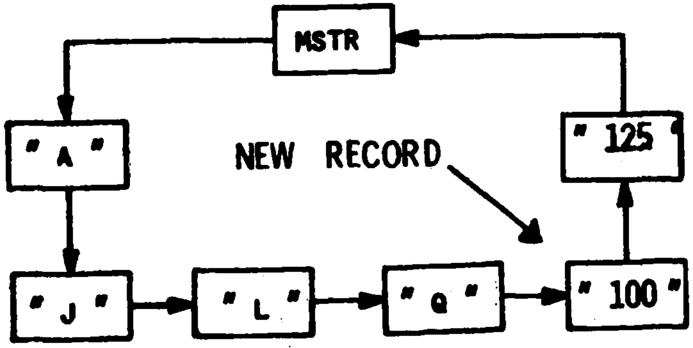
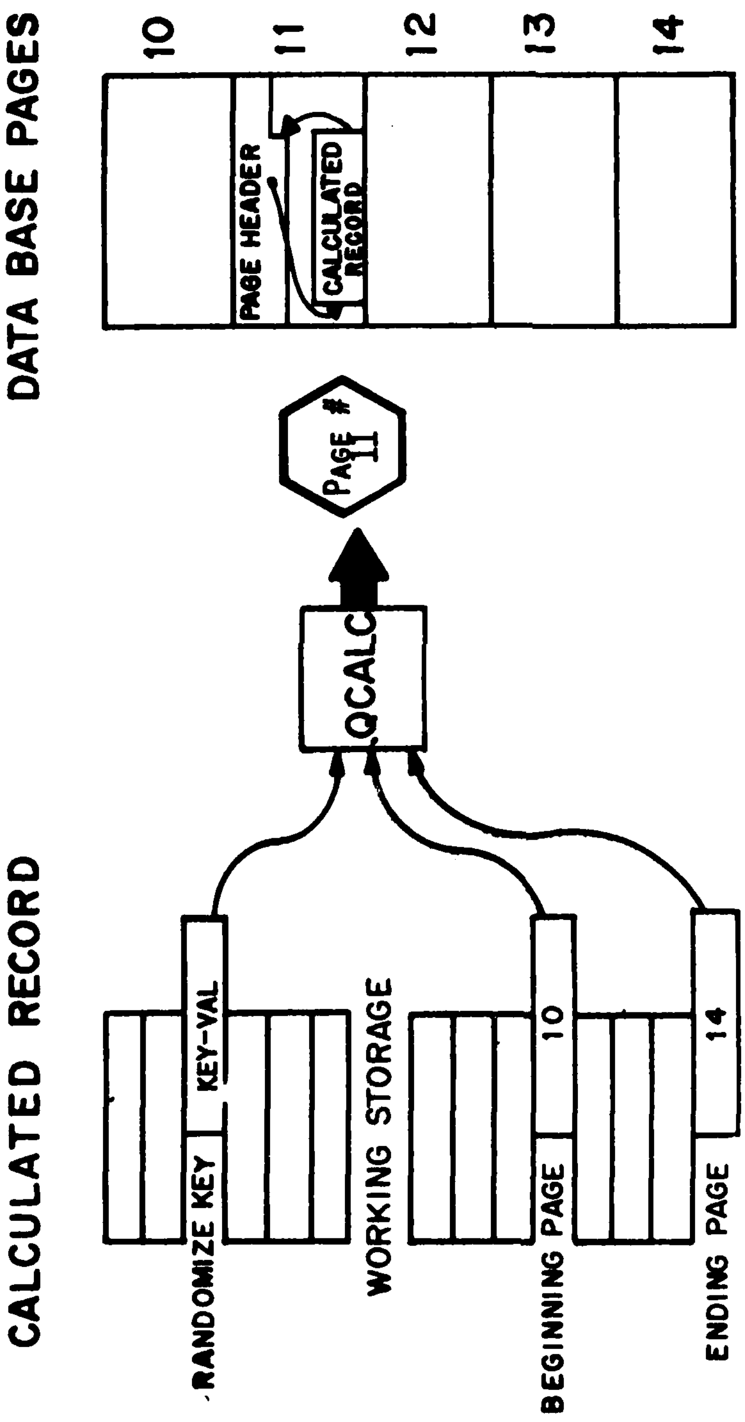


EXHIBIT IDS042-17

STORE/RETRIEVAL OF CALC RECORD

```
IDS SECTION.
MD  DATABASE.
01  RECORD-A                      TYPE IS 010
                                  RETRIEVAL VIA CALC CHAIN
                                  PAGE RANGE IS 10 THRU 14.
    02  CALC-KEY                  PIC ...
           .
           .
           .
    98  CALC CHAIN DETAIL
        RANDOMIZE ON CALC-KEY.
                 .
                 .
                 .
    MOVE KEY-VAL TO CALC-KEY.
    ENTER IDS.
        (RETRIEVE or STORE) RECORD-A IF ERROR ...
                    .
                    .
                    .
```

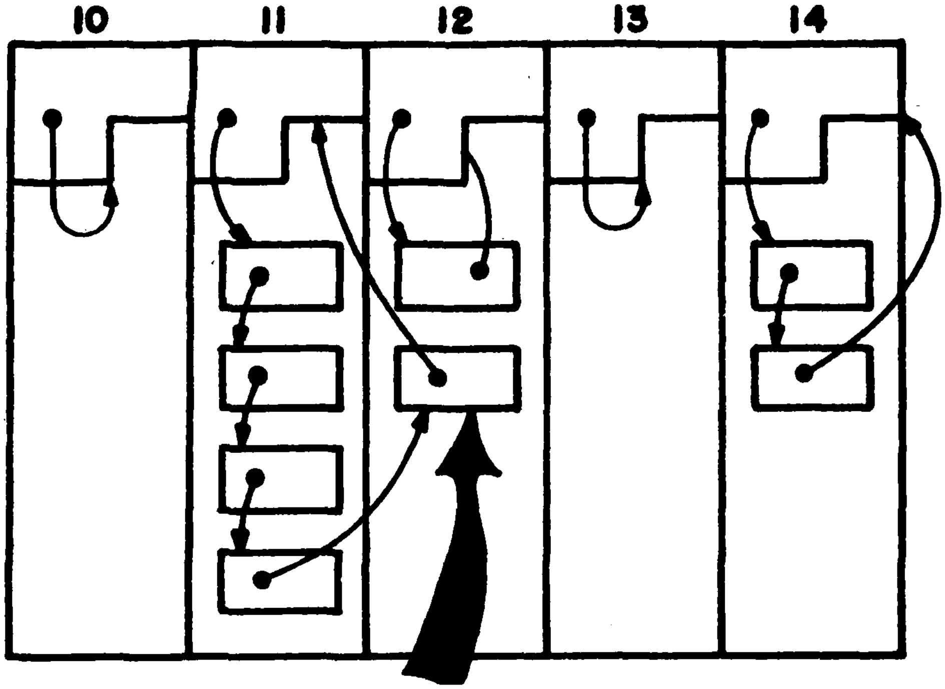
EXHIBIT IDS042-18



LOCATION OF CALCULATED RECORD

EXHIBIT IDS042-19

DATA BASE PAGES



Record has calculated to page 11, but is physically stored on page 12. Overflow on the calc chain.

EXHIBIT IDS042-20

TYPES OF IDS RETRIEVAL TRANSACTIONS

1. Unique retrieval by IDS

   a. Via reference code

   b. Via calc keys

   c. Via sort keys

2. Chain walking user search

3. Sequential user search

4. Physical user search

EXHIBIT IDS042-21

ACCESS CAPABILITIES AGAINST A DATA BASE

TYPE OF ACCESS	METHOD
RANDOM	CALC KEY
LOGICAL	CHAIN WALKING
INDEX-SEQUENTIAL	1 AND 2 ABOVE
DIRECT	REFERENCE CODE
PHYSICAL	PAGE UNPACKING
SEQUENTIAL	PAGE WALKING

EXHIBIT IDS042-22

RETRIEVING NEXT AROUND A CHAIN

```
PROCEDURE DIVISION
        .
        .
        .
ENTER IDS.
    RETRIEVE MSTR RECORD IF ERROR ...
        ELSE MOVE TO WORKING-STORAGE.
GET-NXT-CH1.
    ENTER IDS.
        RETRIEVE NEXT RECORD OF CH-1
        CHAIN IF ERROR... ELSE MOVE,
        IF MSTR RECORD GO TO DONE-IT ELSE
        IF DTL-1 PERFORM DETAIL-1 ELSE
        PERFORM DETAIL-2.
    GO TO GET-NXT-CH1.
DONE-IT.
        .
        .
        .
```

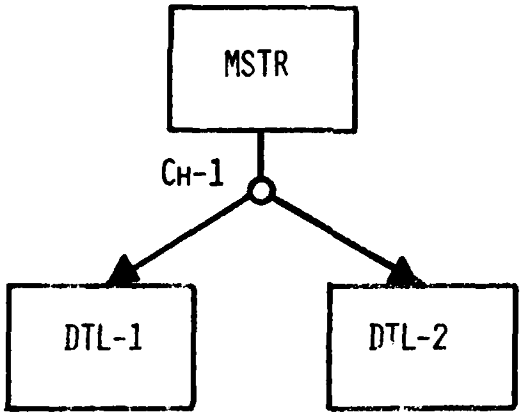


EXHIBIT IDS042-23

RETRIEVING NEXT AROUND A CHAIN

GET-NXT-CH1

FIRST TIME

CURRENT OF TYPES	
MSTR	66
DTL-1	0
DTL-2	0

CHAIN TABLE CH-1	
M	66
P	0
C	66
N	72

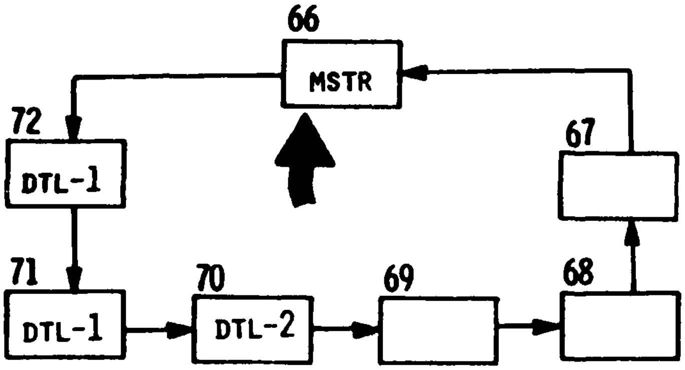


EXHIBIT IDS042-24

RETRIEVING NEXT AROUND A CHAIN

AT GET-NXT-CH1

SECOND TIME

CURRENT OF TYPES

66	MSTR
72	DTL-1
0	DTL-2

CHAIN TABLE CH-1

M	66
P	66
C	72
N	71

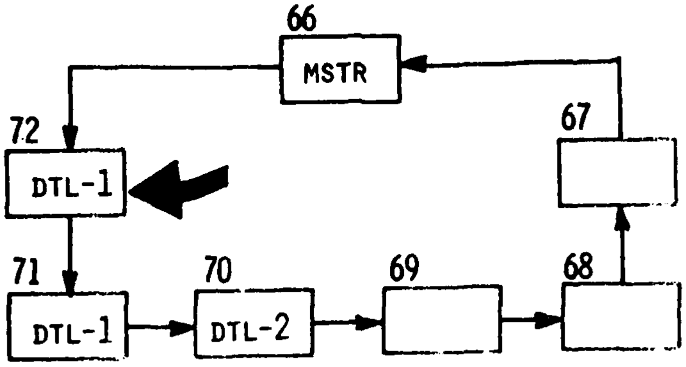


EXHIBIT IDS042-25

RETRIEVING NEXT AROUND A CHAIN

AT GET-NXT-CH1
THIRD TIME

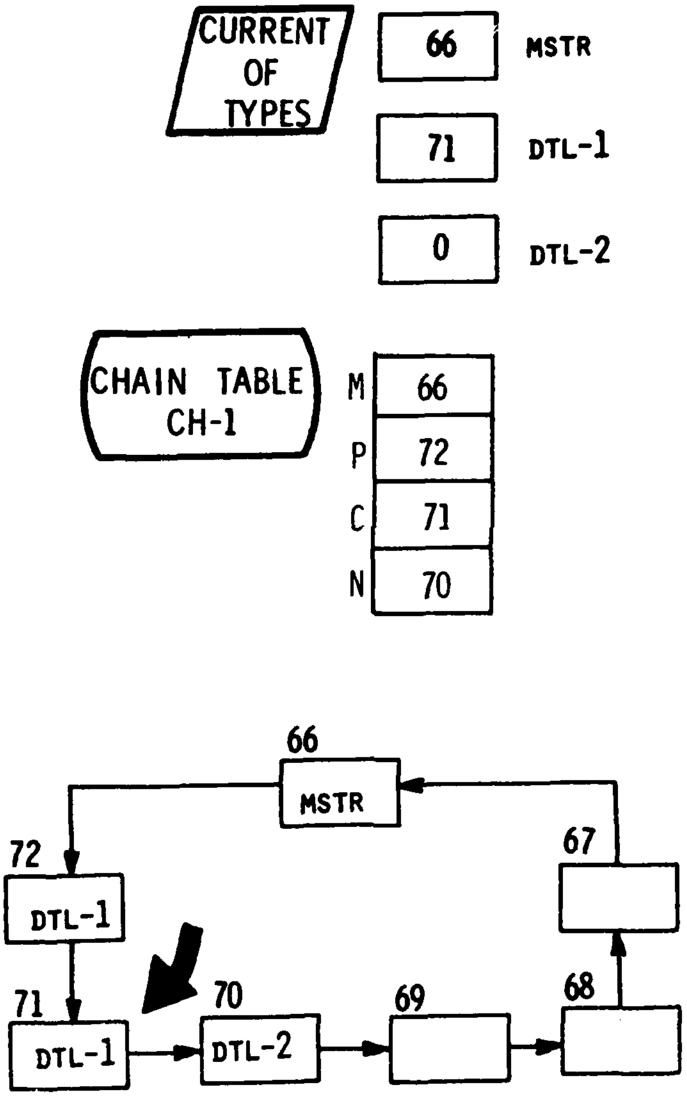


EXHIBIT IDS042-26

RETRIEVING NEXT AROUND A CHAIN

AT GET-NXT-CH1

__FOURTH TIME__

CURRENT OF TYPES		
	66	MSTR
	71	DTL-1
	70	DTL-2

CHAIN TABLE CH-1		
	M	66
	P	71
	C	70
	N	69

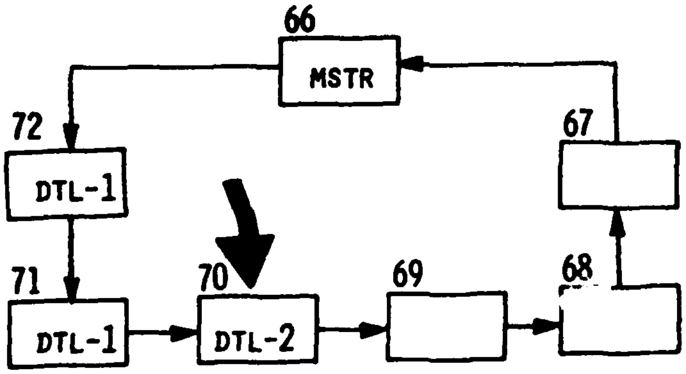


EXHIBIT IDS042-27

RETRIEVING NEXT AROUND A CHAIN

AFTER RETRIEVING PRIOR ONCE
(WITHOUT PRIOR POINTERS)

CURRENT OF TYPES

66	MSTR
71	DTL-1
70	DTL-2

CHAIN TABLE CH-1

M	66
P	0
C	71
N	70

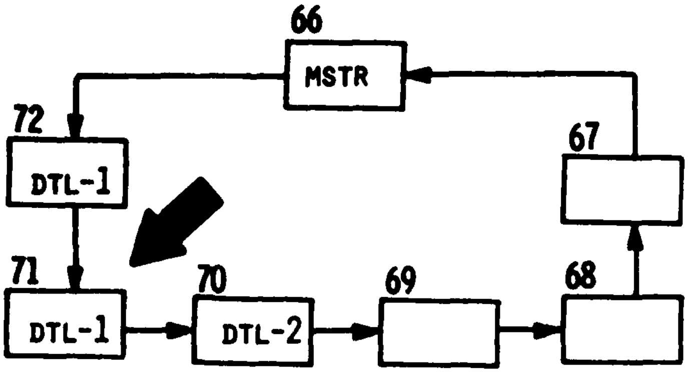


EXHIBIT IDS042-28

SUGGESTED SEQUENCE FOR MODIFYING A RECORD

1. Retrieve record to be modified.

2. Move retrieved record's field(s) to WORKING-STORAGE.

3. COBOL MOVE the new information into the field(s) to be modified.

4. PERFORM the MODIFY verb.

EXHIBIT IDS042-29

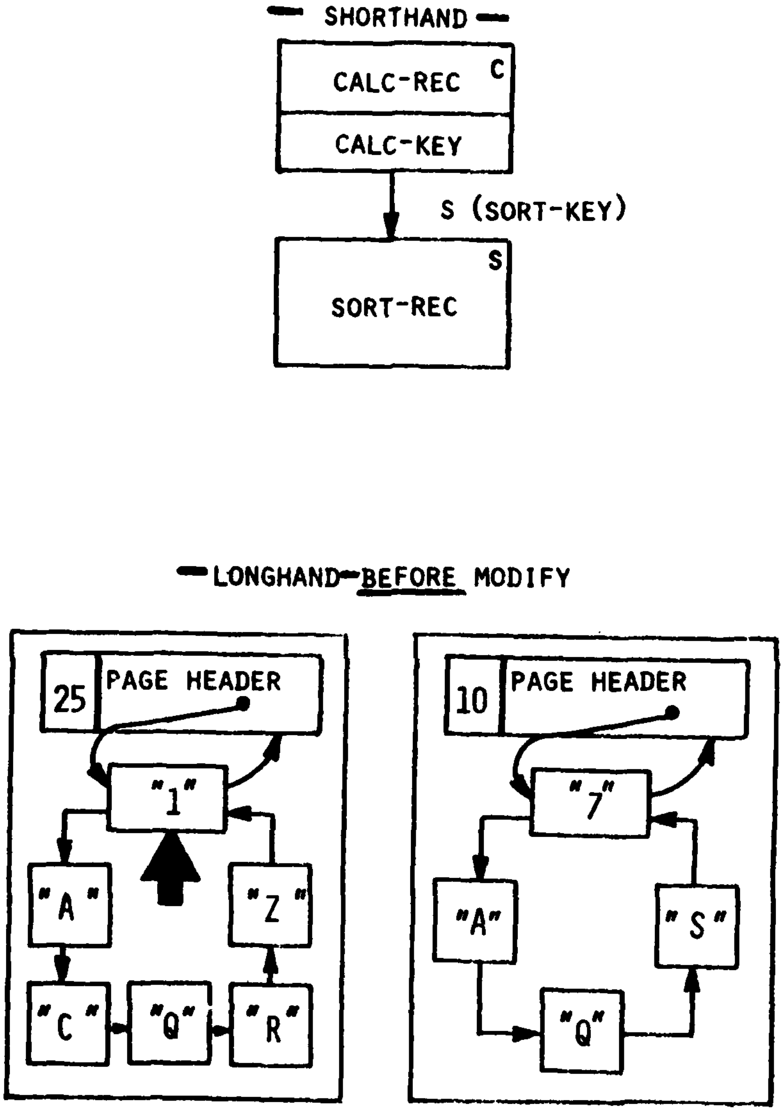
MODIFY capabilities are <u>always</u> against the current system record.

MODIFICATION OF	RESULTS IN
DATA FIELD	Record being updated
CALC KEY	Record re-randomized and linked into new calc chain. Calc key field is updated.
SORT KEY	Chain is re-sorted, record is relinked. Sort key field is updated.
MATCH-KEY	Record is linked into new master occurrence. No fields are updated!

<u>All</u> relinking is <u>logical</u>; record <u>physically</u> remains on the page it was originally stored on.

EXHIBIT IDS042-30

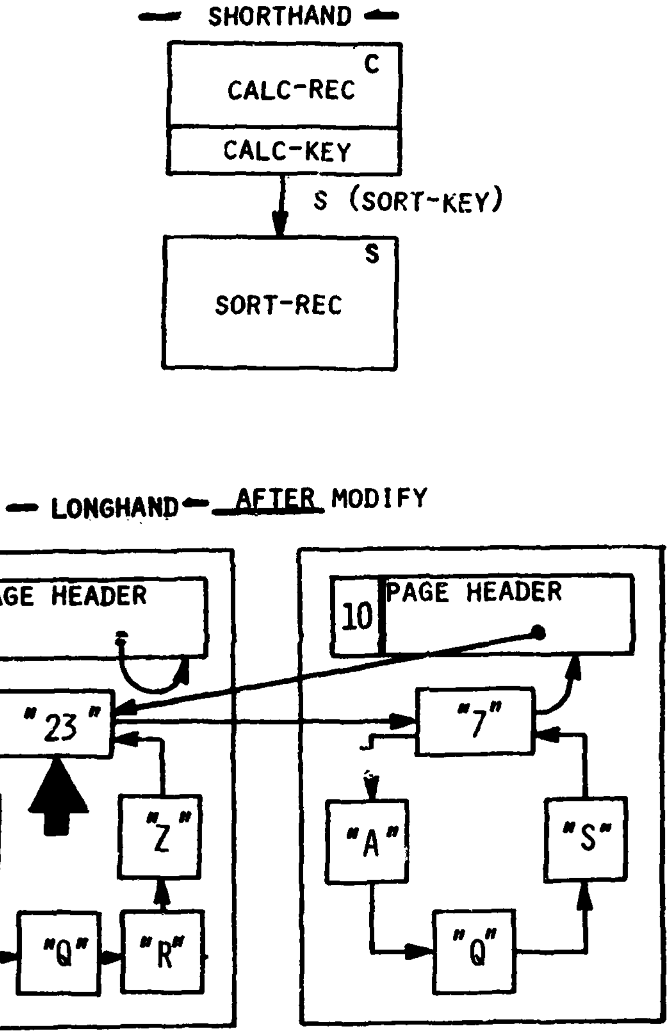
MODIFICATION OF A CALC KEY



MOVE "1" TO CALC-KEY.
ENTER IDS. RETRIEVE CALC-REC, MOVE.

EXHIBIT IDS042-31

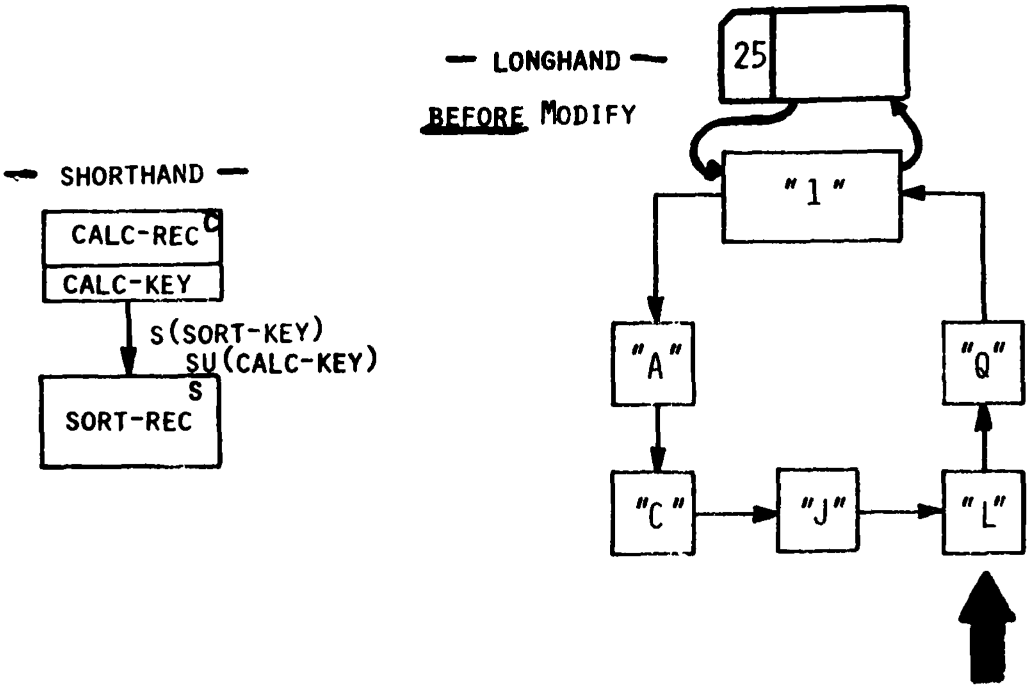
MODIFICATION OF A CALC KEY



MOVE "23" to CALC-KEY.
ENTER IDS. MODIFY CURRENT CALC-REC CALC-KEY.

EXHIBIT IDS042-32

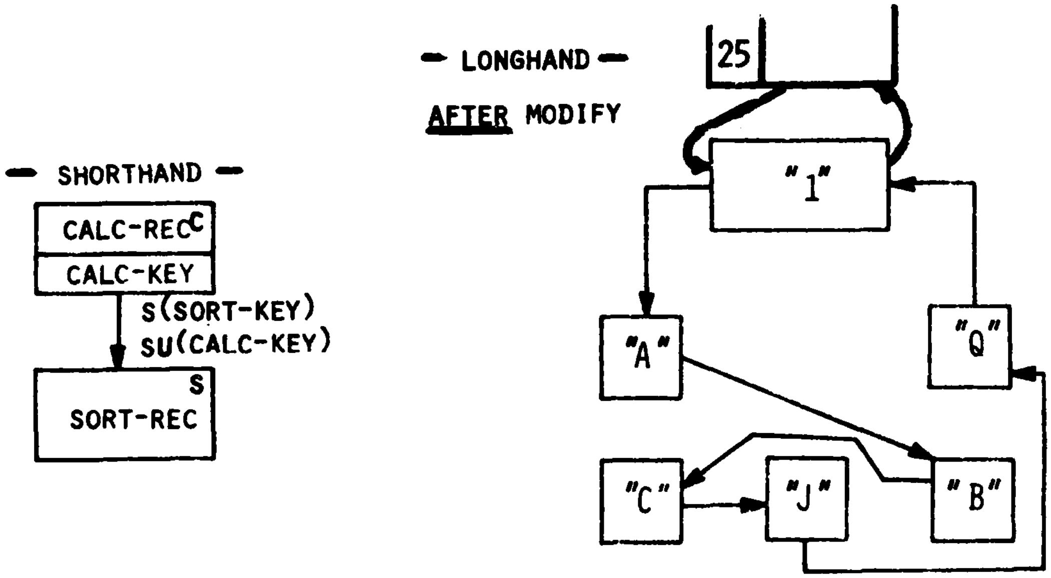
MODIFICATION OF A SORT-KEY



MOVE "1" TO CALC-KEY.
MOVE "L" TO SORT-KEY.
ENTER IDS. RETRIEVE SORT-REC, MOVE.

EXHIBIT IDS042-33

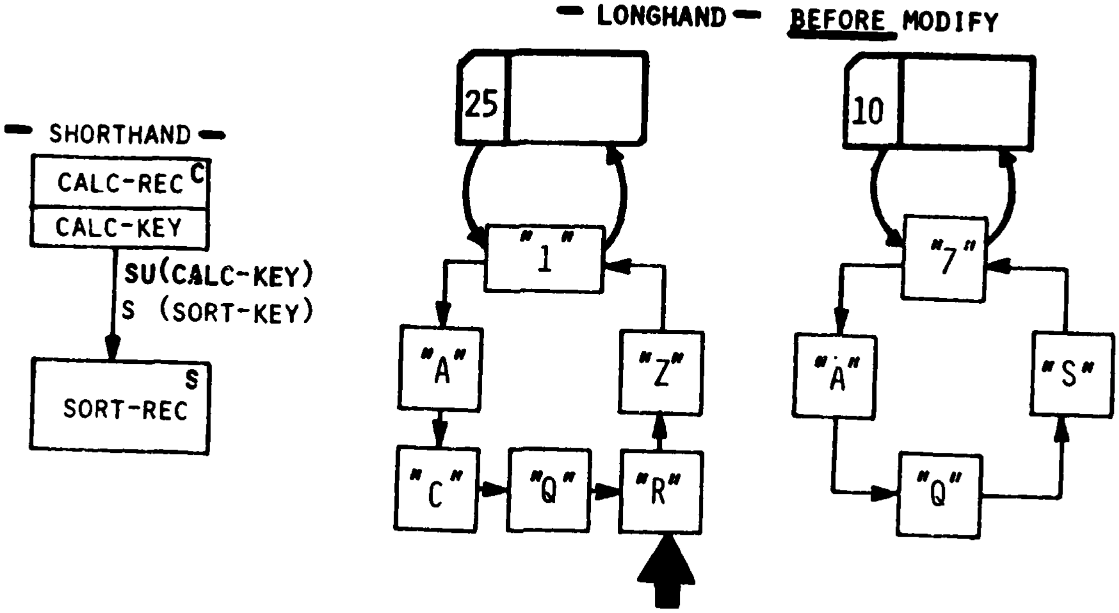
MODIFICATION OF A SORT-KEY



MOVE "B" TO SORT-KEY.
ENTER IDS. MODIFY CURRENT SORT-REC SORT-KEY.

EXHIBIT IDS042-34

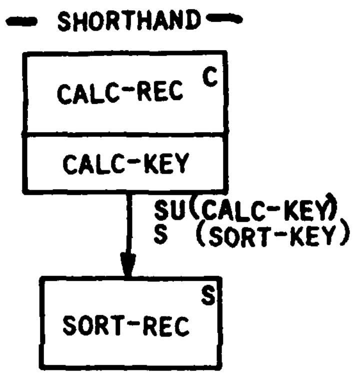
MODIFICATION OF A MATCH-KEY

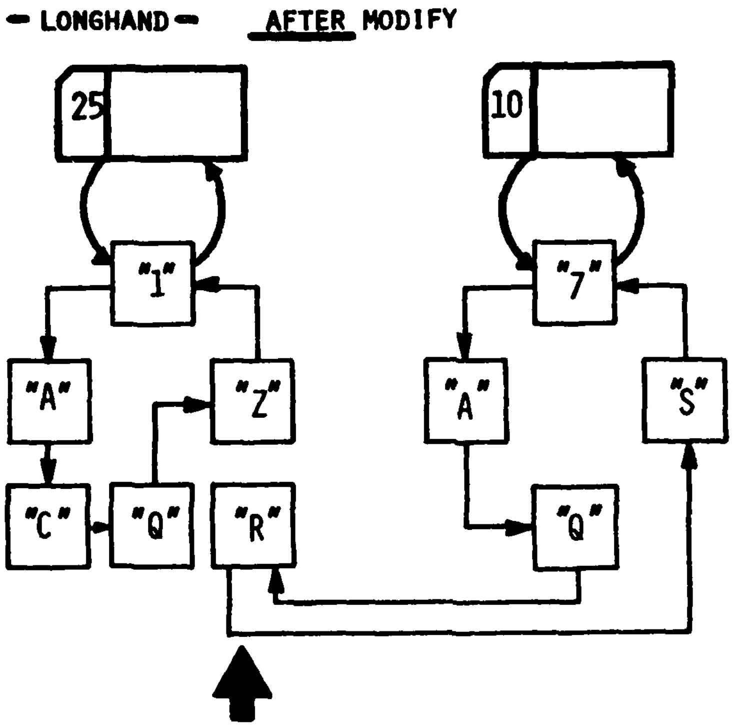


```
MOVE "1" TO CALC-KEY.
MOVE "R" TO SORT-KEY.
ENTER IDS.  RETRIEVE SORT-REC, MOVF
```

EXHIBIT IDS042-35

MODIFICATION OF A MATCH-KEY





MOVE "7" TO CALC-KEY.
ENTER IDS. MODIFY CURRENT SORT-REC CALC-KEY.

EXHIBIT IDS042-36

IDS DELETE TRANSACTIONS

1. Retrieve record to be deleted

2. Perform DELETE verb.

EXHIBIT IDS042-37

IDS DELETION

LOGICAL

1. Sets DELETE switch in record.
2. Begins setting chain pointers to Ø as record is de-linked from the corresponding chains that it participates in.

PHYSICAL

1. Releases line number by resetting appropriate bit.
2. Releases space and updates space available counter in page header (and inventory, if configured).
3. Adjusts other records up over the released space so that all available space is again at the bottom of the page.

EXHIBIT IDS042-38

PROBLEM

Lesson IDS043 will consist of the coding of the PROCEDURE DIVISION of a program which will load an IDS data base.

The program we will code in IDS043 will provide the initial load of a portion of the data base. The portion we will load represents the stable data in the data base (i.e., those records which define the constant factory system). The records we will load are the:

SHOP - Those factory shops which assemble or purchase particular parts.

PART - All parts manufactured (or purchased). These are details of the shop which produces them.

QTY-REQD - A quantity of component parts which are required in the assembling of major parts. (This data must be extracted from the COMPONENT-PART input record.)

Exhibit IDS043-2 gives a broad flowchart of the PROCEDURE DIVISION to be coded. The input format is specified in exhibit IDS043-3. Notice the input sequence:

```
SHOPA
PART #22  (made by SHOP A)
PART #47  (made by SHOP A)
SHOPB
PART #5   (made by SHOP B)
PART #84  (made by SHOP B)
  .
  .
  .
5 units of PART #5 for each PART #22
2 units of PART #84 for each PART #5
6 units of PART #5 for each PART #47,
  etc.
```

In other words, each SHOP record, followed by its PART records will be input until all shops and the parts they produce are exhausted. Then all the QTY-REQD records will be read at the end of the input file. The shop (CSHOP) and part (CPART) input records are pre-sorted; thus, using chain-order AFTER and SELECT CURRENT during the load will result in chains SHOP-LIST and PARTS-CONTROL being sorted by SHOP-ID and PART-NO respectively.

EXHIBIT IDS043-1

To help understand this structure better, let's look at an application. To find the components of a part, you walk the PARTS-LIST chain to get the QTY-REQD details and head up the COMPONENT chain from the detail to identify the component parts.

Ultimately you must have a part (for example, raw material) which is not produced at this factory. To handle this, we will assume that one of our shops is the purchasing department.

Before starting lesson IDS043, you should read and study the contents of the PROCEDURE DIVISION in section 3 of your IDS manual. Pay particular attention to the OPEN, STORE and CLOSE verbs, and the ENTER IDS statement. Then attempt to code the PROCEDURE DIVISION of our load problem using the flowchart in exhibit IDS043-2. The complete coding of the IDENTIFICATION, ENVIRONMENT, and DATA DIVISIONS are made available to you in exhibit IDS043-4. You may also find helpful the IDS structure diagram in exhibit IDS043-7.

To make it easier for you to code this program, assume that your input file is clean and that you will not have to anticipate any processing errors (for example, assume that all PART records corresponding to part numbers on a type 3 card have been already stored). Thus you will not have to validate your input or verify the storing of records on the data base. This limitation will permit you to concentrate on basic IDS programming.

When you have completed as much of the code as you can, take lesson IDS043. In this lesson you will code the PROCEDURE DIVISION. If you fail to properly code a statement, you will be given an explanation of how that statement should be coded and why at that point in the program.

GOOD LUCK!

FLOWCHART

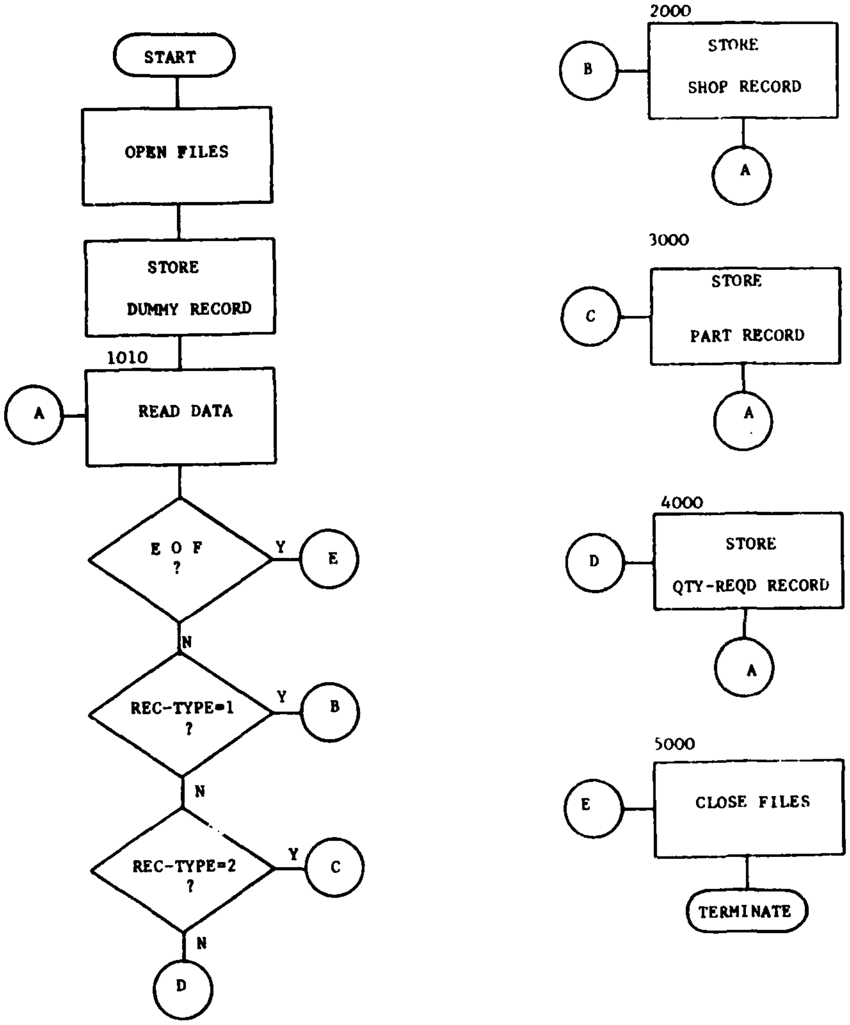


EXHIBIT IDS043-2

INPUT DATA SUMMARY

FORMAT

CONTENTS	CHARACTER POSITION	NUMBER OF CHARACTERS
SHOP RECORD		
TYPE "1"	1	1
SHOP ID	2	10
PART RECORD		
TYPE "2"	1	1
SHOP ID	2	10
PART NO	12	6
QUANTITY ON HAND	18	8
CQTY-REQD RECORD		
TYPE "3"	1	1
QUANTITY REQ	2	8
PART REQ (See Note 1)	10	6
FOR PART (See Note 2)	16	6

NOTE 1: Part number of corresponding subassembly component part.
NOTE 2: Part number of corresponding major assembly.

SEQUENCE

Record types 1 and 2 are sorted:

Major sort key: SHOP ID
Minor sort key: TYPE Number

Record type 2 is sorted:

Major sort key: PART NO

Record type 3 is appended to the end of the file.

EXHIBIT IDS043-3

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  IDS043.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
    SELECT IDS ORDER-CONTROL ASSIGN TO A1.
    SELECT CARDIN ASSIGN TO IN.

I-O-CONTROL.
    APPLY STANDARD ON CARDIN.


DATA DIVISION.

FILE SECTION.

FD  CARDIN LABEL RECORDS ARE STANDARD
    DATA RECORDS ARE CSHOP CPART COMPONENT-PART.

01  CSHOP.
    02  REC-TYPE          PIC X.
    02  CSHOP-ID          PIC X(10).
    02  FILLER            PIC X(69).

01  CPART.
    02  FILLER            PIC X(11).
    02  CPART-NO          PIC X(6).
    02  CQTY              PIC 9(8).
    02  FILLER            PIC X(55).

01  CQTY-REQD.
    02  FILLER            PIC X.
    02  CQTY-REQD-QTY     PIC 9(5).
    02  CPART-REQ         PIC X(6).
    02  FOR-PART          PIC X(6).
    02  FILLER            PIC X(59).

WORKING-STORAGE SECTION.
01  ALT-PART-NO           PIC X(6).


IDS SECTION.
MD  ORDER-CONTROL.
```

EXHIBIT IDS043-4

```
01  DUMMY
    TYPE IS 001
    RETRIEVAL VIA DUMMY-REF FIELD.
    02  DUMMY-REF        PIC 9(8).
    98  CUSTOMER-LIST CHAIN MASTER
        CHAIN-ORDER IS SORTED.
    98  SHOP-LIST CHAIN MASTER
        CHAIN-ORDER IS AFTER.

01  CUSTOMER
    TYPE IS 002
    RETRIEVAL VIA CALC CHAIN
    PAGE-RANGE IS 1 TO 25.
    02  CUST-ID          PIC X(20).
    02  CUST-INFO SIZE 120.
        03  FULL-NAME    PIC X(30).
        03  BILL-ADDR    PIC X(40).
        03  SHIP-ADDR    PIC X(40).
        03  PHONE        PIC X(10).
    98  CUSTOMER-LIST CHAIN DETAIL
        SELECT CURRENT MASTER
        DUPLICATES NOT ALLOWED
        ASCENDING KEY IS CUST-ID.
    98  CALC CHAIN DETAIL
        RANDOMIZE ON CUST-ID.
    98  CUSTOMER-ORDERS CHAIN MASTER
        CHAIN-ORDER IS AFTER.

01  SHOP
    TYPE IS 003
    RETRIEVAL VIA SHOP-LIST CHAIN
    PAGE-RANGE IS 51 TO 150.
    02  SHOP-ID          PIC X(10).
    98  SHOP-LIST CHAIN DETAIL
        SELECT CURRENT MASTER.
    98  DUE-IN CHAIN MASTER
        CHAIN-ORDER IS AFTER.
    98  PARTS-CONTROL CHAIN MASTER
        CHAIN-ORDER IS AFTER.
```

EXHIBIT IDS043-4 (Cont)

```
01  ORDER
    TYPE IS 004
    RETRIEVAL VIA CALC CHAIN
    PAGE-RANGE IS 26 TO 50.
    02  ORDER-NO         PIC 9(8).
    02  DATA-RCVD        PIC X(6).
    98  CUSTOMER-ORDERS CHAIN DETAIL
        SELECT CURRENT MASTER.
    98  CALC CHAIN DETAIL
        RANDOMIZE ON ORDER-NO.
    98  INVOICE CHAIN MASTER
        CHAIN-ORDER IS LAST.
    98  ORDER-QTY CHAIN MASTER
        CHAIN-ORDER IS AFTER.

01  PART
    TYPE IS 005
    RETRIEVAL VIA CALC CHAIN
    PAGE-RANGE IS 51 TO 150.
    02  PART-NO          PIC X(6).
    02  QTY-ON-HAND      PIC 9(8).
    98  PARTS-CONTROL CHAIN DETAIL
        SELECT CURRENT MASTER.
    98  CALC CHAIN DETAIL
        RANDOMIZE ON PART-NO.
    98  DUE-OUT CHAIN MASTER
        CHAIN-ORDER IS LAST.
    98  PARTS-LIST CHAIN MASTER
        CHAIN-ORDER IS FIRST.
    98  COMPONENT CHAIN MASTER
        CHAIN-ORDER IS FIRST.

01  QTY-DUE
    TYPE IS 006
    RETRIEVAL VIA DUE-OUT CHAIN
    PAGE-RANGE IS 51 TO 150.
    02  QTY-DUE-QTY      PIC 9(8).
    98  DUE-IN CHAIN DETAIL
        SELECT CURRENT MASTER.
    98  DUE-OUT CHAIN DETAIL.
        SELECT CURRENT MASTER.
```

EXHIBIT IDS043-4 (Cont)

```
01  QTY-REQD
    TYPE IS 007
    RETRIEVAL VIA PARTS-LIST CHAIN
    PAGE-RANGE IS 51 TO 150.
    02  QTY-REQD-QTY       PIC 9(8).
    98  PARTS-LIST CHAIN DETAIL
        SELECT UNIQUE MASTER
        MATCH-KEY IS PART-NO.
    98  COMPONENT CHAIN DETAIL
        SELECT UNIQUE MASTER
        MATCH-KEY IS ALT-PART-NO SYN PART-NO
        LINKED TO MASTER.

01  FILL
    TYPE IS 008
    RETRIEVAL VIA INVOICE CHAIN
    PAGE-RANGE IS 26 TO 50.
    02  FILL-PART-NO      PIC X(6).
    02  FILL-QTY          PIC 9(8).
    02  DATE-SHIP         PIC X(6).
    98  INVOICE CHAIN DETAIL
        SELECT CURRENT MASTER.

01  QTY-ORDERED
    TYPE IS 009
    RETRIEVAL VIA DUE-OUT CHAIN
    PAGE-RANGE IS 51 TO 150.
    02  QTY-ORDERED-QTY  PIC 9(8).
    98  ORDER-QTY CHAIN DETAIL
        SELECT CURRENT MASTER.
    98  DUE-OUT CHAIN DETAIL
        SELECT CURRENT MASTER.
```

EXHIBIT IDS043-4 (Cont)

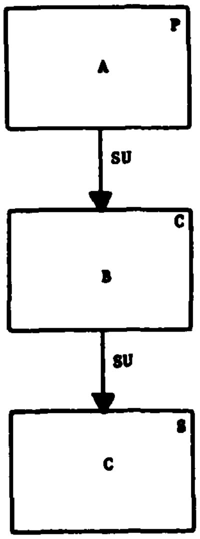


Figure 1

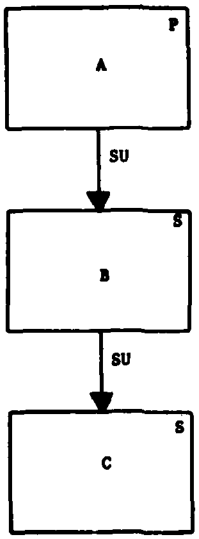


Figure 2

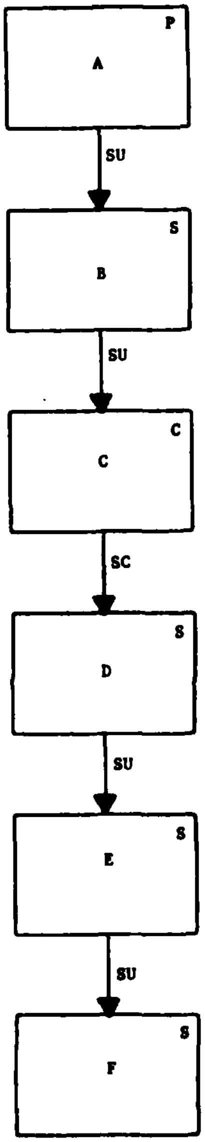


Figure 3

EXHIBIT IDS043-5

IDS DATA BASE LOAD PROGRAM

Use the shorthand structure diagram provided in Exhibit IDS043-8 to code the remaining parts of your data base load problem. The following records should be loaded:

ENTRY1
ENTRY2
LOCATIONX
EMPLOYEE
PART
SALE

The input data record formats are shown in this exhibit. The input records are presorted in ascending order with KODE the major sort key field.

Within KODE (1,2,3):

LOCATION-MAN records are sorted in ascending order with:

I-DISTRICT-NO (major sort key)
I-EMPLOYEE-NO (minor sort key)

PARTS records are sorted in ascending order with:

I-PART-NO (major sort key)

TRANSACTION records are sorted in ascending order with:

T-DISTRICT (major sort key)
T-EMPLOYEE-NO (minor sort key)

INPUT RECORDS

Record Name	Data Field Names	Picture
LOCATION-MAN	I-DISTRICT-NO	9(1)
	I-NAME	X(18)
	I-EMPLOYEE-NO	X(5)
	I-FIXED-SALARY	9(3)V9(2)
	I-SALES-YTD	9(7)V9(2)
	I-WAGE-YTD	9(5)V9(2)
	I-SSAN	9(9)
	KODE (col 55=1)	9(1)
	FILLER	X(25)
PARTS		
	I-PART-NO	9(8)
	I-PART-DESC	X(12)
	I-COST	9(4)V9(2)
	FILLER	X(28)
	KODE (col 55=2)	9(1)
	FILLER	X(25)

EXHIBIT IDS043-6

TRANSACT

T-PART-NO	9(8)
T-SALES-QTY	9(5)
T-EMPLOYEE-NO	X(5)
T-PCT-COM	9(2)
FILLER	X(32)
T-DISTRICT-NO (needed only for sort)	9(1)
FILLER	X(1)
KODE (col 55=3)	9(1)
FILLER	X(25)

NOTE 1: "LOCATION-MAN" input records provide data from both the "LOCATION" and "EMPLOYEE" IDS records.

NOTE 2: "PARTS" input records provide data for the "PART" IDS records.

NOTE 3: "TRANSACT" input records provide data for the "SALE" IDS records.

EXHIBIT IDS043-6 (Cont)

JCL

```
COL     1       8       16

        $       IDENT
        $       USERID  NAME$PASSWORD/UZZ
        $       FILSYS
                USERID  NAME$PASSWORD
        FCREAT/IDS/  CATALOG/FILENAME.
                     BASESIZE/19/,RNG/1,19/,LLINKS/17/,MODE/RAND/,
                     FCLASS/UZZ/,INVENTORY/NO/,PAGESIZE/256/
        $       PROGRAM QUTI
        $       PRMFL   A1,W,R,CATALOG/FILENAME
        $       DATA    I*
        IDS     INITIAL 1,19
        $       IDS     NDECK
        $       SET     18,19
        $       LIMITS  ,,,2000
                .
                . Load Program Source Code
                .
        $       EXECUTE
        $       LIMITS  ,23K
        $       PRMFL   FC,R,S,CDTSCDTS/IDS/LOADDATA
        $       PRMFL   A1,W,R,CATALOG/FILENAME
        $       PROGRAM QUTD
        $       LIMITS  ,20K
        $       PRMFL   A1,W,R,CATALOG/FILENAME
        $       DATA    I*
        IDS     PRINT   RNG/1,19/,NULL
        $       BREAK                     }
        $       FILSYS                    }     Only needed
                USERID  NAME$PASSWORD     }     during debugging
        FP      CATALOG/FILENAME          }
        $       ENDJOB
        ***EOF
```

NOTE 1: Replace "NAME$PASSWORD" with your USERID and PASSWORD.

NOTE 2: Replace "CATALOG/FILENAME" with a filename to be created under your USERID.

NOTE 3: Replace "FC" with the file codes that correspond to your FILE-CONTROL paragraph.

EXHIBIT IDS043-6 (Cont)

"ORDER-CONTROL" DATA BASE STRUCTURE

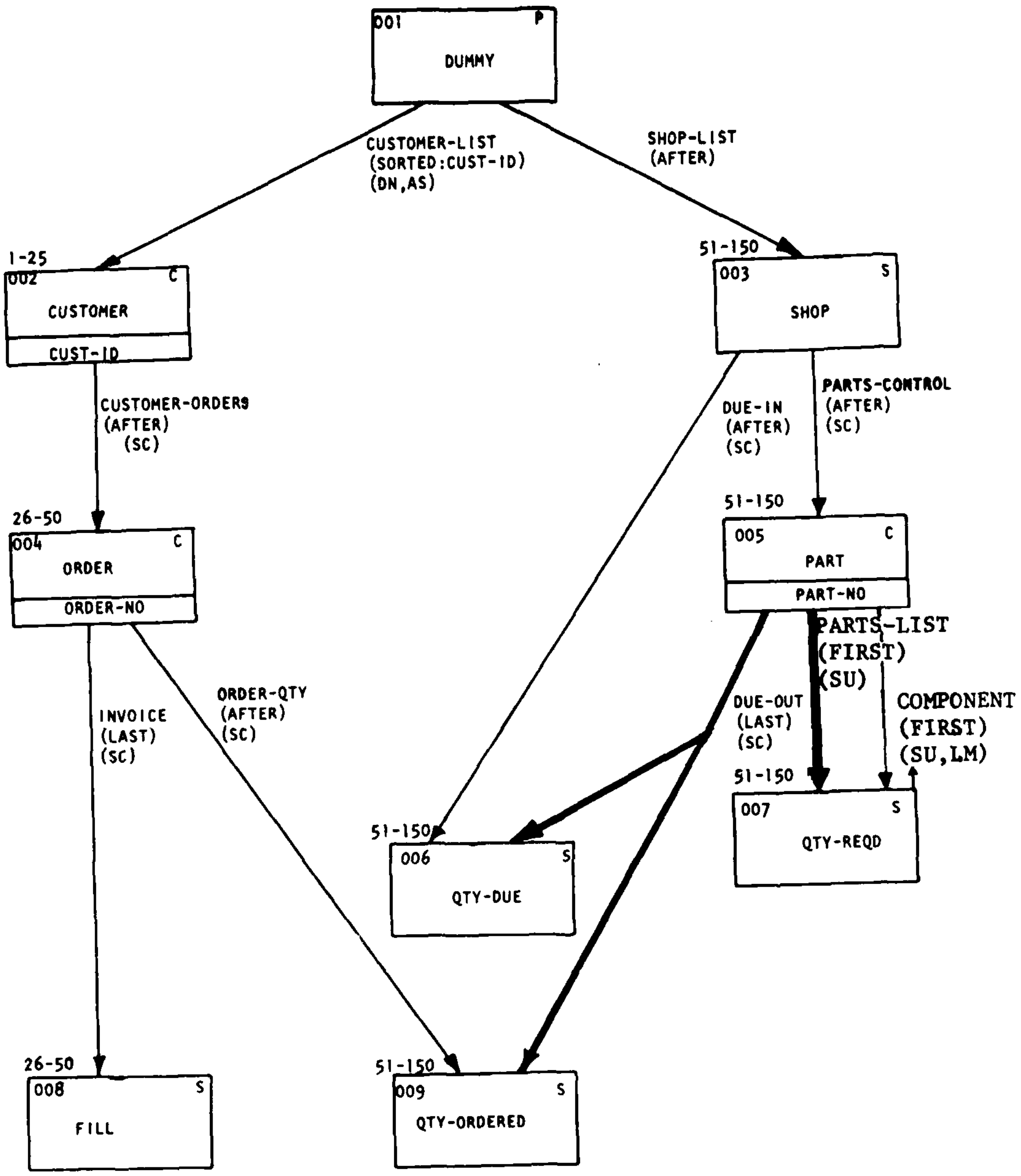


EXHIBIT IDS043-7

RECORD NAME	DATA FIELDS	PIC-SIZE
DUMMY		
	DUMMY-REF	9(8) (Retrieval Field)
CUSTOMER		
	CUST-ID	X(20)
	CUST-INFO	X(120)
	(contains the following subfields)	
	FULL-NAME	X(30)
	BILLING-ADDR	X(40)
	SHIP-ADDR	X(40)
	PHONE	X(10)
SHOP		
	SHOP-ID	X(10)
ORDER		
	ORDER-NO	9(8)
	DATE-RCVD	X(6)
PART		
	PART-NO	X(6)
	QTY-ON-HAND	9(8)
QTY-DUE		
	QTY-DUE-QTY	9(8)
QTY-REQD		
	QTY-REQD-QTY	9(8)
FILL		
	FILL-PART-NO	X(6)
	FILL-QTY	9(8)
	DATE-SHIP	X(6)
QTY-ORDERED		
	QTY-ORDERED-QTY	9(8)

EXHIBIT IDS043-7 (Cont)

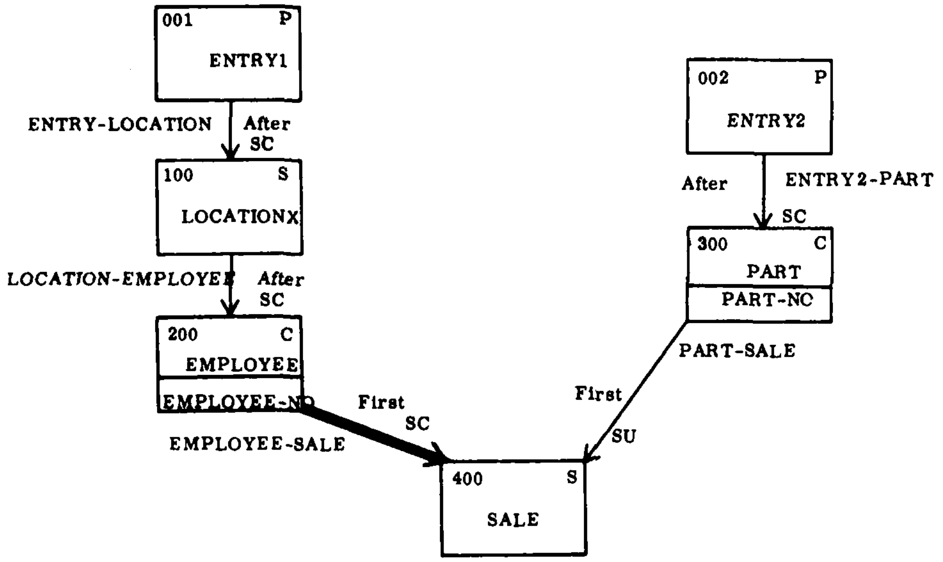
```
COL       1      8       16

          $      IDENT
          $      USERID  NAME$PASSWORD/UZZ
          $      IDS     NDECK
                 IDENTIFICATION DIVISION.
                 PROGRAM-ID.  IDW043.
                 ENVIRONMENT DIVISION.
                 CONFIGURATION SECTION.
                 SOURCE-COMPUTER.  6000 WITH EIS.
                 OBJECT-COMPUTER.  6000 WITH EIS.
                 INPUT-OUTPUT SECTION.
                 FILE-CONTROL.
                 SELECT IDS ORDER-CONTROL ASSIGN TO A1.
                 DATA DIVISION.
                 WORKING-STORAGE SECTION.
                 .
                 . Your IDS Section goes here
                 .
                 PROCEDURE DIVISION.
                 PARAGRAPH-001.
                         STOP RUN.
          $      SET     18,19
          $      ENDJOB
          ***EOF
```

NOTE: Replace "NAME$PASSWORD" with your USERID and PASSWORD.

EXHIBIT IDS043-7 (Cont)

"PART-SALES" DATA BASE

Use the structure diagram provided to code the IDS section of the Data Division; i.e., MD, 01, 02, and 98 level entries.



Record Name	Data Field Names	Picture
ENTRY1	field for a reference code	9(8)
ENTRY2	field for a reference code	9(8)
LOCATION	DISTRICT-NO	9
EMPLOYEE	NAME	X(18)
	EMPLOYEE-NO	X(5)
	FIXED-SALARY	999V99
	YTD-SALES	9(7)V99
	YTD-WAGE	9(5)V99
	SSAN	9(9)
SALE	SALES-QTY	9(5)
	PCT-COMMISSION	99
PART	PART-NO	9(8)
	PART-DESC	X(12)
	COST	9(4)V99

EXHIBIT IDS043-8

DATA BASE CREATION

To create a permanent IDS file/subfile, the options listed on the "FC/IDS/" card are:

1. CAT/FILENAME string
   To name the file to be created
2. "W" or "M"
   What permissions the file is to have.
   NOTE: Default, no permissions
3. LLINKS or BLOCKS/XX,XX/, LINKS or SIZE
   How large the file will be.
   NOTE: An IDS file can't grow.
4. MODE/RAND/
   An IDS file/subfile must be random.
   NOTE: Default is "RAND"
5. FCLASS/UZZ/
   Or whatever classification is required on WWMCCS Data Bases.
6. ACCESS/MONITOR/
   Provides FMS file protection features.
7. ABORT/ROLLBACK/
   Provides multi-user file protection and journalization.
8. BASESIZE/YYY/
   How many pages the entire "DB" is to contain.
9. PAGESIZE/RRR/
   Indicates how many words each physical page should contain.
   NOTE: Default 320, minimum 64, maximum 640
10. RNG/AA,BBB/
    Specifies how many of the "DB" pages are to be attributed to this GCOS file.
    Any range less than basesize implies that subfiling techniques are being used.
11. LINESPERPAGE/KK/
    States how many of the available 63 user lines will be attributed to each physical page.
    Anything < 63 creates "pagettes".
    NOTE: Default 63
12. INVENTORY/FF/or INVENTORY/NO/
    States at what percentage page fill, on any page in this subfile, inventory will be maintained and updated.
    If no inventory is desired, "NO" should be listed.
    NOTE: Default 75%
13. AREA/##/
    If this subfile is to exist in a particular area
    NOTE: Default Ø

EXHIBIT IDS050-1

```
                              DATA BASE ATTRIBUTES

    4 FILES ALLOCATED,                             BUFFERS  15

AREA #  10  RNG      1 - 29     BASESIZE - 29

AREA #  15  RNG      1 - 29     BASESIZE - 29

                                                         LLINKS  LLINKS  ACCESS
FILCD/AREA   RANGE    PAGESIZE  PAGES/PAGE  LINES/PAGE  ALLOC   NEC     MODE    INV

  A1/10      1 - 19     128         1           63        12      8      TEMP    90%

  B1/10     20 - 29      64         1           63        12      3      TEMP    90%

  A2/15      1 - 19     128         1           63        12      8      TEMP    90%

  B2/15     20 - 29      64         1           63        12      3      TEMP    90%
```

EXHIBIT IDS050-2

```
                       TOTAL I/O PERFORMED ON DATA BASE

FILECODE   # OF READS   # OF WRITES   INVENTORY READS   INVENTORY WRITES

   A1         1666          145              1                  1

   B1         1054           70              1                  1

   A2         1681          166              1                  1

   B2         1071           77              1                  1
```

EXHIBIT IDS050-3

I/O PERFORMED ON DATA BASE AS A FUNCTION OF EACH IDS SUBROUTINE

SUBROUTINE STATISTICS

NAME	# TIMES CALLED	# READS	# WRITES
.QSTOR	592	5211	314
.QGET	726	119	127
.QCHN	426	27	3
.QHEAD	360	115	10
.QMOVE	426	0	0
.QSET	6	0	0

EXHIBIT IDS050-4

.QSTC JCL

```
                  1
1        8        6

$        IDENT
$        USERID

    PROGRAM

$        USE      .QSTC
$        EXECUTE
$        PRMFL
$        DATA     .Q

IDS      OPTION   PRTREC       TRACE       ,ONFC/file-code
                  TRACE        PRTREC

         ETC    , null       TYPES/rec-type-1,...rec-type-50
                  ALL        VERBS/verb-1,...verb-10/
                  DO         RNG/1B,1E,...5B,5E/
                  DONTDO

$        ENDJOB
```

EXHIBIT IDS050-5

SAMPLE OF .QSTC OPTIONS

```
DIRECTIVE:  IDS   OPTION  PRTREC,TRACE,DO/TYPES/010,020/,

            ETC   DO/VERBS/MODIFY,STORE/,

            ETC   DONTDO/RNG/9,19/
```

EXHIBIT IDS050-6

SAMPLE OUTPUT

```
***** IDS-TRACE * ALTER NO. - 395   CTYPE-STOR   RTYPE-  10 AREA/PG/LN 0/ 1/ 6

                                    DISTRICT-NO  STA

STOR  DISTRICT             1/ 6     1             N

***** IDS-TRACE * ALTER NO. - 407   CTYPE-STOR   RTYPE-  20 AREA/PG/LN 0/ 7/ 6

                                    DISTRICT-NO MAT NAME              STA SALESMAN-NO RDM

STOR  SALESMAN             7/ 6     1             N BAILEY, BEETLE B.  AN A0010       AN

                                    FIXED-SALARY    YTD-SALES           YTD-WAGE

                                    50000        N  000000000        N  000000000    N

                                    SSAN

                                    121912102    N

                                           .

                                           .

                                           .

***** IDS-TRACE * ALTER NO. - 617   CTYPE-MDFY   RTYPE-  20 AREA/PG/LN 0/ 8/25

                                    DISTRICT-NO MAT NAME            * STA SALESMAN-NO RDM

MDFY  SALESMAN             8/25     2             N  SMITH, LINDA    AN  A0230       AN

                                    FIXED-SALARY    YTD-SALES

                                    60000        N  000000000       N  000000000    N

                                    SSAN

                                    834345481    N
```

EXHIBIT IDS050-7

SLAVE I-D-S CONTROL TABLE

0	13	TOTAL CONTROL ENTRY
0	13	FILE/SUBFILE CONTROL ENTRY
0		OTHER SUBFILES CONTROL ENTRIES

EXHIBIT 060-1

TOTAL CONTROL ENTRY

0 17	18 35
POINTER TO CURRENT ENTRY	MBZ
MAXIMUM PAGE SIZE	BASE SIZE
LOWEST PAGE NUMBER	HIGHEST PAGE NUMBER
MAXIMUM INVENTORY SECTOR	PAGE BUFFER SIZE
MBZ | HIGHEST AREA NUMBER |	MBZ
MBZ	
PERM FILE INDICATOR	
COEXISTENT FILE COUNT	
CONFIGURATION INDICATOR	MULTI-ACCESS INDICATOR
MBZ	COUNT OF ENTRIES
JOURNAL INDICATOR	MBZ
TSS INDICATOR	
	NON-ZERO IF ACCESS/MONITOR/
	NON-ZERO IF ABORT/ROLLBACK/

EXHIBIT IDS060-2

SUBFILE CONTROL ENTRY

0	RANGE R1	RANGE R2
1	INVENTORY WRITE CONTROL COUNTER	PAGE SIZE
2	PAGES/PAGE NO	LINES PER PAGE
3	MBZ ꞁ AREA NUMBER ꞁ MBZ	INVENTORY PERCENT FILL
4	RELAT!VE BLOCK ADDRESS OF CURRENT PAGE	
5	SECTORS/PAGE	SECTOR SIZE
6	GROSS WRITE COUNTER	GROSS READ COUNTER
7	INVENTORY READ COUNTER	FILCB - 0
8	FILCB -5	FILCB-1
9	ACCESS MODE/SUBFILE ENTRY NUMBER	FILCB-4
0 1	IDS SUBFILE NAME	
2	RELATIVE BLOCK ADDRESS CURRENT INVENTORY	
3	BASE RELATIVE BLOCK ADDRESS - INVENTORY	

EXHIBIT 060-3

AREA TABLES

0	MIN PAGE AREA 0	MAX PAGE AREA 0	AREA RANGE TABLE
1	MIN PAGE AREA 1	MAX PAGE AREA 1	
2	MIN PAGE AREA 2	MAX PAGE AREA 2	
⋮	⋮		
n	MIN PAGE AREA n	MAX PAGE AREA n	
o	BSSZ AREA o	MBZ	AREA BASE SIZE TABLE
1	BSSZ AREA 1	MBZ	
2	BSSZ AREA 2	MBZ	
⋮	⋮		
n	BSSZ AREA n	MBZ	

EXHIBIT IDS060-4

PAGE BUFFER ACTIVITY TABLE

BUFFER NUMBER	PRIOR	NEXT
0		
1		
2		
. . .		
N		

EXHIBIT IDS060-5

PAGE BUFFER

BUFFER	PAGE
4	5
3	4
2	3
1	2
0	1

Retrieve Records Sequentially On Pages 1 thru 5

BUFFER	PAGE
1	2
4	5
3	4
2	3
0	1

Retrieve Record On Page 2

BUFFER	PAGE
0	6
1	2
4	5
3	4
2	3

Retrieve Record On Page 6.

EXHIBIT IDS060-6

DATA PAGE BUFFER

WORD	
-1 7	PAGE HEADER WORK AREA
8 19	JOURNAL TAPE HEADER
20 N	DATA PAGE

EXHIBIT 060-7

PAGE HEADER WORK AREA

1	SICT ENTRY ADDRESS THIS PAGE		MBZ
0	POINTER TO NEXT BUFFER		BUFFER NUMBER
1	MBZ	AREA NUMBER	BEGINNING REF-CODE OF IDS PAGE
2	MBZ	AREA NUMBER	ENDING REF-CODE OF IDS PAGE
3	MBZ		CHARACTER SPACE AVAILABLE
4	AVAILABLE LINE FLAG INDICATOR		
5	VERIFY INDICATOR		
6	RELATIVE BLOCK ADDRESS THIS PAGE		
7	DOUBLE BUFFERING INDICATOR FOR RETRIEVE SERIAL		

EXHIBIT IDS060-8

JOURNAL TAPE HEADER

			17	18		35
8 9 10	WORK AREA FOR GFRC I/O CONTROL					
11	SIZE			ACCOUNTING RECORD TYPE		
12	CHECKSUM					
13	JOB NUMBER					
14	START DATE					
15	START TIME					
16	MBZ	JOURNAL RECORD TYPE	MBZ	AREA NUMBER	MBZ	ACTIVITY NUMBER
17	LINES/PG FOR B/A PAGE			SEQUENCE NUMBER		
18 19	IDS SUBFILE NAME					

EXHIBIT 060-9

INVENTORY PAGE BUFFER

-2	SICT ENTRY ADDRESS THIS PAGE			MBZ
-1	RELATIVE BLOCK ADDRESS THIS PAGE			
0	POINTER TO NEXT BUFFER		V F Y	BUFFER NUMBER
1	MBZ	AREA NUMBER	BEGINNING REFERENCE CODE	
2	MBZ	AREA NUMBER	ENDING REFERENCE CODE	
	INVENTORY RECORD AREA			

EXHIBIT IDS060-10

CORE USAGE

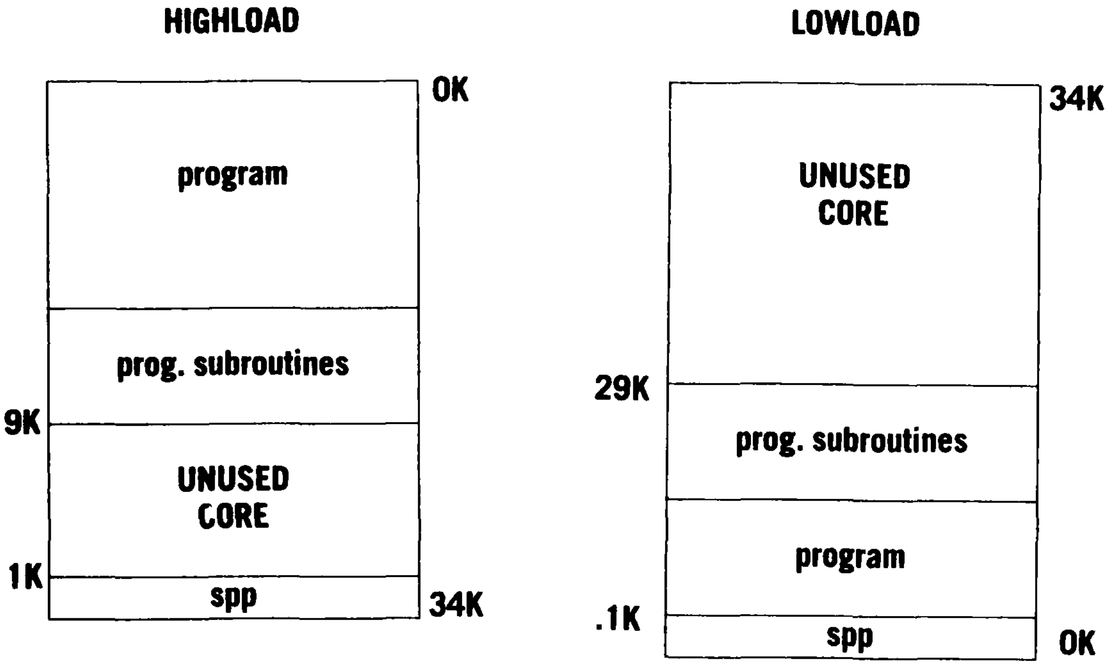


EXHIBIT IDS060-11

```
$          USE .QMAX/1/,.QAREA/N/,.QMIN/1/                FOR HIGHLOAD
$          USE .QMIN/1/,.QAREA/N/,.QMAX/1/                FOR LOWLOAD
```

WHERE "N" =

14 (NF + 1) + NB + (HA + 1) * 2 + (NB - NBO) * (MPS + 21)

GIVEN:

NF = NUMBER OF IDS FILES ALLOCATED

NB = TOTAL NUMBER OF BUFFERS DESIRED
HA = HIGHEST AREA NUMBER ALLOCATED
NBO = NUMBER OF BUFFERS OVERLAYING .QOPE
MPS = LARGEST PAGE SIZE OF THE FILES ALLOCATED

and NB ≥ NBO

HA ≥ 1 ELSE THE MODULE = 0

EXHIBIT IDS060-12

SCHEMA

(IDS-DEFINITION SECTION)

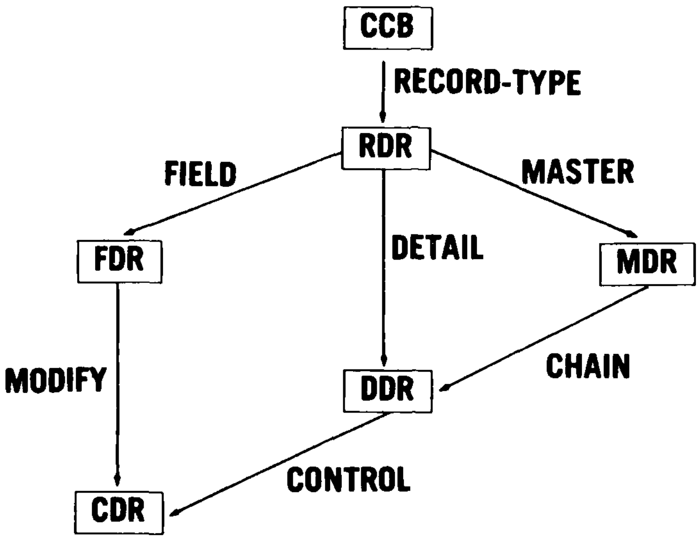


EXHIBIT IDS 070-1

COMMUNICATIONS CONTROL BLOCK

<table>
<tr><td>CCBLOXK</td><td>00</td><td>AREA</td><td colspan="3">DIRECT-REFERENCE</td></tr>
<tr><td>+1</td><td>MBZ</td><td>AREA</td><td colspan="3">FIRST-REFERENCE</td></tr>
<tr><td>+2</td><td>MBZ</td><td>AREA</td><td colspan="3">LAST-REFERENCE</td></tr>
<tr><td>+3</td><td colspan="2">MBZ</td><td colspan="3">RECORD-TYPE</td></tr>
<tr><td>+4</td><td colspan="3">RECORD TYPE CHAIN NEXT</td><td>MBZ</td><td>FILE-CODE</td></tr>
<tr><td>+5</td><td colspan="3">MBZ</td><td colspan="2">ERROR-REFERENCE</td></tr>
<tr><td>+6</td><td colspan="3">MBZ</td><td>AUTHORITY</td><td>OPEN MODE</td></tr>
</table>

EXHIBIT IDS070-2

RECORD DEFINITION RECORD

NAME RECORD	01	MBZ	RECORD TYPE	RECORD SIZE	D	V	A	S	P	R
+1	PAGE INTERVAL			MASTER CHAIN NEXT						
+2	FIELD CHAIN NEXT			DETAIL CHAIN NEXT						
+3	MBZ	AREA NUMBER	CURRENT RECORD REFERENCE CODE							
+4	RECORD TYPE CHAIN NEXT			AUTHORITY	MBZ					
+5	MINIMUM PAGE RANGE			MAXIMUM PAGE RANGE						
+6	AREA DEFINITION RANGE			MBZ						

EXHIBIT IDS070-3

MASTER DEFINITION RECORD

LOC-SYM	02	MBZ	DATA RECORD TYPE	MASTER CHAIN HEAD
+1	CHAIN CHAIN NEXT			MASTER CHAIN NEXT
+2	MBZ	AREA NUMBER	REFERENCE CODE OF CHAIN MASTER	
+3	NEXT POSITION		REFERENCE CODE OF CHAIN PRIOR	
+4	PRIOR POSITION		REFERENCE CODE OF CHAIN CURRENT	
+5	MBZ		REFERENCE CODE OF CHAIN NEXT	
+6	MBZ		REFERENCE CODE OF KEY RECORD	

EXHIBIT IDS070-4

DETAIL DEFINITION RECORD

LOC-SYM	04	MBZ	DATA RECORD TYPE	MBZ	ORDER	DUP	C	U	S	R
+1	CHAIN CHAIN NEXT			CONTROL CHAIN NEXT						
+2	CHAIN CHAIN HEAD			DETAIL CHAIN NEXT						
+3	NEXT POSITION		MBZ	DETAIL CHAIN HEAD						
+4	PRIOR POSITION		HEAD POSITION			MBZ				

EXHIBIT IDS070-5

FIELD DEFINITION RECORD

LOC-SYM	10	MBZ	C	A	F	U	MBZ	FIELD INCREMENT	
+1	LOCATION OF WORKING STORAGE		FIELD SIZE					M B Z	FIRST CHAR
+2	FIELD CHAIN NEXT		MODIFY CHAIN NEXT						

EXHIBIT IDS070-6

CONTROL DEFINITION RECORD

LOC-SYM	02	MBZ	R	CNTL	CONTROL CHAIN HEAD
+1	LOCATION OF MATCH-KEY FIELD DEFINITION				CONTROL CHAIN NEXT
+2	MODIFY CHAIN HEAD				MODIFY CHAIN NEXT

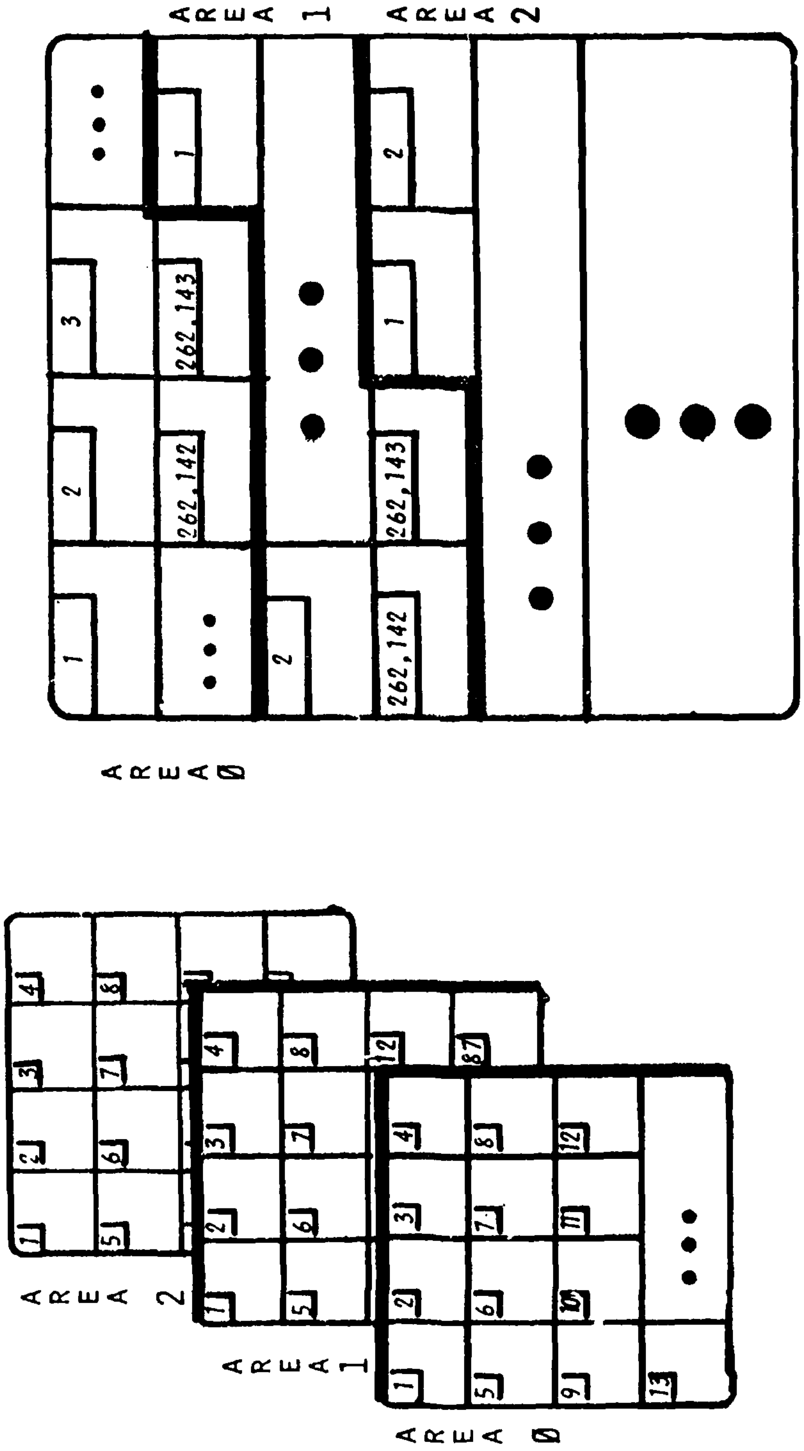
EXHIBIT 070-7

CHAIN

EACH CHAIN THE RECORD IS *MASTER* IN		
Master	CURRENT OF SYSTEM	
Prior	0	PRIOR POINTERS
		SYSTEM KNOWS FROM PROCESSING
Current	CURRENT OF SYSTEM	
Next	CURRENT OF SYSTEM +1	

EACH CHAIN THE RECORD IS *DETAIL* IN		
Master	0	SYSTEM KNOWS FROM PROCESSING
		MASTER POINTERS
Prior	0	SYSTEM KNOWS FROM PROCESSING
		PRIOR POINTERS
Current	CURRENT OF SYSTEM	
Next	CURRENT OF SYSTEM +1	

EXHIBIT IDS070-8



SINGLE DATABASE

MULTIPLE DATABASES

EXHIBIT IDS080-1

MULTI-AREA CODING ASPECTS

```
1)   01  REC-NAME       TYPE IS 010                    2)   PROCEDURE DIVISION.
                        RETRIEVAL VIA ...
                                                                    .
                        WITHIN AREA 2.                              .
                                                                    .
                .
                .
                .                                           MOVE 17 TO AREA-CONTROL.

     01  ANOTHER-REC    TYPE IS 745                         ENTER IDS.
                        RETRIEVAL VIA ...                      SET AREA TO AREA-CONTROL.
                        WITHIN AREA AREA-CONTROL.                   .
                                                                    .
                                                                    .


3)  TEMPORARY FILES - IDS  CREATE  FC/A1/,BSSZ/1750/, ... AREA/17/

    PERMANENT FILES - FC/IDS/ CAT/FILE-STRING,W,BASESIZE/2300/, ... AREA/2/
```

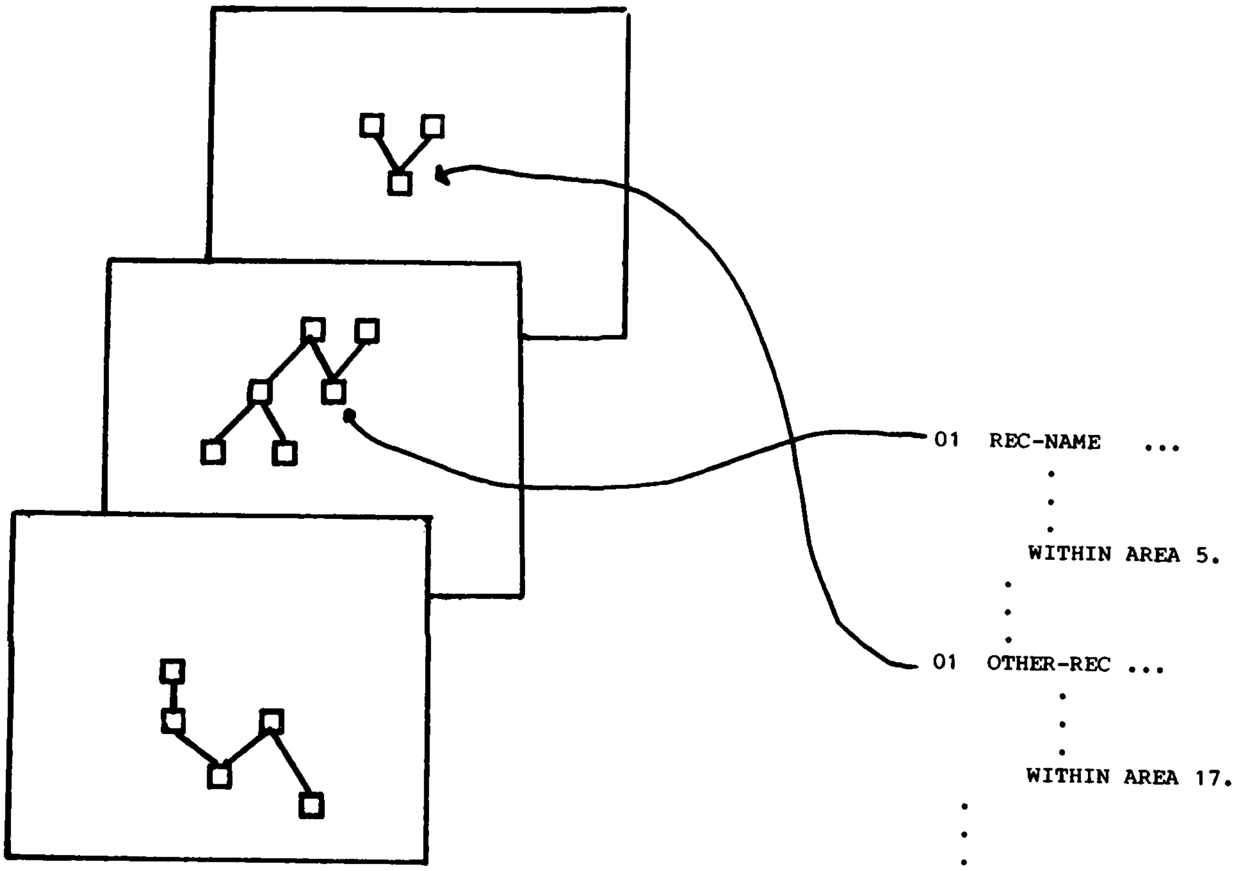
EXHIBIT IDS080-2

FORCING FILES TO ANOTHER AREA

```
$   FILSYS
FC/IDS/   MDBD/AREA4A,W,BASESIZE/10000/,RNG/1,2000/,...AREA/4/
FC/IDS/   MDBD/AREA4B,W,BASESIZE/10000/,RNG/2001,4000/,...AREA/4/

                             .
                             .
                             .
FC/IDS/   MDBD/AREA12A,W,BASESIZE/15000/,RNG/1,650/,...AREA/12/
FC/IDS/   MDBD/AREA12B,W,BASESIZE/15000/,RNG/651,800/,...AREA/12/

                        .
                        .
                        .

+ = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + =


$     IDENT
   .
   .
   .
$     PRMFL     A1,W,R,MDBD/AREA4B
$     PRMFL     F1,W,R,MDBD/AREA12B
         .
         .
         .
$     DATA      .Q
IDS   CREATE    FC/A1/,AREA/0/
IDS   CREATE    FC/F1/,AREA/0/
                 .
                 .
                 .
```

EXHIBIT IDS080-3

WITHIN AREA <INTEGER - 1>



```
01  REC-NAME   ...
         .
         .
         .
      WITHIN AREA 5.
    .
    .
    .
01  OTHER-REC ...
         .
         .
         .
      WITHIN AREA 17.
 .
 .
 .
```

EXHIBIT IDS080-4

WITHIN AREA DATA-NAME

```
77  AREA-CONTROL   PIC 99 USAGE COMP-1 VALUE 0.
                .
                .
                .
01  REC-NAME   ...
      .
      .
      .
    WITHIN AREA AREA-CONTROL.
                .
                .
                .
```

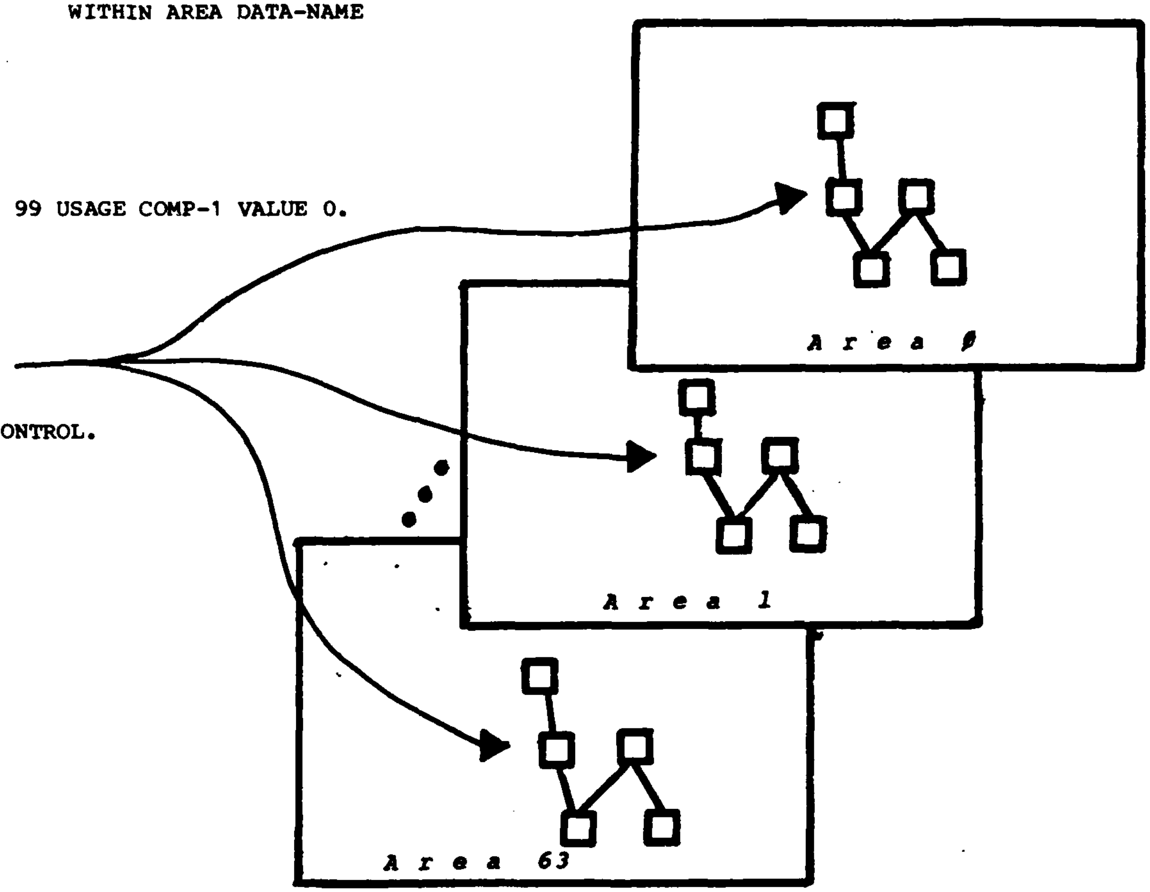


EXHIBIT IDS080-5

INTERNAL REVENUE SERVICE (IRS) MULTI-AREA APPLICATION

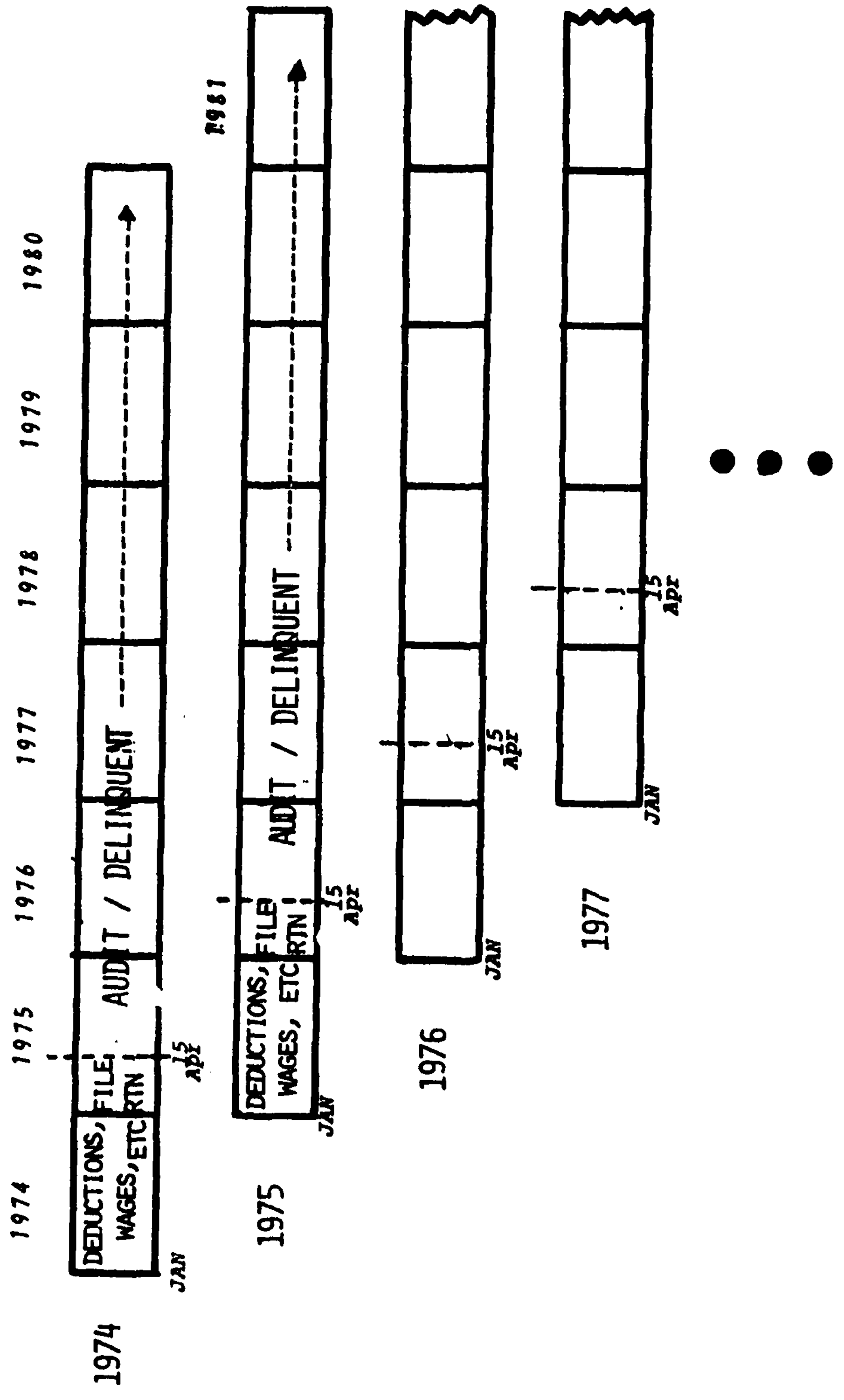


EXHIBIT IDS080-6

*** WITH TEMPORARY FILES ***

```
$       IDENT
$       USERID
   (QUTI OR QUTL ACTIVITY)
   **  PROGRAM  **
$       EXECUTE
$       FILE    A1,X1S,10R
$       FILE    A2,X2S,10R
$       FILE    A3,X3S,10R
$       FILE    A4,X4S,10R
$       DATA    .Q
IDS     CREATE  FC/A1/,BSSZ/240/,RNG/1,240/
IDS     CREATE  FC/A2/,BSSZ/240/,RNG/1,240/,AREA/1/
IDS     CREATE  FC/A3/,BSSZ/240/,RNG/1,240/,AREA/2/
IDS     CREATE  FC/A4/,BSSZ/240/,RNG/1,240/,AREA/3/
$       ENDJOB
```

*** WITH PERMANENT FILES ***

```
$       PRMFL   A1,W/C,R,IRS/YR1974
$       PRMFL   A2,W/C,R,IRS/YR1975
$       PRMFL   A3,W/C,R,IRS/YR1976
$       PRMFL   A4,W/C,R,IRS/YR1977
```

(** AREAS ASSIGNED WITH FCREATE IN PREVIOUS FILSYS ACTIVITY **)

EXHIBIT IDS080-7

DYNAMIC AREA SWITCHING

```
DATA DIVISION.
          .
          .
          .
WORKING-STORAGE SECTION.
77      TAX-YEAR        PICTURE 99 USAGE COMP-1 VALUE 0.
          .
          .
          .
IDS SECTION.
MD      DATABASE.

01      INDIVIDUAL                          TYPE IS 001
                                            RETRIEVAL VIA RETURN-CHAIN CHAIN
                                            WITHIN AREA TAX-YEAR.
        02 ...
98      RETURN-CHAIN CHAIN DETAIL ...

01      INCOME                              TYPE IS 002
                                            RETRIEVAL VIA ...
                                            WITHIN AREA TAX-YEAR.
        02 ...
98      ...

01      BUSINESS                            TYPE IS 003
                                            RETRIEVAL VIA ...
                                            WITHIN AREA TAX-YEAR.
        02 ...
98      ...
          .
          .
          .
PROCEDURE DIVISION.
          .
          .
          .
        ENTER IDS. SET AREA TO TAX-YEAR.
          .
          .
          .
```

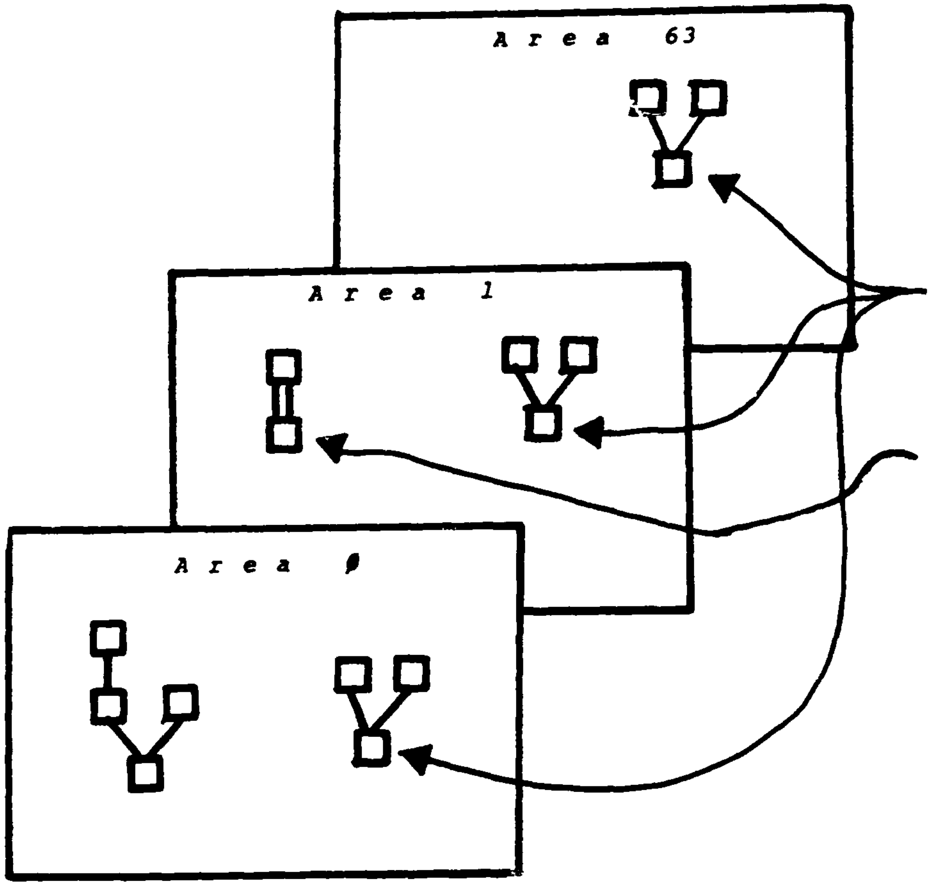
EXHIBIT IDS080-8

FIXED AREA CONTROL

```
DATA DIVISION.
      .
      .
      .
MD    DATABASE.
01    INDIVIDUAL-74                  TYPE IS 010
                                     RETRIEVAL VIA ...
           .
           .
           .
01    INDIVIDUAL-76                  TYPE IS 020
                                     RETRIEVAL VIA ...
                                     WITHIN AREA 2.
      .
      .
      .
01    INDIVIDUAL-77                  TYPE IS 030
                                     RETRIEVAL VIA ...
                                     WITHIN AREA 3.
         .
         .
         .
PROCEDURE DIVISION
    .
    .
    .
     MOVE 1 TO AREA-CONTROL.
     ENTER IDS.  SET AREA TO AREA-CONTROL.
     ENTER IDS. STORE INDIVIDUAL-75 IF ERROR ...
     MOVE 0 TO AREA-CONTROL.
     ENTER IDS.  SET AREA TO AREA-CONTROL.
     ENTER IDS.  STORE INDIVIDUAL-74 IF ERROR ...
                   .
                   .
                   .
```

EXHIBIT IDS080-9

FIXED/DYNAMIC AREA CONTROL DESIGN



```
77  AREA-CONTROL PIC 99 USAGE COMP-1.
        .
        .
        .
01  REC-NAME          TYPE IS ...
                      RETRIEVAL VIA ...
                      WITHIN AREA AREA-CONTROL.
      .
      .
      .
01  ANOTHER-REC TYPE IS ...
                      RETRIEVAL VIA ...
                      WITHIN AREA 1.
          .
          .
          .
```

EXHIBIT IDS080-10

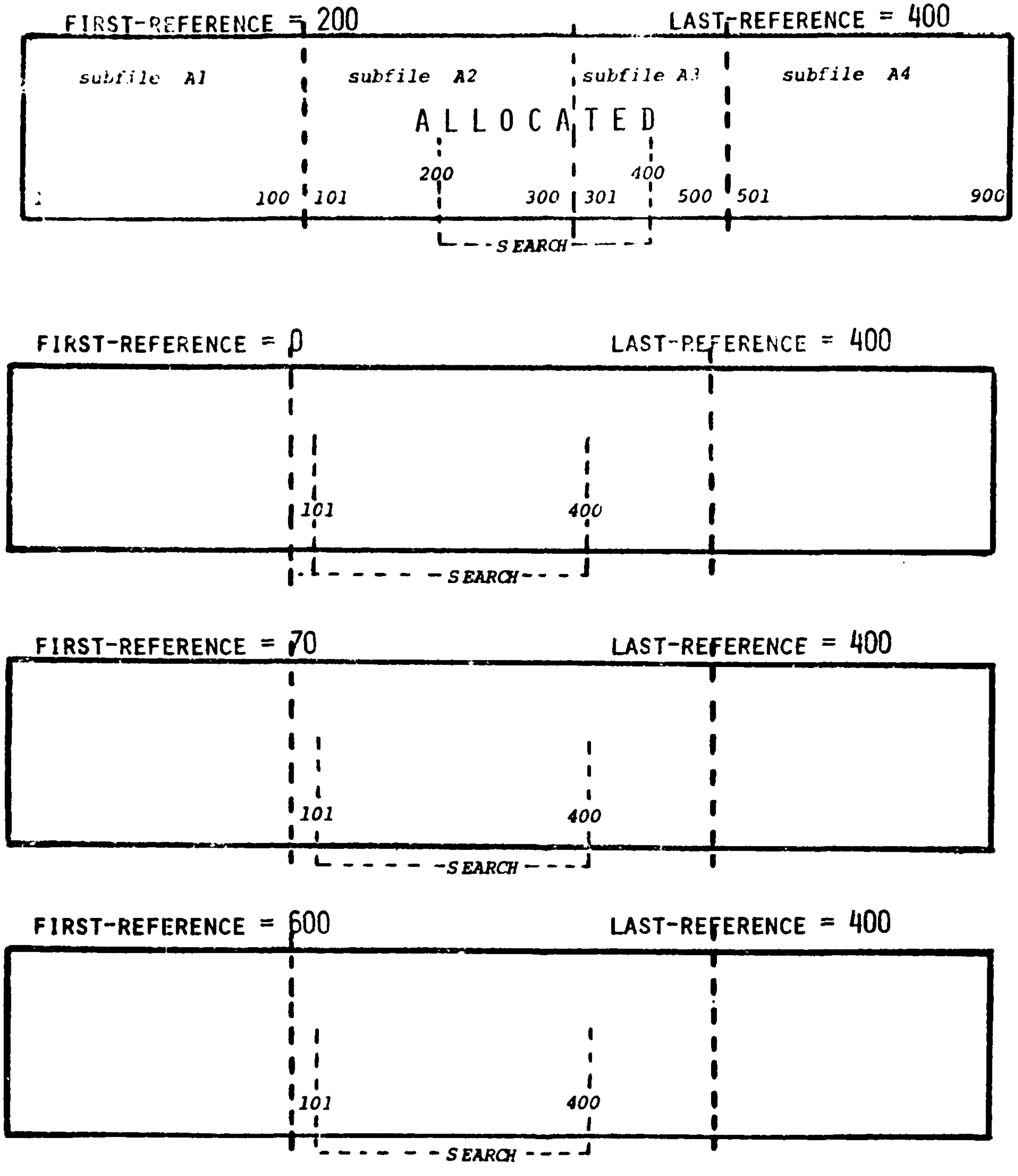


EXHIBIT IDS080-11

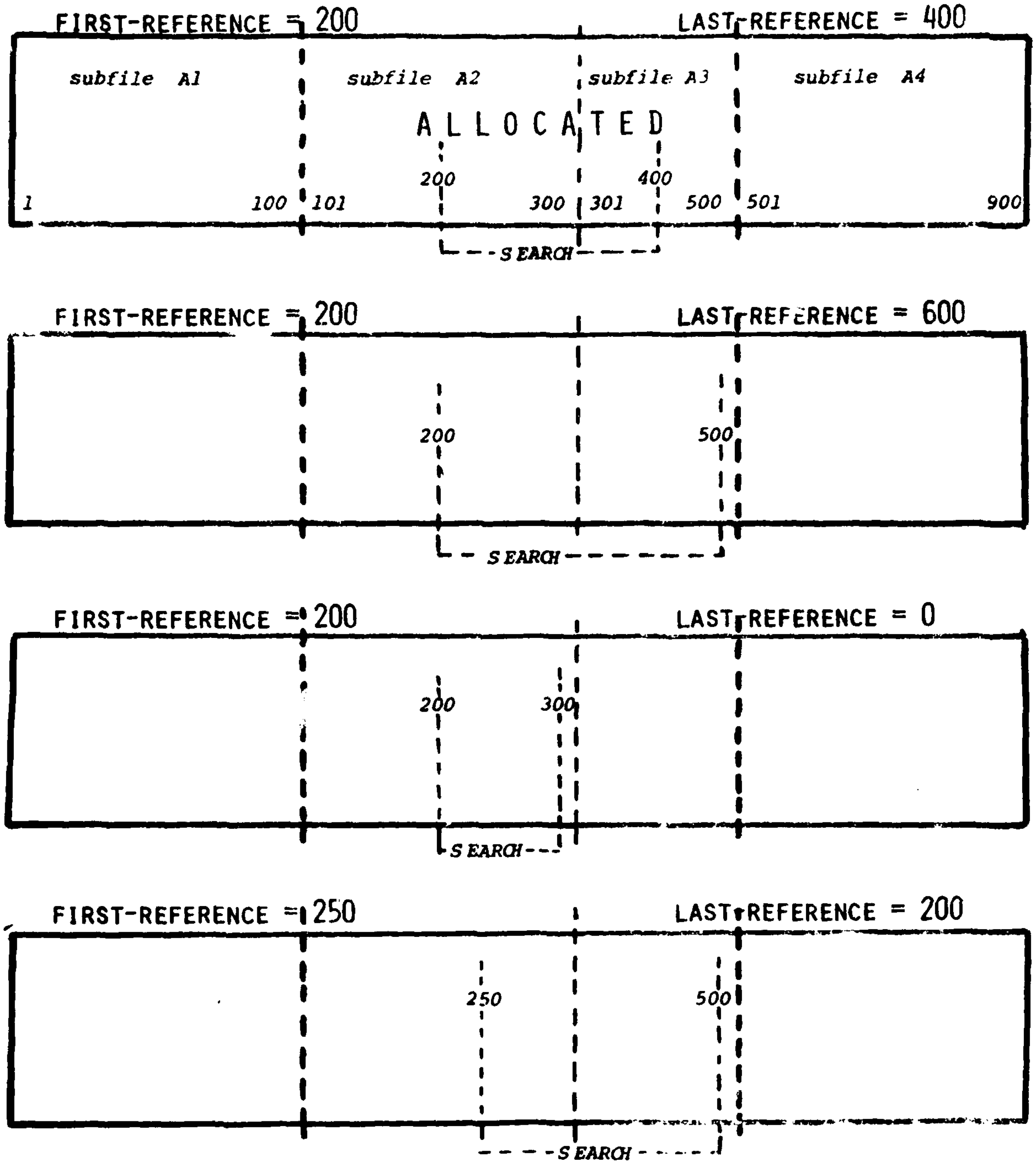


EXHIBIT IDS080-12

CALLING RETRIEVE SERIAL SUBROUTINE

```
ENTER GMAP.

CALL .QGETS (ARG1,ARG2)

ENTER COBOL.
```

WHERE ARG1: SPECIFIES THE ADDRESS OF THE LIST OF RECORD TYPES TO BE RETRIEVED.

ARG2: AN ADDRESS TO WHERE THE PROGRAM IS TO GO ONCE THE SEARCH HAS BEEN COMPLETED.

EXHIBIT IDS080-13

```
PROCEDURE DIVISION.
      .
      .
      .
GET-RECORD.
ENTER GMAP.
    CALL .GETS ( ARG-1, ...)
ENTER COBOL.
     .
     .
     .
    GO TO GET-RECORD.
STOP RUN.
ENTER GMAP.
ART-1          DEC    (# OF RECORD TYPE ENTRIES THAT FOLLOW,
               VFD
                .
                .
                .
```

EXHIBIT IDS080-14

```
PROCEDURE DIVISION.
     .
     .
     .
GET-RECORD.
ENTER GMAP.
     CALL .QGETS (ARG-1, ...)
ENTER COBOL.
     .
     .                                                            returns to
     .
     GO TO GET-RECORD.
STOP RUN.
ENTER GMAP.
ARG-1          DEC    4
               VFD    18/0,6/0,12/23
               VFD    18/0,6/0,12/410
               VFD    18/0,6/0,12/500
               VFD    8/0,6/0,12/620
```

EXHIBIT IDS080-15

```
PROCEDURE DIVISION.
     .
     .
     .
ENTER DEFINITIONS.
     SYMBOL   GTSIT   =   PROCEDURE   PRINT-SPECIAL-MESSAGE.
        .
        .
        .
GET-RECORD.
ENTER GMAP.
     CALL .QGETS (ARG-1,...)
ENTER COBOL.
ENTER IDS.   ....
     DISPLAY "TYPE 23,410 OR 500 RETRIEVED".
     GO TO GET-RECORD.
PRINT-SPECIAL-MESSAGE.
     DISPLAY "TYPE 620 RETRIEVED".
     GO TO GET-RECORD.
          .
          .
          .
STOP RUN.
ENTER GMAP.
ARG-1          DEC    4
               VFD    18/0,6/0,12/23
               VFD    18/0,6/0,12/410
               VFD    18/0,6/0,12/500
               VFD    18/GTSIT,6/1,12/620

returns to
```

EXHIBIT IDS080-16

```
PROCEDURE DIVISION.
         .
         .
         .
ENTER DEFINITIONS.
         SYMBOL  GTSIT  =  PROCEDURE PRINT-SPECIAL-MESSAGE.
         SYMBOL  ENDIT  =  PROCEDURE  SEARCH-COMPLETE.
ENTER COBOL.
         COMPUTE FIRST-REFERENCE = 1 * 64.
         COMPUTE LAST-REFERENCE  = 500 * 64.
GET-RECORD.
ENTER GMAP.
         CALL .QGETS (ARG-1,  ENDIT  ).
ENTER COBOL.
ENTER IDS. ....
         DISPLAY "TYPE 23,410 OR 500 RETRIEVED".
         GO TO GET-RECORD.
PRINT-SPECIAL-MESSAGE.
         DISPLAY "TYPE 620 RETRIEVED".
         GO TO GET-RECORD.
SEARCH-COMPLETE.
         DISPLAY "SEARCH COMPLETED".
                     .
                     .
                     .
STOP RUN.
ENTER GMAP.
ART-1 ...
```

EXHIBIT IDS080-17

```
PROCEDURE DIVISION.
     .
     .
     .
ENTER DEFINITIONS.
     SYMBOL   HALTI    =   PROCEDURE  STOP-IMMEDIATELY.
               .
               .
               .
ENTER COBOL.
     .
     .
     .
GET-RECORD.
ENTER GMAP.
     CALL .QGETS (ARG-1,ENDIT).
ENTER COBOL.
ENTER IDS.
     IF ERROR PERFORM ERROR-ROUTINE.
     GO TO GET-RECORD.
          .
          .
          .
STOP-IMMEDIATELY.
ENTER GMAP.
     CALL .QGETS (0,  EARLY )
       .
       .
       .
STOP RUN.
ENTER GMAP.
ARG-1          DEC    1
               VFD    18/HALTI,6/1,12/443
EARLY          DEC    1
               VFD    18/0,18/-1
```

EXHIBIT IDS080-18

$ USE OPTIONS

```
$     USE     .QBREV,QBIT1/1/,.QBITS/N/,.QBIT2/1/

$     USE     .QDLN1/1/,.QNODL/1/,.QDLN2/1/

$     USE     .QERT    OR    .QEUP

$     USE     .QMIN/1/,.QAREA/N/,.QMAX/1/

$     USE     .QPGB1/1/,.QBUF/N/,.QPGB2/1/

$     USE     .QSTB

$     USE     .QSTC

$     USE     .QTCP
```

EXHIBIT IDS080-19

QUTA

```
1      8         16

                                  RNG/PAGE-1,PAGE-2/
IDS    OPTION    STATS/       ,   GRAPH/ N/
                                  TYPE/  T1  ,...T10          ...
                                         ALL
```

EXHIBIT IDS090-1

QUTC

```
1      8         16
                                        RNG/PAGE-1,PAGE-2/
                                        MAX/NNNNNN/
                                ANAL/ , INCR/NNNNN/
IDS    OPTION    GENERATE/,             FILL/NN/

                                          RNG/PAGE-1,PAGE-2/
                                  RAND    FILL/NN/                  ...
                                  RANDA   CF/BG,1C/
                                            ,CF/BG,2C/
                                           ,CF/BG,3C /                 ...
```

EXHIBIT IDS090-2

QUTD

```
1       8       16
        DUMP     RNG/PAGE1,PAGE2/
         PRINT                            NULL
IDS      PDUMP   RNG/PAGE1,PAGE2/         EMPTY
         DPRINT                         , TYPES/A1 ,...A10  /
                                          DELETE                     ...
                                                                         ...
        EOR
```

EXHIBIT IDS090-3

QUTI

```
8          16
INITIAL
           PAGE-1, PAGE-2
HEADER
```

EXHIBIT IDS090-4

```
                                         QUTL


                           LOAD
                           PRINT
               DTAPE                                      EMPTY
IDS    OPTION        /,      LPRINT  /   ,RNG/pg1,pg2/   ,TYPES/a1 ...a10 /  ...
               STAPE         PLOAD                        DELETE
                                                                                 ...
                           EOR

               DTAPE       ,REFORMAT/


                                                pg1,pg2/
                                          RNG   /pg1,pg2,AREA,n/
                                                /0,AREA,n/

                           LOAD           FILE/subfile-name  YYMMDD  ,HH,TTT   /
                           PRINT               ,PAGE/pg1 ...pg10   /

IDS    OPTION  JTAPE/,       LPRINT  /  , CLEANPOINT/YYMMDD ,HH,TTT  /
                             PLOAD
                                          SNUMB/sssss/   ,ACT/a1 ,a2 /

                                            AFTER
                                            BEFORE
                                                                                 ...
                                          TRAN
                           EOR
                                    TYPES/tp1  ...tp10  /
      ETC                  ,NORW/ ,                         ...
                                    DELETE
                                                                   ...
```

EXHIBIT IDS090-5

QUTL DECK SETUP

-LOAD
CAUSES PAGES FROM THE TAPE TO BE WRITTEN ONTO THE PASS STORAGE DEVICE.

- PRINT
CAUSES THE PAGES ON THE TAPE TO BE CONVERTED TO A PRINT FORMAT AND DIRECTED TO SYSOUT, VIA (P*).

- LPRINT, PLOAD
DOES BOTH OF THE ABOVE.

- REFORMAT (D TAPE ONLY)
USED IN CONJUNCTION WITH LOADING, CAUSES THE TAPE RECORDS TO BE RESTORED, IN LINE NUMBER ORDER, TO THE MASS STORAGE DEVICE, WHERE THE FILE SPACE MAY NOW HAVE A DIFFERENT PAGE SIZE AND/OR LPP THAN WHAT IS DESCRIBED ON THE TAPE.

- FOR
FORCES AN END-OF-REEL CONDITION ON THE INPUT TAPE.

- RNG (OPTIONAL)
DIRECTS WHICH PAGES FROM THE TAPE ARE TO BE EXAMINED.
DEFAULT, ALL ALLOCATED PAGES

- EMPTY (OPTIONAL) (DTAPE, STAPE ONLY)
PRINTS ALL RECORDS WITHIN THE PAGE RANGE (TO INCLUDE PAGE HEADERS).
DEFAULT, EMPTY PAGES ARE ANNOTATED WITH A SUCCESSION ENTRY.

- TYPES (OPTIONAL)
CAN SPECIFY THAT ONLY CERTAIN RECORD TYPES OF PRINTED OUT (MAXIMUM OF 10).
DEFAULT, ALL RECORDS PRINTED.

- DELETE
GENERATES A FILE LISTING WITH REFERENCE CODES, SIZES AND TYPES OF ALL LOGICALLY DELETED RECORDS ON THE TAPE.

NOTE: Doesn't physically remove them from the DB.

- NOFWD (JTAPE ONLY)
SUPPRESSES REWINDING OF THE TAPE AFTER A DIRECTIVE.

- FILE (JTAPE ONLY)
SPECIFIES THE SUBFILE TO BE RESTORED AND, IF PRESENT, THE POINT IN TIME THAT PAGES ARE TO BE CONSIDERED AND RESTORED.

- PAGE (JTAPE ONLY)
CAN SPECIFY UP TO 10 PAGE NUMBERS TO BE SELECTIVELY PRINTED.

EXHIBIT IDS090-6

- CLEANPOINT (JTAPE ONLY)
  SPECIFIES SELECTION OF ALL PAGES THAT FOLLOW AN IDS BEGIN OR CLEANPOINT RECORD

- BEFORE, AFTER (JTAPE ONLY)
  CHOOSES TYPE OF PAGE IMAGES TO BE RESTORED.
  ONE PER DIRECTIVE.
  DEFAULT, BEFORE

- SNUMB (JTAPE ONLY)
  SPECIFIES SELECTION OF PAGE IMAGES OCCURRING FOR THAT SNUMB ONLY.
  IF "ACT/A1/" IS PRESENT, ONLY THAT ACTIVITY WILL BE EXAMINED.
  IF "A1" AND "A2" ARE SPECIFIED, THE RANGE OF ACTIVITIES WILL BE EXAMINED

- TFAN (JTAPE ONLY)
  CREATES A FILE OF THE TRANSACTIONS THAT EXIST ON THE TAPE, BASED ON THE SNUMB, FILE, ETC. OPTIONS SELECTED

.QUTP.

```
1       8       16
                                           B
IDS     SELECT  A1/SSSSS,A2/SSSSS,         A

                B                   B
        ETC     A / L-P-1/H-P-1  ,... A /L-P-8/H-P-8     ...
```

EXHIBIT IDS090-7

.QUTX.

```
                MOVE/PAGE-BEGIN,PAGE-END,OFFSET /

                  TYPE/NNN/
                  RNG/PAGE-BEGIN,PAGE-END/
                  DELETE                            ...

                CF/ CHAR-BEGIN-POS,OLD-FLD-LENGTH,NEW-FLD-LENGTH /

                VAL/.../
IDS     REFORM
                COEXIST

                CHAIN

                PRIOR

                HEAD

                COPY/ BEGIN-CHAR-POSITION /
              BCD
        ADD/        , ± INTEGER /
              BIN
```

EXHIBIT IDS090-8

QUTX DECK SETUP

- MOVE

  BY SPECIFYING THE BEGINNING AND ENDING PAGE NUMBERS AND OFFSET VALUE, PAGES WITHIN THAT RANGE WILL BE SHIFTED THE OFFSET VALUE OF PAGES AND ALL RECORDS ON THOSE PAGES WILL BE CONSIDERED TO RESIDE WITHIN THE NEW PAGE RANGE.

  NOTE: must be to an unused portion of the DB.

- DELETE

  BY SPECIFYING A "TYPE//", ALL RECORDS OF THAT TYPE WILL BE PHYSICALLY REMOVED FROM THE "DB.

  STRESS: NO ALTERING OF CHAINS OR DELINKING TAKES PLACES. IF NOT PLANNED FOR, THE DELETED RECORDS WILL CREATE BROKEN CHAINS.

- RNG (OPTIONAL)

  RECORDS ONLY WITHIN THAT RANGE WILL BE EFFECTED BY THE OTHER OPTIONS. DEFAULT, ALL RECORDS ALLOCATED WILL BE SCANNED.

- CF

  WILL EFFECT CHANGES TO FIELD LENGTH (ADDING/DELETING/CHANGING A FIELD).
  PROPER FIELD LOCATION/INSERTION POINT CAN BE DISCERNED FROM THE IDS STRUCTURE SECTION.
  A FIELD OF > 30 CHARACTERS CAN BE INSERTED BY USING MULTIPLE BACK-TO-BACK "CF" OPTIONS.

- VAL

  CAN INSERT A CONSTANT VALUE INTO A FIELD (LOCATED BY A "CF" CLAUSE).

- COEXIST

  INDICATES THE DTAPE CONTAINS A COEXISTENT FILE AND IS ONLY APPLICABLE W/ "MOVE//".

- CHAIN

  WHEN USED WITH A "CF" CREATE, THE NEW OR MODIFIED FIELD BECOMES A MASTER RECORD POINTER WITHIN THAT RECORD (I.E. GIVEN A VALUE TO POINT AT ITSELF; TO BE USED LATER AS A MASTER RECORD OF A NEW CHAIN OR "ZEROED" IF IT ONCE HAD DETAILS THAT HAVE SINCE BEEN QUTX"D DELETED).

- PRIOR

  SPECIFIES THE NEW FIELD WILL BE A PRIOR POINTER WITH A VALUE [illegible] 777777 (TELLS IDS TO FILL THIS FIELD WITH A "REAL" ADDRESS FROM THE CHAIN TABLE WHEN IT IS ENCOUNTERED DURING NORMAL PROCESSING).

- HEAD

  SAME AS PRIOR EXCEPT THE FIELD IS CONSISDERED A MASTER POINTER FOR THE DETAIL RECORD IT IS IN.

- COPY

  SPECIFIES THE FIELD (LOCATED BY A "CF") WILL BE LOADED WITH THE INFORMATION LOCATED ELSEWHERE IN THE RECORD.

- ADD

  CAN ADD A CONSTANT NUMBER TO A FIELD (LOCATED BY A "CF").

EXHIBIT IDS090-9

.QCHX.

```
                                                  DOALL
RNG      / PG1,PG2 /    ,AREA/A1/                 DOONLY/ CH1,CH2, ... /
RANGE                                             REPORTLONG


                   CHASE
,PRINT                           /CHAIN-NAME,PAGE-NUM,LINE-NUM /
                   CHASEP


                                             PRIOR
,REPAIR/CH1,PAGE-A,LINE-A,                    NEXT        ,PAGE-B,LINE-B /
                                             HEAD
```

EXHIBIT IDS100-1

QCHX DECK SETUP

- RNG OR RANGE (OPTIONAL)
  LIMITS THE SEARCH FOR MASTER RECORDS OF CHAINS TO THIS RANGE OF PAGES. DEFAULT, ALL ALLOCATED PAGES.

  NOTE: ALL DETAILS OF "FO'ND" MASTERS MUST ALSO RESIDE WITHIN THE RANGE.

- AREA
  REQUIRED IF MULTI-AREA FILES ARE BEING PROCESSED.

  NOTE: WILL AUTOMATICALLY EXECUTE A "SET" VERB ACTION FOR ALL RECORD TYPES CONTROLLED BY ANY DATA NAME.

- DOALL
  ALL MASTER RECORDS OF ALL CHAINS ARE PLACED IN THE PROCESSING LIST.

- DOONLY
  ONLY THE MASTER RECORD(S) OF THE SPECIFIED CHAIN(S) ARE PLACED IN THE PROCESSING LIST.

- PRINT
  CHAINS PROCESSED THRU ARE PRINTED OUT WITH RECORD TYPE AND PAGE/LINE NUMBER LOCATION.

- CHASE
  ALLOWS A SPECIFIC CHAIN OCCURRENCE TO BE PROCESSED AND PRINTED. THE PAGE AND LINE NUMBER CAN BE THE ADDRESS OF ANY RECORD IN THAT CHAIN AND THE OUTPUT TERMINATES WITH A COUNT OF THE NUMBER OF RECORDS ON THE CHAIN.

- CHASER
  SAME AS "CHASE" EXCEPT IT PROCESSES THE SAME SPECIFIED BACKWARDS.

  NOTE: TO PRECLUDE EXCESSIVE I/O, "PRIOR" POINTERS SHOULD BE AVAILABLE IN THE CHAIN BEING.

- REPORTLONG (NOT WITH CHASE, CHASEP, REPAIR)
  CAUSES EACH OCCURRENCE OF A CHAIN THAT EXCEEDS A SPECIFIED NUMBER OF DETAILS TO BE OUTPUTTED.

- REPAIR
  PERMITS "REPAIRING" OF CHAINS. BOTH RECORDS ARE RETRIEVED AND THEIR ASSOCIATED POINTERS, OF THE SPECIFIED CHAIN, ARE ADJUSTED TO REFLECT WHETHER THE SECOND RECORD IS TO BE "NEXT" AFTER THE FIRST, "PRIOR" OF THE FIRST OR A MASTER POINTER TO BE FILLED WITH THE ADDRESS OF THE FIRST.

- (COEXISTENCE)
  WHEN NO OPTIONS ARE SPECIFIED (NOT INCLUDING "RNG", "AREA" OR "REPORTLONG"). DEFAULT IS THAT ALL CHAINS/RECORDS ALLOCATED (OR WITHIN THE RANGE/AREA SPECIFIED) WILL BE EXAMINED AND WILL FILL IN "NULL" POINTERS, WHEN THEY ARE FOUND, WITH PROPER ADDRESSES.

EXHIBIT IDS100-2

CREATING THE COPY IDS FILE

```
$       FILSYS

FCREATE/IDS/      CAT/FILE-STRING, MODE/RAND/,FCLASS/---/,
                  AREA/1/,BASESIZE/N/,RNG/1,N/,BLOCKS/N/,
                  PAGESIZE/320/,INVENTORY/90/

        WHERE "N" = (NUMBER OF SOURCE LINES TO BE STORED ON THE FILE) /
                    (50 SOURCE LINES PER PAGE)

$       PROGRAM   QUTI

$       PRMFL     RC,W,R,CAT/FILE-STRING

$       DATA      I *

IDS     INITIAL   1,N
```

EXHIBIT IDS100-3

CREATING THE COPY IDS FILE

```
$       PROGRAM   QCPY

$       LIMIT     ,21K

$       PRMFL     RC,W,R,CAT/FILE-STRING

$       PRMFL     PG,R,S,CAT/BCD-INPUT-DATA

$       SYSOUT    AA    (REQUIRED)

$       DATA      I*,COPY

        DIRECTIVES AND ALTERS

$       END COPY
```

EXHIBIT IDS100-4

DIRECTIVES

```
$       CREATE    ENTRY-NAME-1,FC

$       ALTER     M,N

$       UPDATE    ENTRY-NAME-1 (,FC)

$       RETRIEVE  ENTRY-NAME-1 (,FC)

$       DELETE    ENTRY-NAME-1

$       REPORT

$       REORG
```

EXHIBIT IDS100-5

*IDSCOPY

```
$       IDENT
    .
    .
    .
$       PRMFL     RC,R,R,CAT/FILE-STRING-1

$       PRMFL     RD,R,R,CAT/FILE-STRING-2
             .
             .
             .
        IDENTIFICATION DIVISION.
                .
                .
                .          .
        PROCEDURE DIVISION.
    .                 .
    .                 .
    .                 .
       *IDS COPY  ENTRY-NAME-1
             .
             .
             .
       *IDSCOPY   ENTRY-NAME-2.
```

EXHIBIT IDS100-6

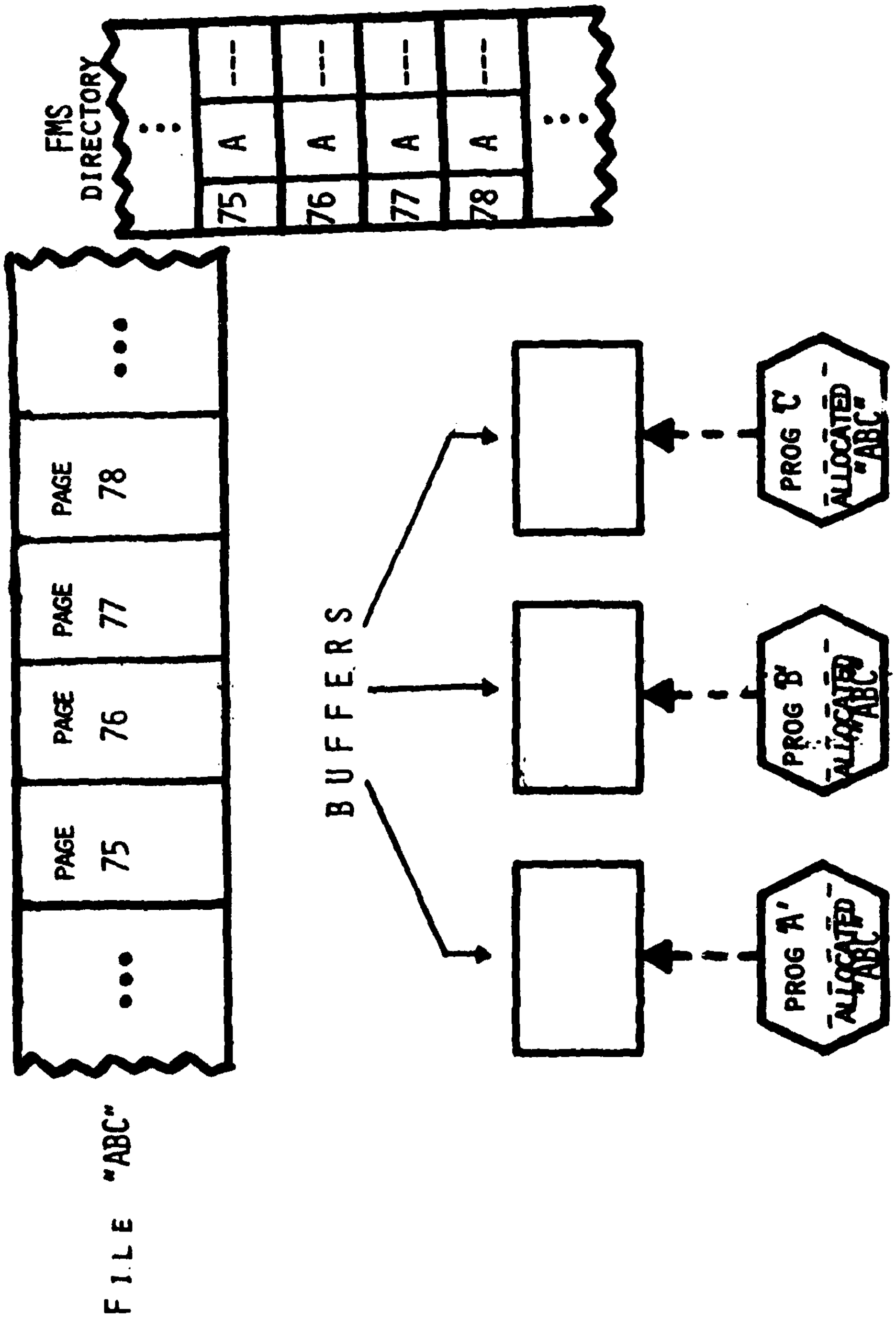
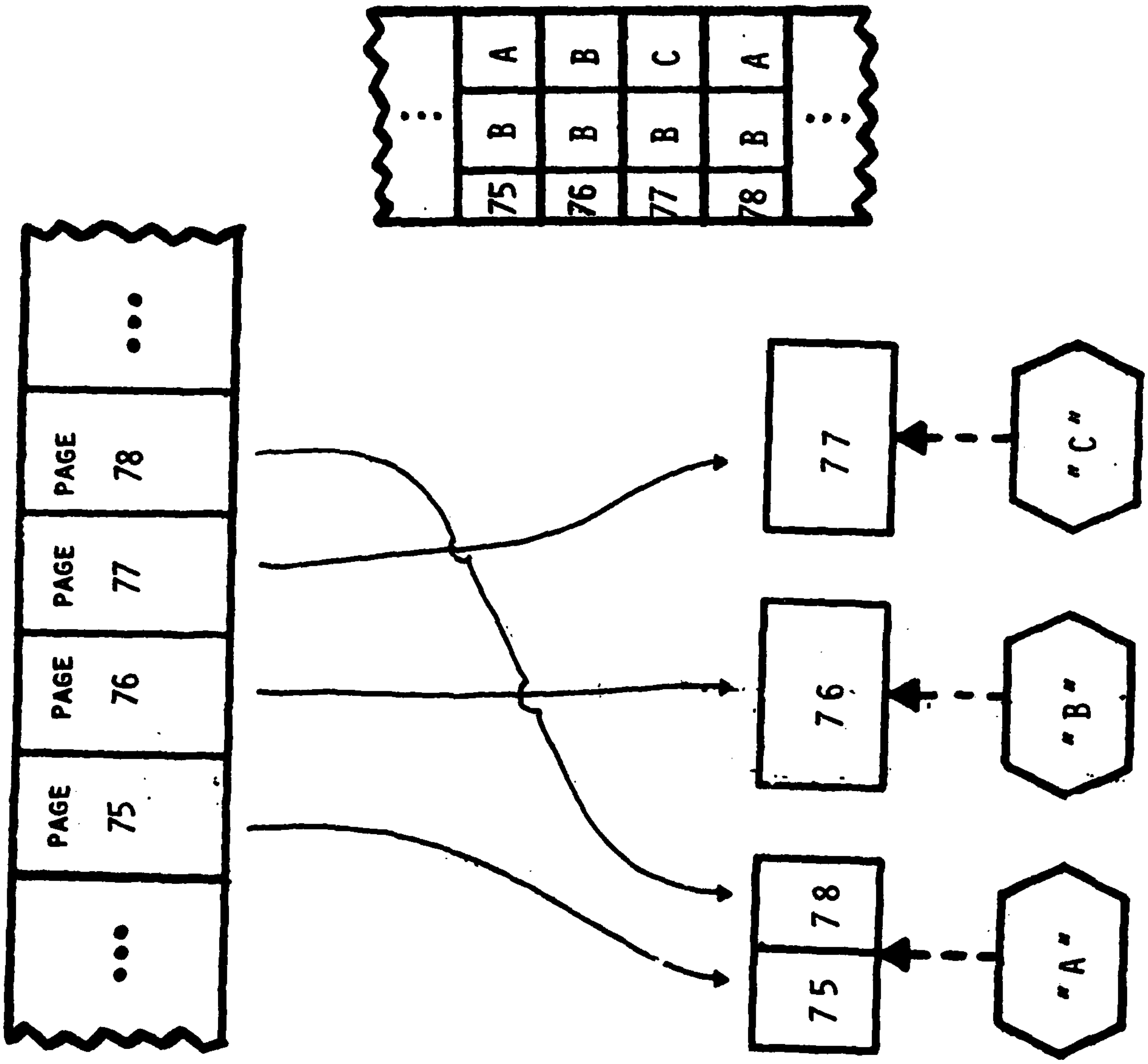


EXHIBIT IDS100-7



	...	
75	B	A
76	B	B
77	B	C
78	B	A
	...	

EXHIBIT IDS100-8

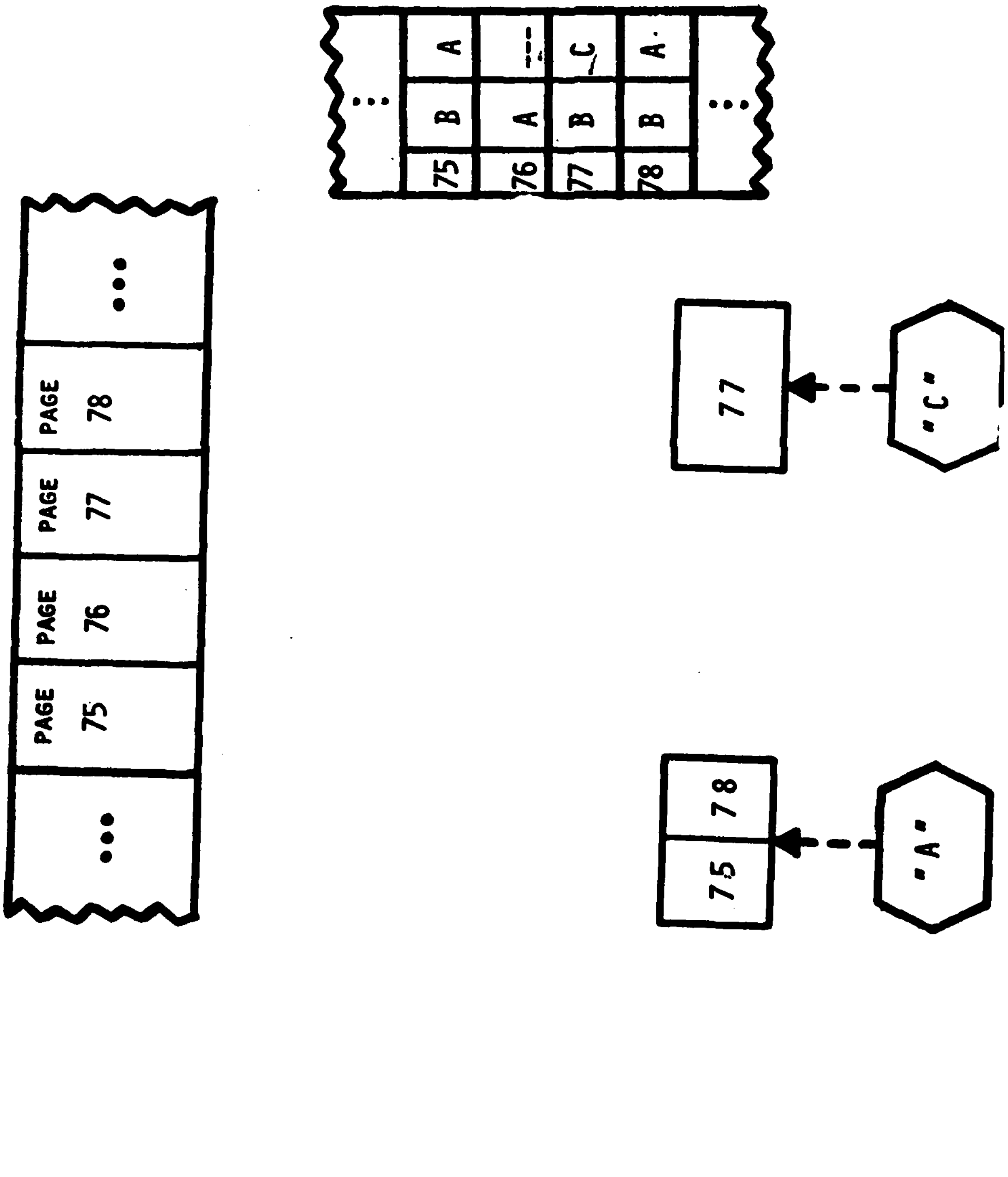
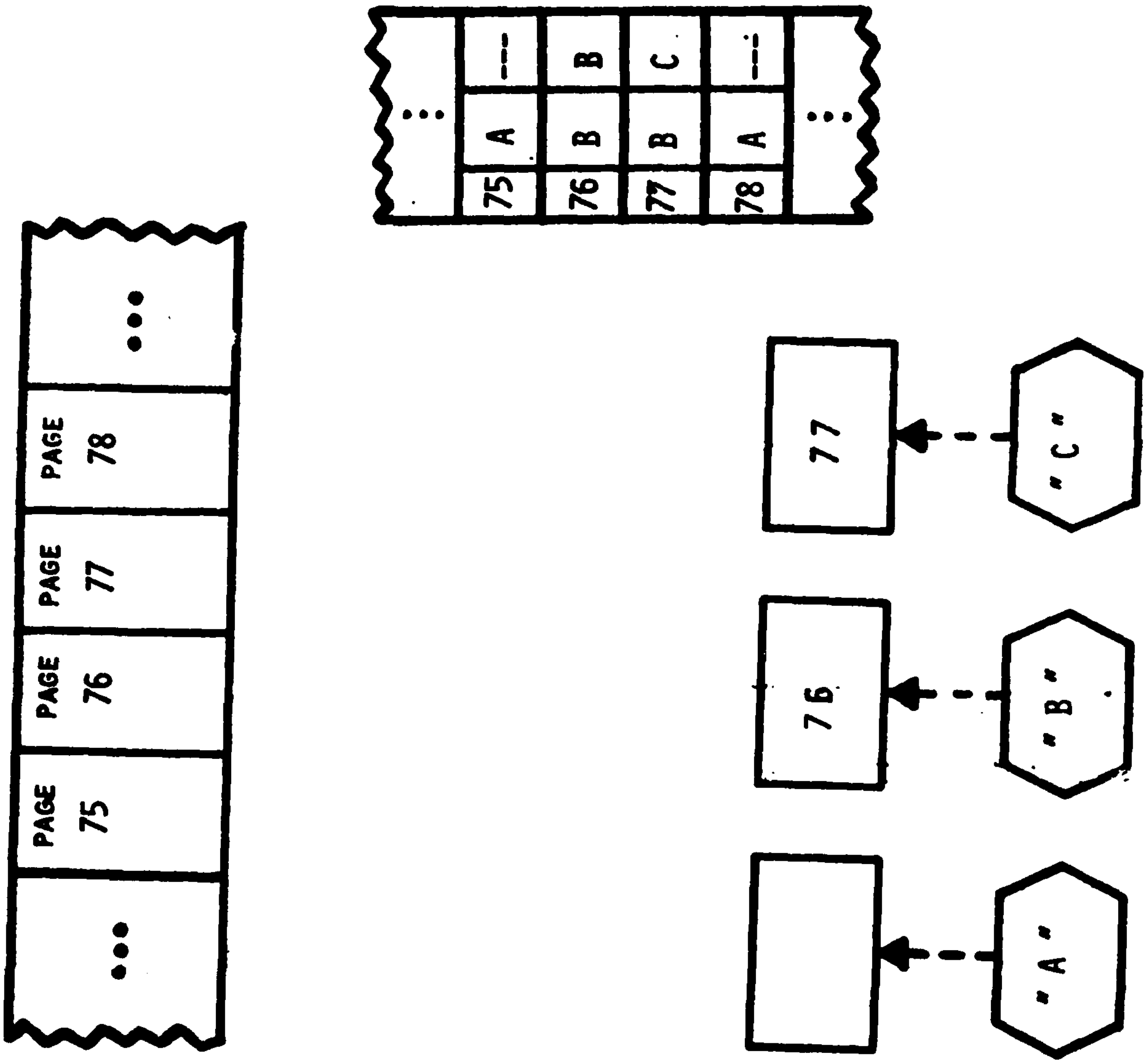


EXHIBIT IDS100-9



...		
75	A	---
76	B	B
77	B	C
78	A	---
...		

EXHIBIT IDS100-10

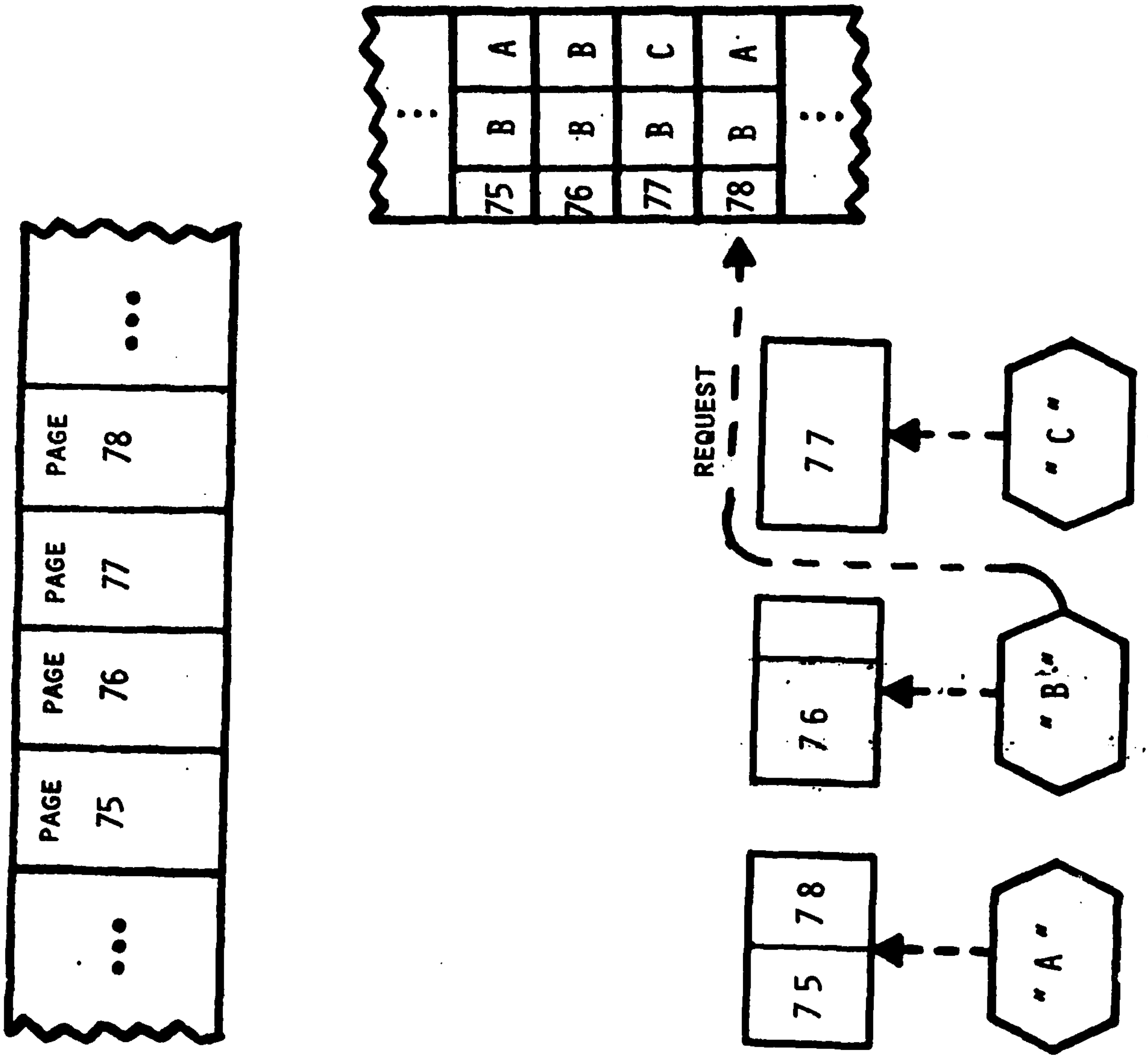


EXHIBIT IDS100-11

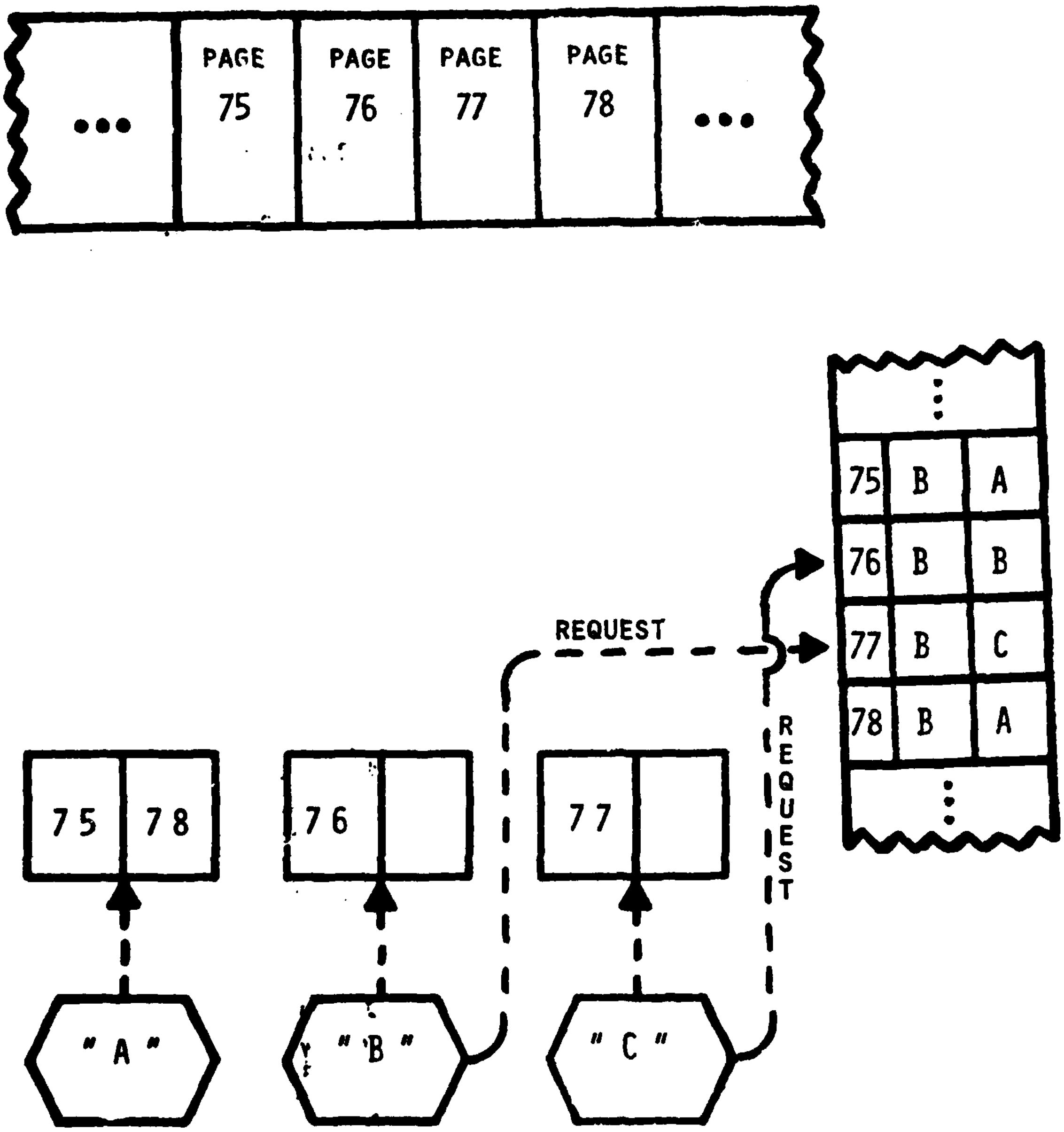


EXHIBIT IDS100-12

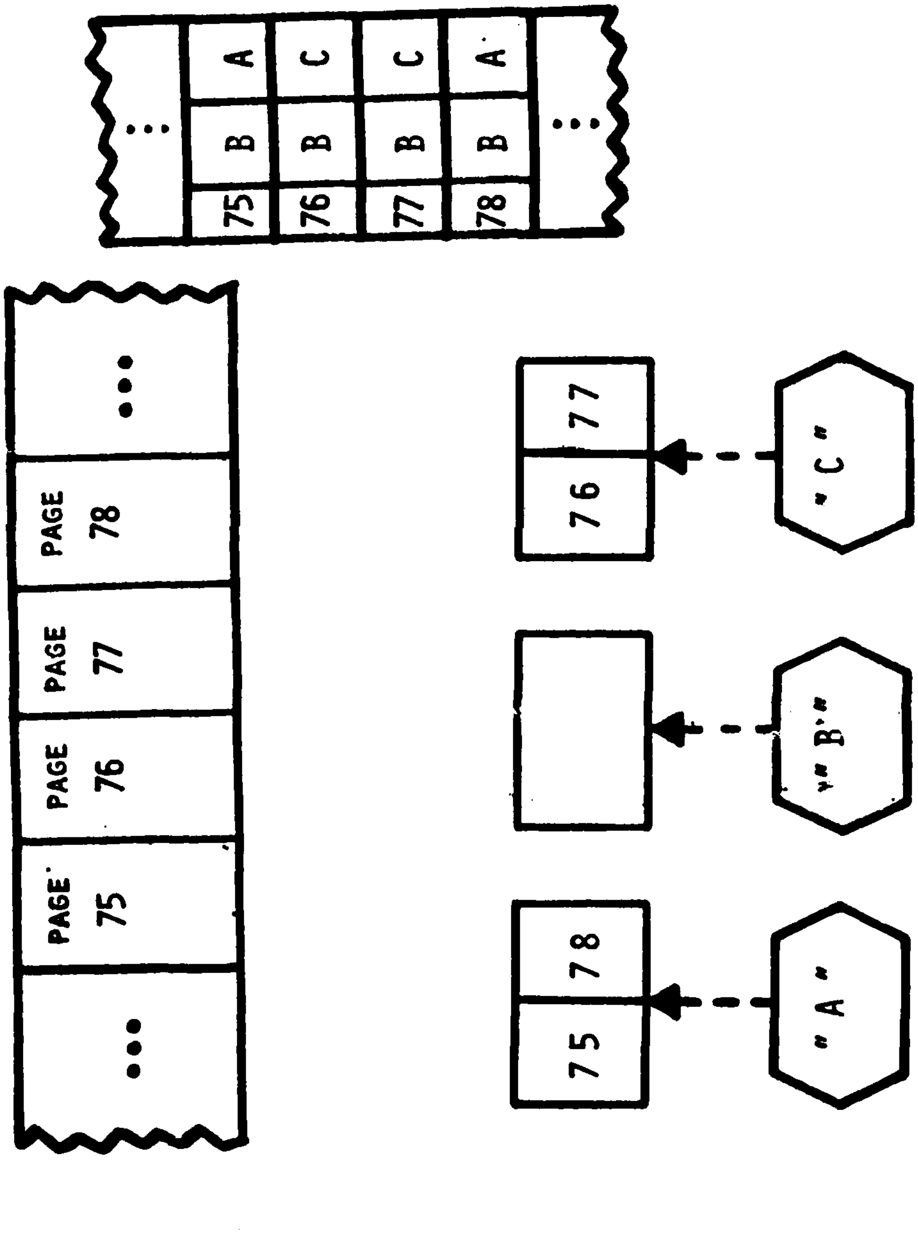


EXHIBIT IDS100-13

PROBLEM

The project for Lesson IDS110 will consist of the coding of the PROCEDURE DIVISION of a program which will process transactions against the DUE-OUT chain in the "ORDER-CONTROL" data base.

The program we will code will read transactions, each containing a part number and the quantity of that part which has been completed. The corresponding PART record is then retrieved and the DUE-OUT chain is walked. Each QTY-DUE or QTY-ORDERED record is processed until either the input quantity or the DUE-OUT chain is exhausted.

Each QTY-DUE or QTY-ORDERED record processed will be either removed (if completely satisfied) or updated (if partially satisfied). If a customer order is partially filled, a corresponding FILL record must be generated.

Output is printed to reflect the disposition of each part (i.e., to customer or shop). At the end of processing, the PART record is updated to indicate current parts on hand.

Before starting lesson IDS110, you should again read and study the contents of the PROCEDURE DIVISION in section 3 of your IDS Programmer's Guide. Then attempt to code the PROCEDURES DIVISION of this transaction problem (when directed by the lesson) using the flowchart in exhibit IDS110-2. Exhibit IDS110-3 states the input/output formats. The complete coding of the IDENTIFICATION, ENVIRONMENT and DATA DIVISIONS are made available to you in exhibit IDS110-4. You may also find helpful the IDS structure diagram in exhibit IDS043-7.

To make it easier for you to code this program, assume that your input file is clean and that you will not have to anticipate any processing errors. Thus, you will not have to validate your input or verify the storing and retrieving of records on the data base. This limitation will permit you to concentrate on basic and common programming.

When you have completed as much of the code as you can, sign back on to lesson IDS110. In this lesson, as in the previous two, you will code the PROCEDURE DIVISION; and if you fail to properly code a statement, you will be given an explanation of how that statement should be coded and why at that point in the program.

GOOD LUCK!!!

EXHIBIT IDS110-1

FLOWCHART

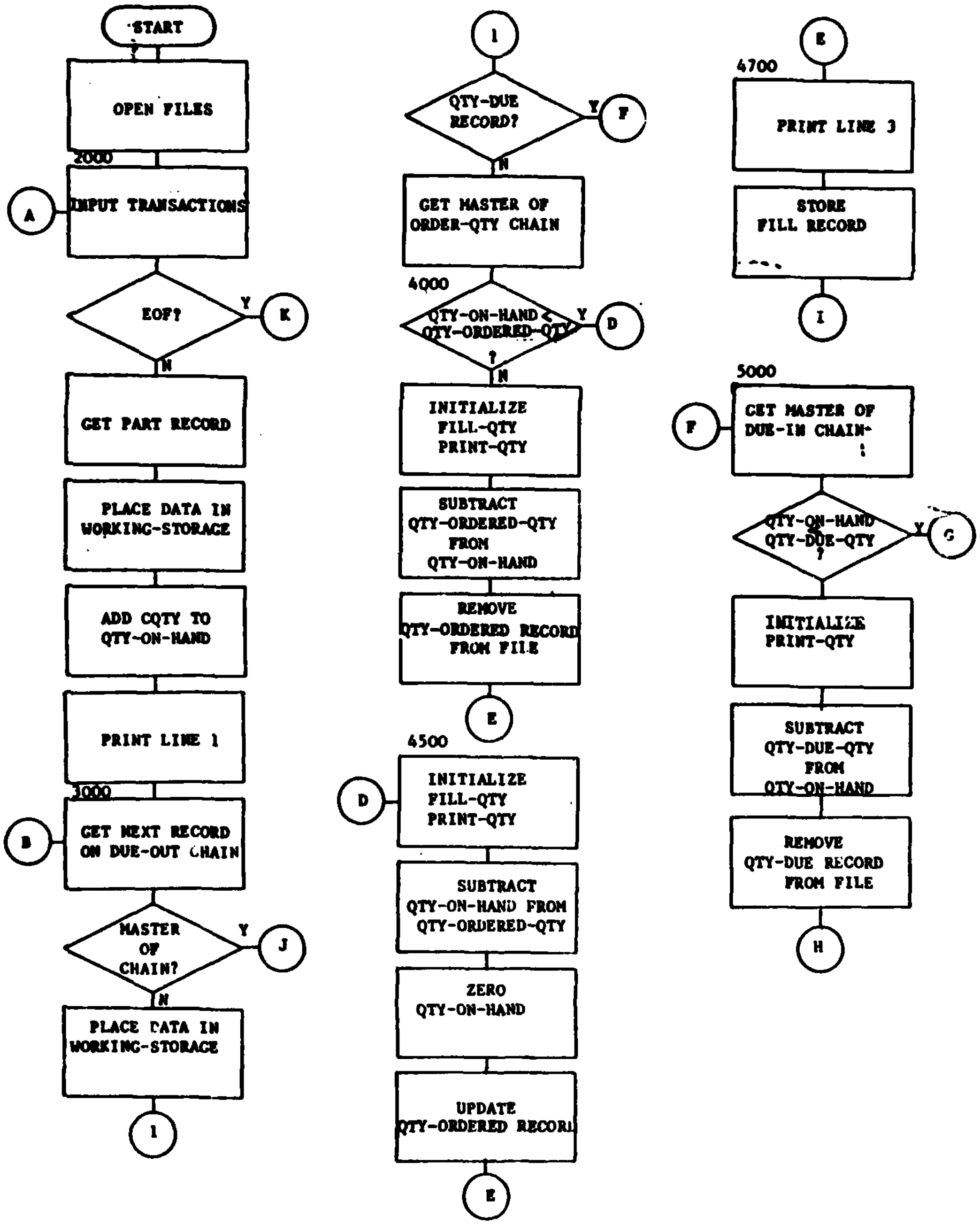

START
OPEN FILES
2000
A
INPUT TRANSACTIONS
EOF?
Y
K
N
GET PART RECORD
PLACE DATA IN WORKING-STORAGE
ADD CQTY TO QTY-ON-HAND
PRINT LINE 1
3000
B
GET NEXT RECORD ON DUE-OUT CHAIN
MASTER OF CHAIN?
Y
J
N
PLACE DATA IN WORKING-STORAGE
1
1
QTY-DUE RECORD?
Y
F
N
GET MASTER OF ORDER-QTY CHAIN
4000
QTY-ON-HAND < QTY-ORDERED-QTY ?
Y
D
N
INITIALIZE FILL-QTY PRINT-QTY
SUBTRACT QTY-ORDERED-QTY FROM QTY-ON-HAND
REMOVE QTY-ORDERED RECORD FROM FILE
E
4500
D
INITIALIZE FILL-QTY PRINT-QTY
SUBTRACT QTY-ON-HAND FROM QTY-ORDERED-QTY
ZERO QTY-ON-HAND
UPDATE QTY-ORDERED RECORD
E
E
4700
PRINT LINE 3
STORE FILL RECORD
I
5000
F
GET MASTER OF DUE-IN CHAIN
QTY-ON-HAND QTY-DUE-QTY ?
Y
G
INITIALIZE PRINT-QTY
SUBTRACT QTY-DUE-QTY FROM QTY-ON-HAND
REMOVE QTY-DUE RECORD FROM FILE
H

EXHIBIT IDS110-2

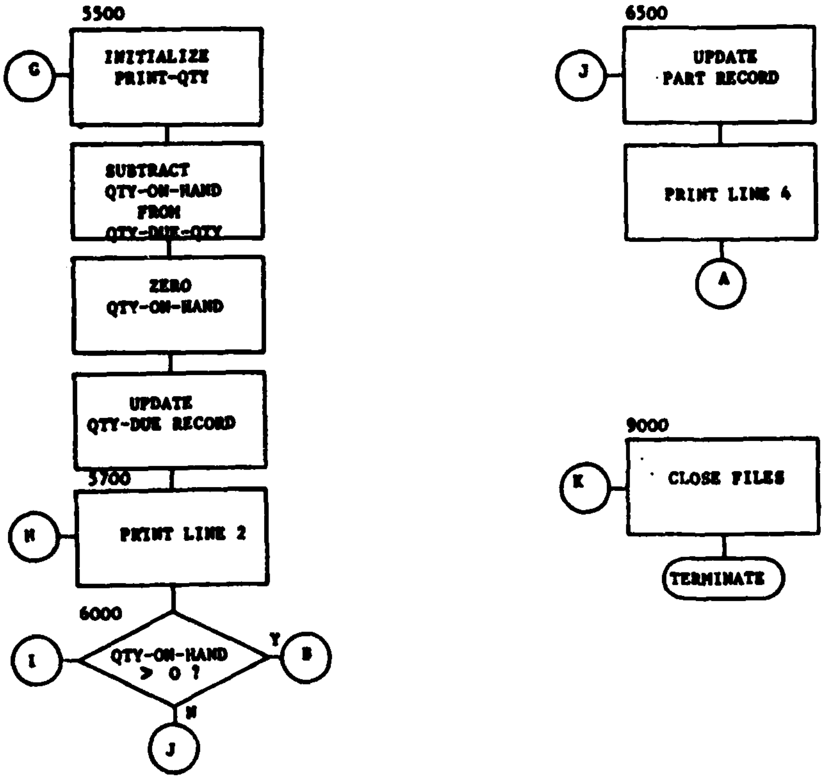


EXHIBIT IDS110-2 (Cont)

TRANSACTION INPUT FORMAT

CONTENT	CHARACTER POSITION	NUMBER OF CHARACTERS
Part number	1	6
Quantity	7	8

TRANSACTION OUTPUT FORMAT

```
                                                                          LINE

PART      ....................     QUANTITY       ................         1
...........................     UNITS TO SHOP      .................       2
.......................     UNITS TO SHIPPING ORDER     .................  3
.......................     UNITS ON-HAND STOCK     .................      4
```

NOTE 1: Line 1 will be produced for each input transaction.

NOTE 2: Line 2 will be produced for each QTY-DUE record modified or deleted.

NOTE 3: Line 3 will be produced for each QTY-ORDERED record modified or deleted.

NOTE 4: Line 4 will be produced after processing each transaction.

EXHIBIT IDS110-3

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  IDS110.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL NAMES.
    GETIME IS THE-DATE.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT IDS ORDER-CONTROL ASSIGN TO A1.
    SELECT CARDIN ASSIGN TO IN.
    SELECT PRINT-FILE ASSIGN TO OT FOR LISTING.
I-O-CONTROL.
    APPLY STANDARD ON CARDIN PRINT-FILE.
DATA DIVISION.
FILE SECTION.
FD CARDIN LABEL RECORDS ARE STANDARD
    DATA RECORD IS TRANSACTION.
01  TRANSACTION.
    02  CPART-NO PIC X(6).
    02  CQTY PIC 9(8).
    02  FILLER PIC X(66).
FD  PRINT-FILE LABEL RECORDS ARE STANDARD
01  OUT-LINE PIC X(132).
WORKING-STORAGE SECTION.
77  PRINT-QTY PIC 9(8)
77  ALT-PART-NO PIC X(6).
01  CURRENT-DATE.
    02  MON PIC 99.
    02  DAY PIC 99.
    02  YEAR PIC 99.
    02  FILLER PIC X(6).
01  REP-HED PIC X(34) JUST VALUE
        "DISPOSITION OF COMPLETED PARTS".
01  LINE-PART.
    02  FILLER PIC X(9) JUST VALUE "PART ".
    02  L-PART-NO PIC X(7).
    02  FILLER PIC X(9) VALUE "QUANTITY".
    02  L-QTY-ON-HAND PIC 9(8).
01  LINE-SHOP.
    02  FILLER PIC X(7) VALUE SPACES.
    02  L-PRINT-QTY PIC 9(8).
    02  FILLER PIC X(15) VALUE " UNITS TO SHOP".
    02  L-SHOP-ID PIC X(10).
01  LINE-SHIP.
    02  FILLER PIC X(7) VALUE SPACES.
    02  LS-PRINT-QTY PIC 9(8).
    02  FILLER PIC X(28) VALUE
           " UNITS TO SHIPPING ON ORDER".
    02  L-ORDER-NO PIC 9(8).
01  LINE-STOCK.
    02  FILLER PIC X(7) VALUE SPACES.
    02  LS-QTY-ON-HAND PIC 9(8).
    02  FILLER PIC X(20) VALUE
        "UNITS ON-HAND STOCK".
```

```
IDS SECTION.
MD ORDER-CONTROL.
01  DUMMY
    TYPE 001
    RETRIEVAL DUMMY-REF FIELD.
    02  DUMMY-REF PIC 9(8).
    98  CUSTOMER-LIST MASTER
        CHAIN-ORDER SORTED.
    98  SHOP-LIST MASTER
        CHAIN-ORDER AFTER.
01  CUSTOMER
    TYPE 002
    RETRIEVAL CALC CHAIN
    PAGE-RANGE 1 25.
    02  CUST-ID PIC X(20).
    02  CUST-INFO SIZE 120.
        03  FULL-NAME PIC X(30).
        03  BILL-ADDR PIC X(40).
        03  SHIP-ADDR PIC X(40).
        03  PHONE PIC X(10).
    98  CUSTOMER-LIST DETAIL
        SELECT CURRENT
        DUPLICATES NOT
        ASCENDING CUST-ID.
    98  CALC DETAIL
        RANDOMIZE CUST-ID.
    98  CUSTOMER-ORDERS MASTER
        CHAIN-ORDER AFTER.
```

EXHIBIT IDS110-4

```
01  SHOP
    TYPE 003
    RETRIEVAL SHOP-LIST CHAIN
    PAGE-RANGE 51 150.
    02  SHOP-ID PIC X (10).
    98  SHOP-LIST DETAIL
        SELECT CURRENT.
    98  DUE-IN MASTER
        CHAIN-ORDER AFTER.
    98  PARTS-CONTROL MASTER
        CHAIN-ORDER AFTER.
01  ORDER
    TYPE 004
    RETRIEVAL CALC CHAIN
    PAGE-RANGE 26 50.
    02  ORDER-NO PIC 9(8).
    02  DATE-RCVD PIC X(6).
    98  CUSTOMER-ORDERS DETAIL
        SELECT CURRENT.
    98  CALC DETAIL
        RANDOMIZE ORDER-NO.
    98  INVOICE MASTER
        CHAIN-ORDER LAST.
    98  ORDER-QTY MASTER
        CHAIN-ORDER AFTER.
01  PART
    TYPE 005
    RETRIEVAL CALC CHAIN
    PAGE-RANGE 51 150.
    02  PART-NO PIC X(6).
    02  QTY-ON-HAND PIC 9(8).
    98  PARTS-CONTROL DETAIL
        SELECT CURRENT.
    98  CALC DETAIL
        RANDOMIZE PART-NO.
    98  DUE-OUT MASTER
        CHAIN-ORDER LAST.
    98  PARTS-LIST MASTER
        CHAIN-ORDER FIRST.
    98  COMPONENT MASTER
        CHAIN-ORDER FIRST.
01  QTY-DUE
    TYPE 006
    RETRIEVAL DUE-OUT CHAIN
    PAGE-RANGE 51 150
    02  QTY-DUE-QTY PIC 9(8).
    98  DUE-IN DETAIL
        SELECT CURRENT.
    98  DUE-OUT DETAIL
        SELECT CURRENT.
    01  QTY-REQD
    TYPE 007
    RETRIEVAL PARTS-LIST CHAIN
    PAGE-RANGE 51 150.
    02  QTY-REQD QTY PIC 9(8).
    98  PARTS-LIST DETAIL
        SELECT UNIQUE
        MATCH-KEY PART-NO.
    98  COMPONENT DETAIL
        SELECT UNIQUE
        MATCH-KEY ALT-PART-NO SYN PART-NO
        LINKED MASTER.
01  FILL
    TYPE 008
    RETRIEVAL INVOICE CHAIN
    PAGE-RANGE PIC X(6).
    02  FILL-PART-NO PIC X (6).
    02  FILL-QTY PIC 9(8).
    02  DATE-SHIP SIZE 6.
        03 YEAR PIC 99.
        03 MON PIC 99.
        03 DAY PIC 99.
    98  INVOICE DETAIL
        SELECT CURRENT.
01  QTY-ORDERED
    TYPE 009
    RETRIEVAL DUE-OUT CHAIN
    PAGE-RANGE 51 150.
    02  QTY-ORDERED-QTY PIC 9(8).
    98  ORDER-QTY DETAIL
        SELECT CURRENT.
    98  DUE-OUT DETAIL
        SELECT CURRENT.
```

EXHIBIT IDS110-4 (Cont)

IDS ABBREVIATED JOURNALIZATION

```
                             NBEFOR

                             NAFTER
IDS       JBREVE                              / FILE-CODE  ,...  /
                             FBEFOR

                             LAFTER                                  ...

+ = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = +  =

$         USE         ,.QbREV,.QBIT1/1/,.QBITS/N/,.QBIT2/1/

      WHERE    "N"  =  ( NF * 2 ) + ( NB * 65 )

               NF   =  NUMBER OF SUBFILES WITH "JBREVE"

               NB   =  NUMBER OF BUFFERS FOR JOURNAL PAGES


+ = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = +  =


$            FILE         1.,X1S,"N"R

      WHERE

                                            N = NF  WITH LAFTER &/OR FBEFOR SPECIFIED
            "N" = TOTAL # LINKS =  64 *               (TP_N /2196
                                            N = 1


            TP  = NUMBER OF PAGES IN THE SUBFILE
```

EXHIBIT IDS110-5

FILE RESTORE

1. FILSYS RESTORE OR QUTL LOAD

2. "ROLL" THE DATABASE "FORWARD" WITH

   A. QUTL (JTAPE)

   B. RERUNNING ALL TRANSACTIONS

UTILITY RESTORE

1. JX TAPES INTO QUTL (JTAPE)

2. JX TAPES INTO QUTP, INTO QUTS, INTO QUTL (STAPE)

EXHIBIT IDS110-6

COEXISTENCE

```
+ = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + = + =

$              FILSYS
FC/IDS/     CAT/FILE-STRING,BASESIZE/1000/,...,COEXISTENCE/YES/
```

EXHIBIT IDS110-7

COMPUTER OPERATION (H6000-CDT) COURSE EXHIBITS

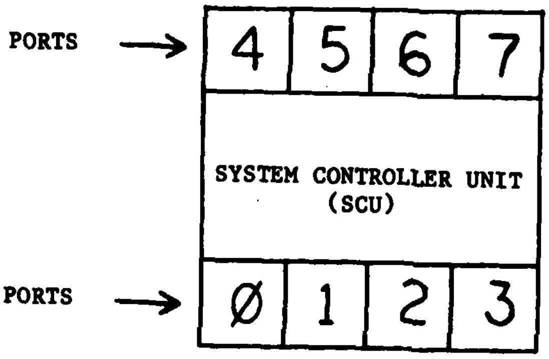


EXHIBIT OPR010-1

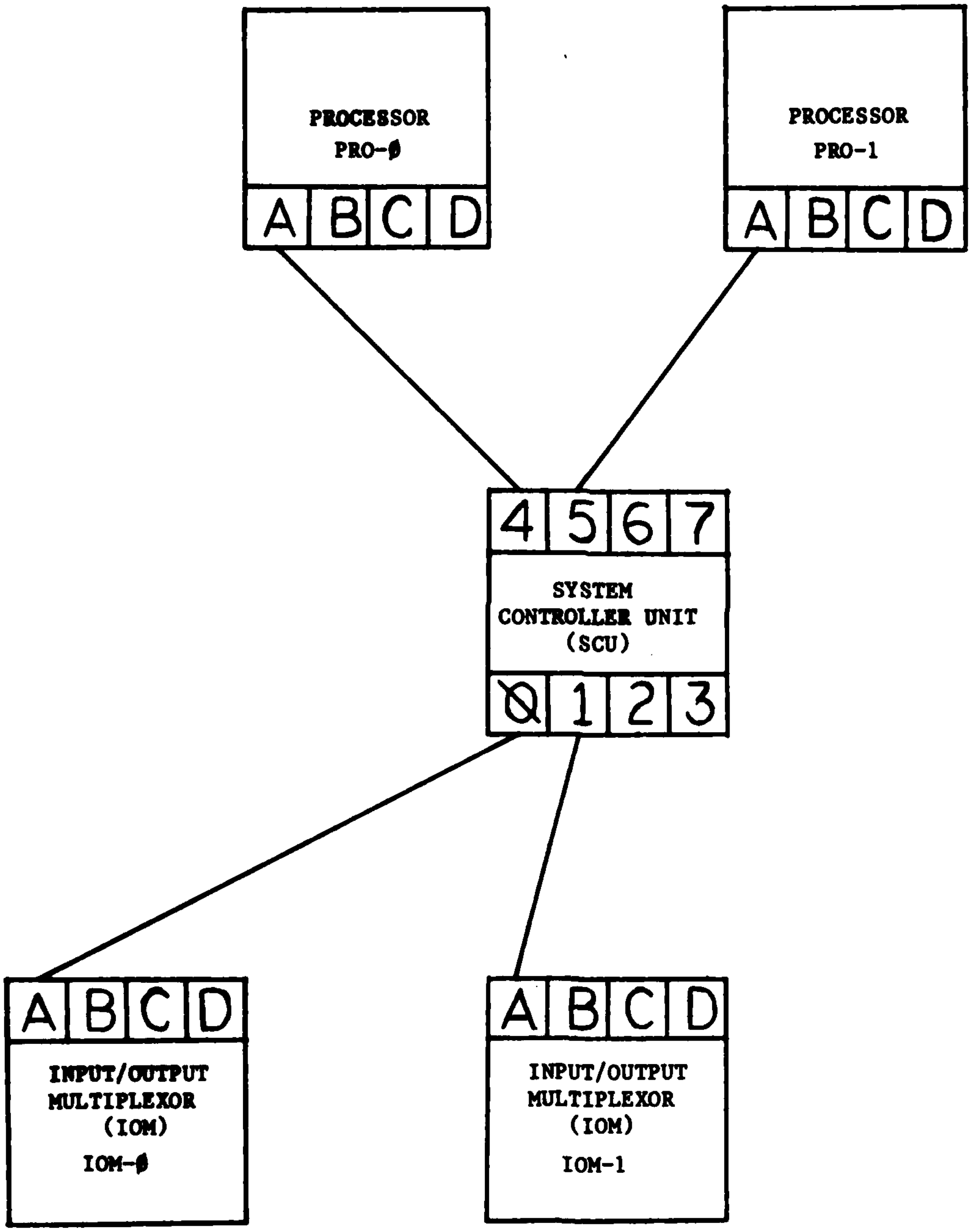


EXHIBIT OPR010-2

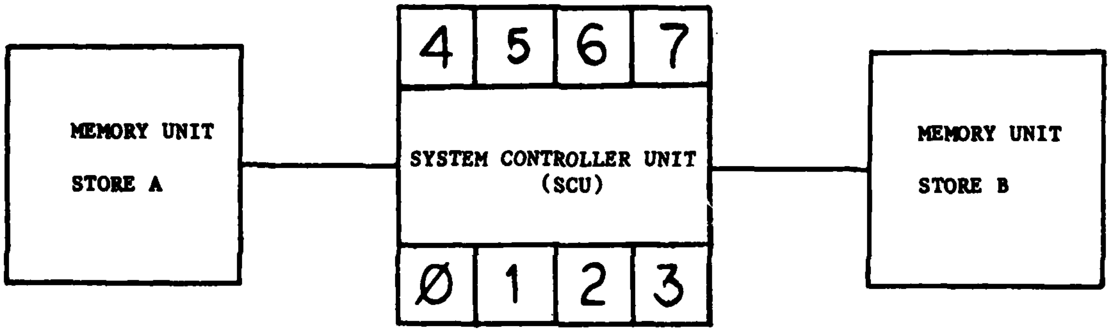


EXHIBIT OPR010-3

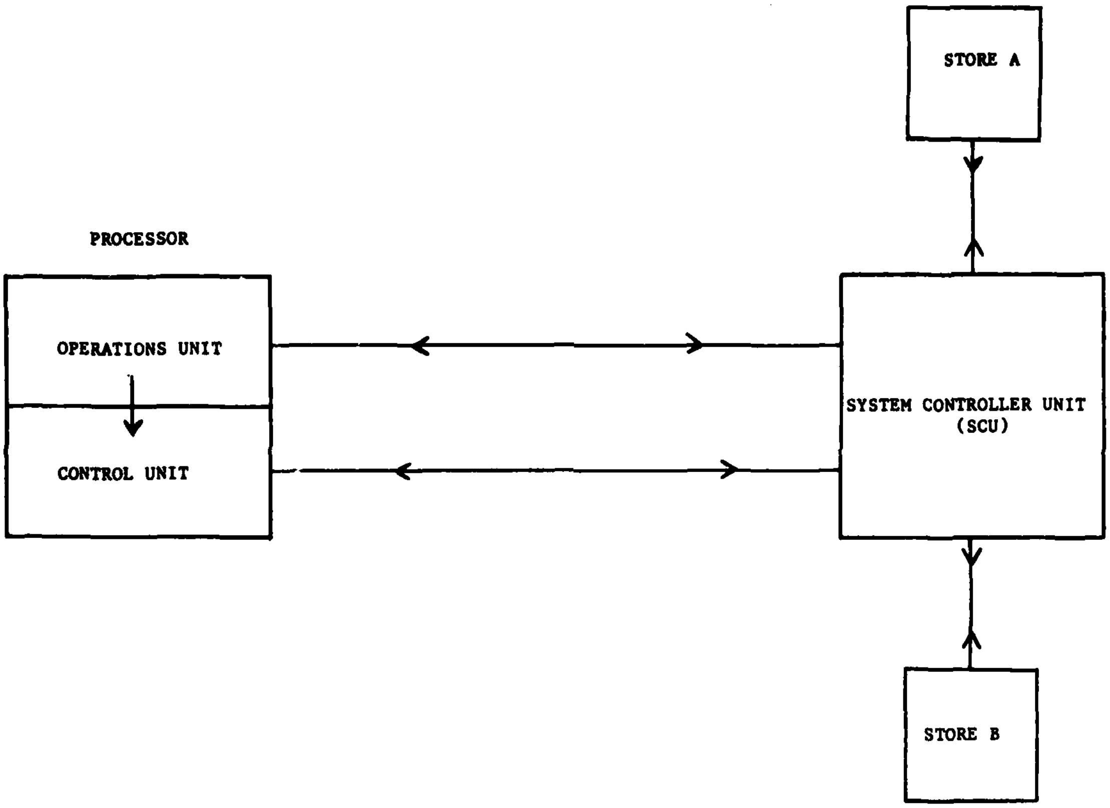


EXHIBIT OPR010-4

BAR CONCEPT

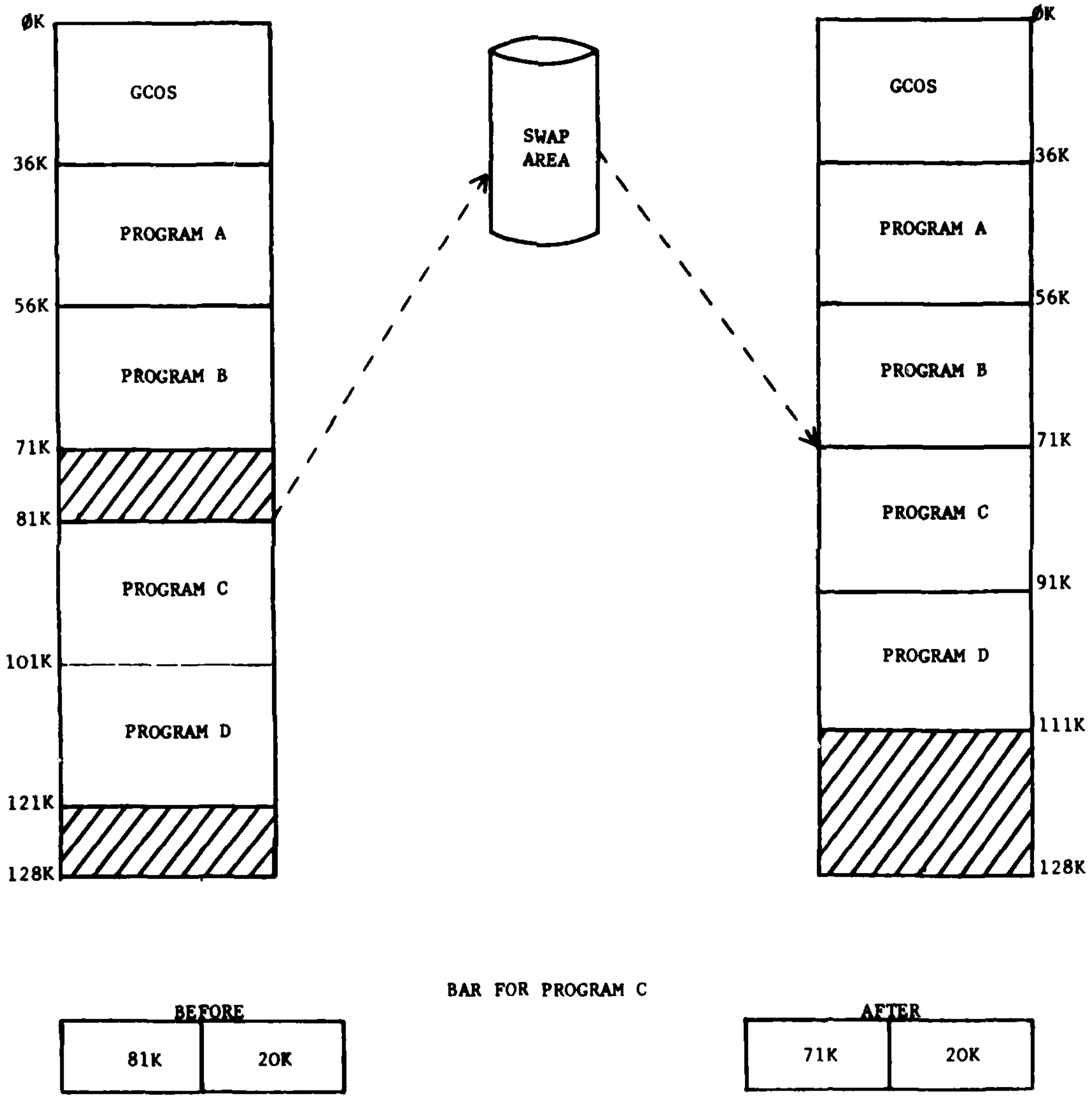


EXHIBIT OPR010-5

SINGLE PROCESSOR CONNECTED TO FOUR SYSTEM CONTROLLER UNITS

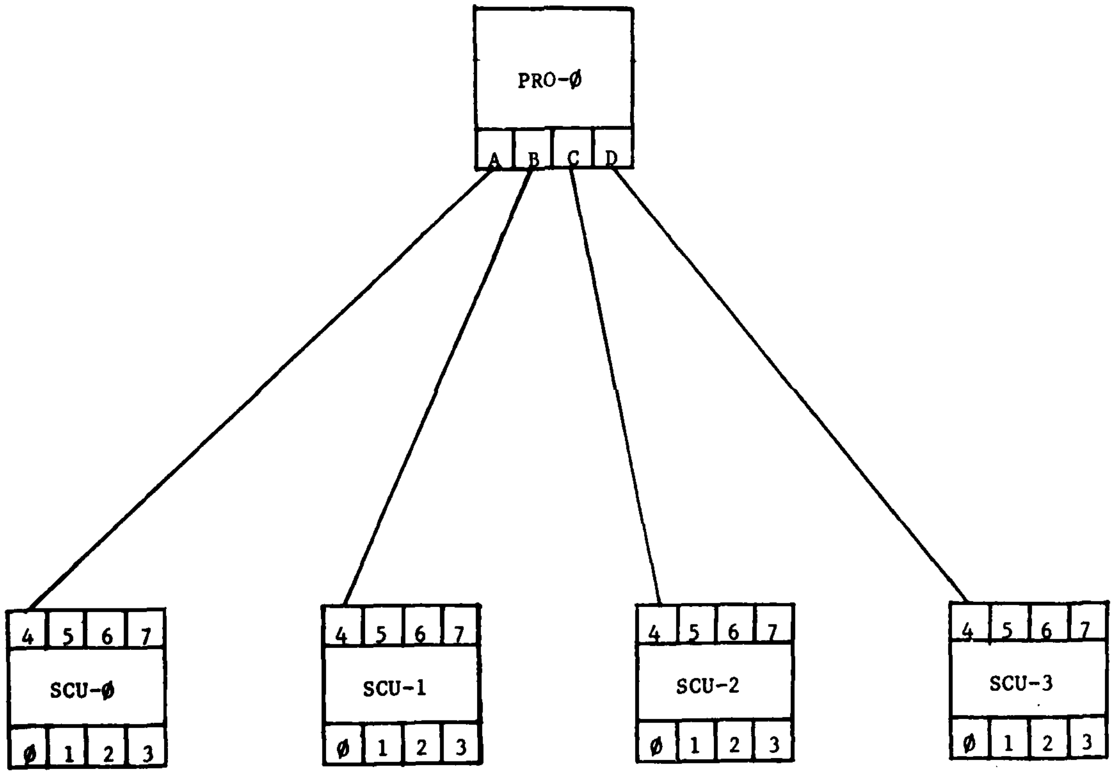


EXHIBIT OPR010-6

FOUR PROCESSORS CONNECTED TO TWO SYSTEM CONTROLLER UNITS

(CONTROL PROCESSOR)

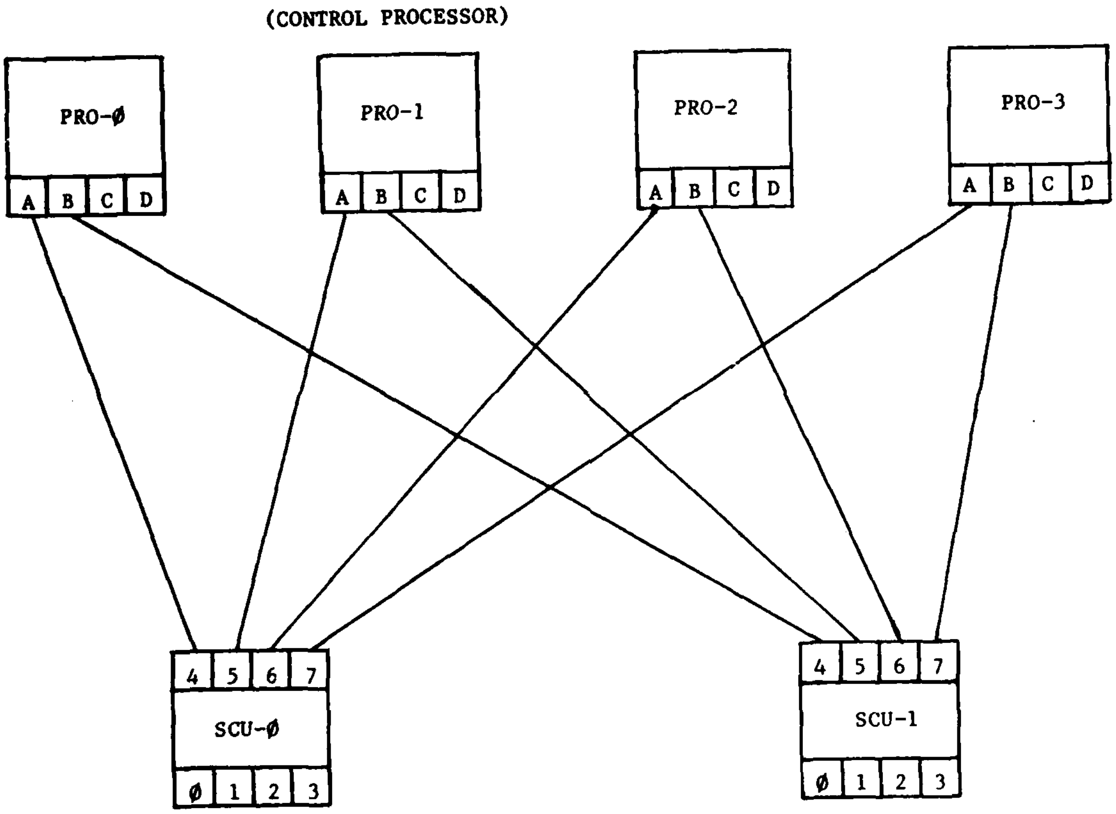


EXHIBIT OPR010-7

SINGLE INPUT/OUTPUT MULTIPLEXOR CONNECTED TO FOUR SYSTEM CONTROLLER UNITS

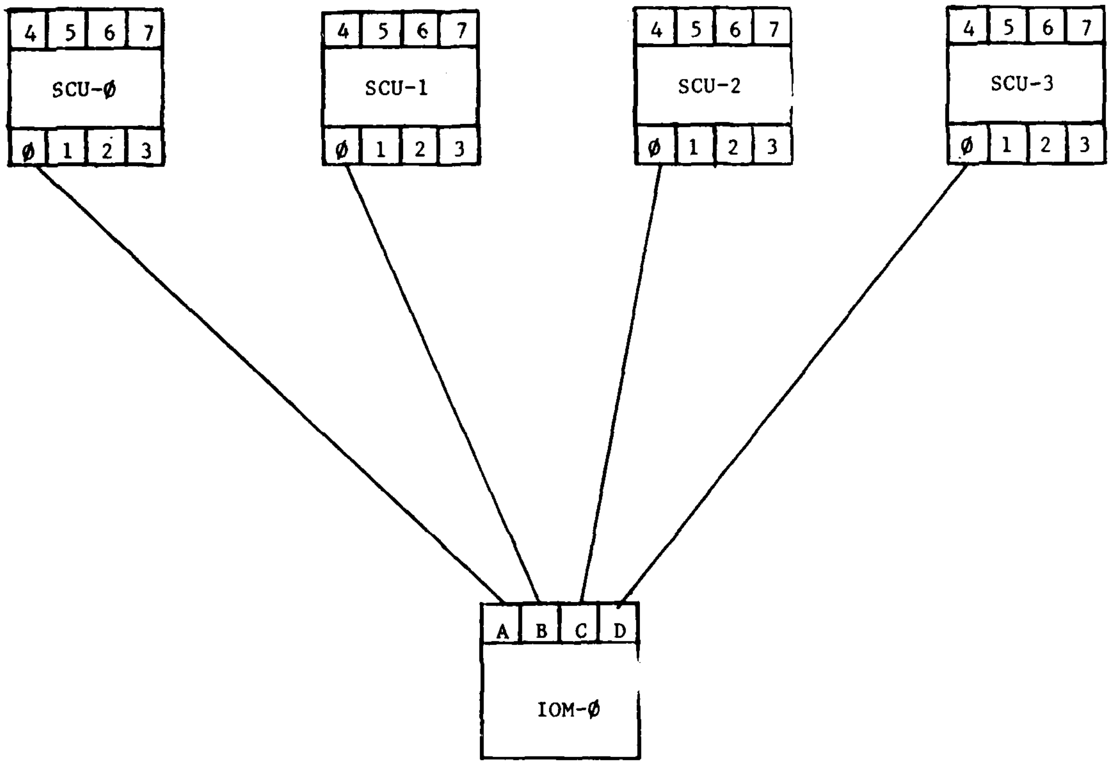


EXHIBIT OPR010-8

FOUR INPUT/OUTPUT MULTIPLEXORS CONNECTED TO THREE SYSTEM CONTROLLER UNITS

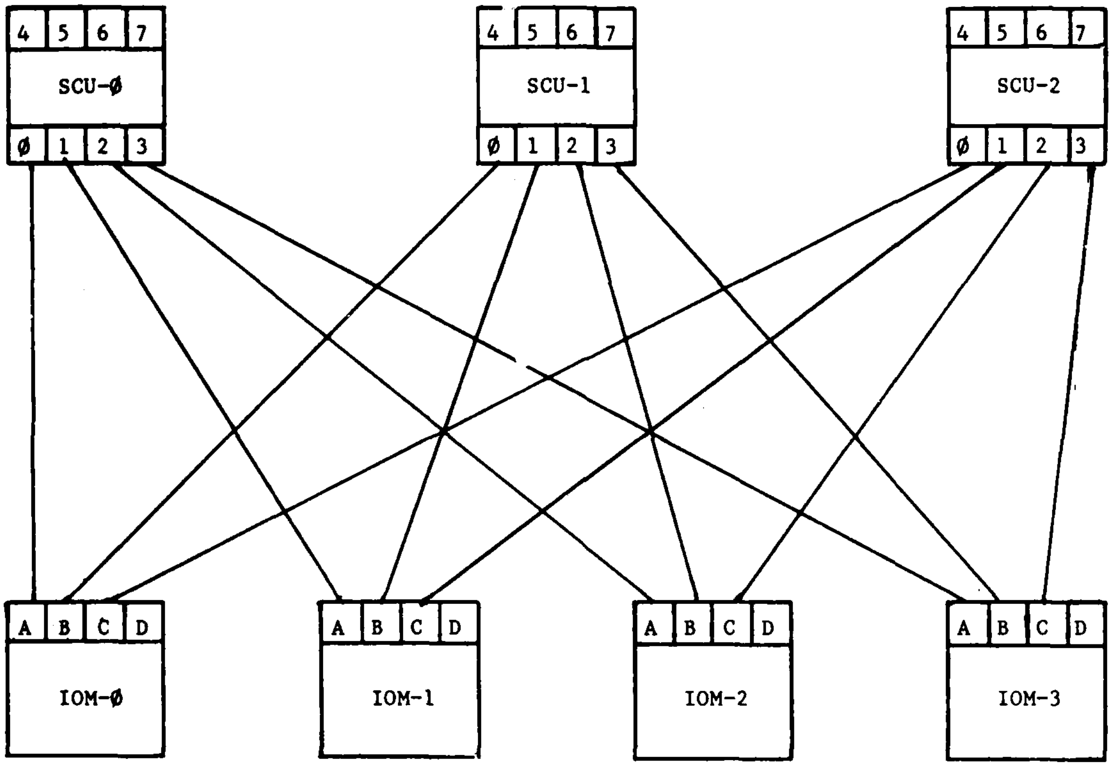


EXHIBIT OPR010-9

OPERATOR'S CONTROL PANEL

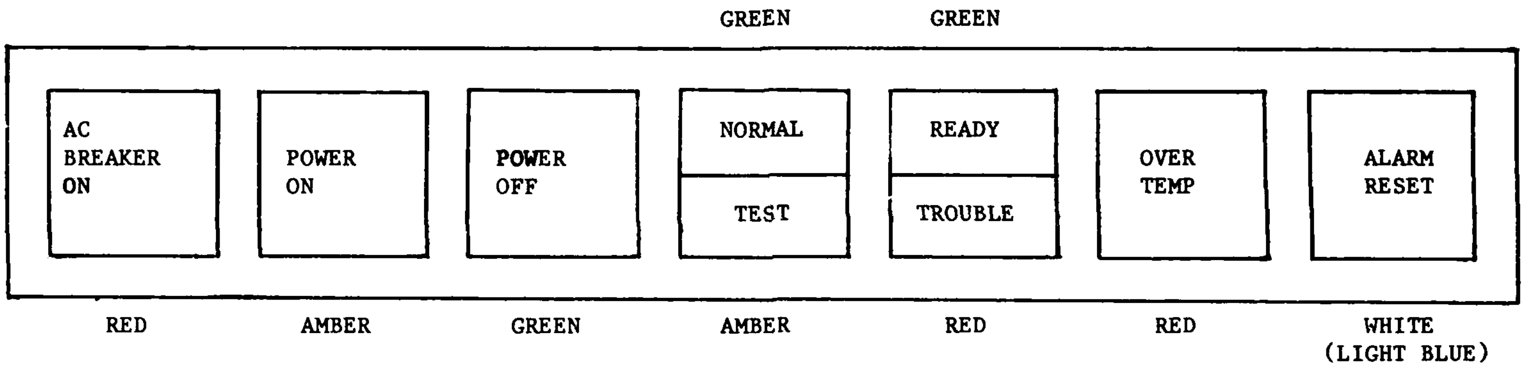


EXHIBIT OPR010-10

LESSON OPR010 OUTLINE

HARDWARE COMPONENTS

A. MEMORY MODULE

1. Purpose: Provides the required amount of directly addressable memory.

2. Functional Units:

a. System Controller Unit (SCU)

- serves as the hub for all information interchanged between the processor module and the Input/Output Multiplexor (IOM).
- serves as the overall coordinating control unit.
- provides communication between the high speed memory and the active modules.

Ports

- eight ports for connection to active modules.
- positional priority in the order of their number (0......7).
- other main frame modules are connected in the following priority:
  first - Input/Output Multiplexor (IOM)
  second - Processor

b. Memory Units

Store A incremented/decremented in multiples of 32K
Store B 1K is equal to 1024-words of memory

B. PROCESSOR MODULE

1. Purpose: Conducts all actual computational processing within the Series H6000 System.

2. Functional Units:

a. Operations Unit - contains the logic to execute arithmetic and logical operations.

b. Contrcl Unit - provides the interface between the Operations Unit and the System Controller Unit (SCU).

c. Extension Unit - provides the additional logic required to prepare and process instructions that belong to the special feature "Extended Instruction Set" (EIS).

3. Special Features: Under the control of the General Comprehensive Operating Supervisor (GCOS).

a. Dual Mode Operations - provides effective operating control of the multiprogramming environment to GCOS.

EXHIBIT OPR010-11

Modes of Operation

Master Mode
- reserved for GCOS.
- permits initiation of data input/output operations through the Input/Output Multiplexer (IOM).
- permits the setting of control registers.

Slave Mode
- used for the execution of all user programs.
- restricts memory references to assigned program boundaries.
- causes all memory references to be relative to a Base Address Register (BAR).

b. Base Address Register (BAR):

- contains the beginning address of the program in memory and the number of 1024-words (K) assigned to the program.
- performs both address translation and memory protection functions in Slave Mode.
- used for program swapping and memory compaction.

c. Timer Register:

- initiates a program interrupt at the end of a pre-established interval of time.
- used by GCOS to time programs for automatic termination.
- prevents programs from monopolizing the processor.
- provides detailed accounting information on processor and peripheral use time.

d. Processor Faults: Cause interruption of sequential instruction execution and transfer of control to one of sixteen fault vector locations for appropriate action by GCOS. The fault vectors provide:

- program control.
- system control.
- a call for operating system services.

e. Extended Instruction Set (EIS): Well suited for a heavy business workload.

4. <u>Ports</u>
- four input/output ports labeled A, B, C, and D.
- provides the Processor with direct access to the memory connected to the System Controller Unit (SCU).

C. INPUT/OUTPUT MULTIPLEXOR (IOM)

1. <u>Purpose</u>: Coordinates all input/output operations between memory and peripherals.

2. Ports

- four input/output ports labeled A, B, C, and D.
- provides the IOM with direct access to the memory connectea to the System Controller Unit (SCU).

D. FRONT-END NETWORK PROCESSOR (DATANET 355)

1. Purpose: Provides a communication link between the remote users and the system.

2. Connection: Within the WWMCCS community, the Datanet 355 is configured as a peripheral device connected to the input/Output Multiplexer (IOM) through the use of a Direct Interface Adapter (DIP).

E. OPERATOR'S CONTROL PANEL

1. Purpose: Indicates/controls the status and operation of the main frame module.

2. Indicators: Indicates the status of the main frame modules.

a. AC Breaker on (lit)
- main circuit breaker is turnea on.
- AC power is available.

b. Normal/Test

c. Ready/Trouble

Ready (lit) - normal operation. Trouble (lit) - malfunction in the module.

d. Over Temp (lit) - over-temperature condition in either the logic or power supply of the main frame module.

3. Pushbutton/Indicators: Indicates and controls the operation of the main frame module.

a. Power On

Pushbutton (pressed)
- DC power is applied to the module.
- "Power Off" pushbutton/indicator is turned "OFF".

Indicator (lit) - module is online.

b. Power Off

Pushbutton (pressed)
- DC power is removed from the module.
- "Power On" pushbutton/indicator is turned "OFF".

Indicator (lit) - module is off-line.

c. Alarm Reset

Pushbutton (pressed)
- alarm bell and indicator are turned "OFF".

Indicator (lit) - an over-temperature condition exists.
- a fault-on-fault condition exists.

OPR020 TEST INSTRUCTIONS

STEP 1. Indicate, on Exhibit OPR020-2, the necessary switch settings for the system configuration listed below.

STEP 2. Establish the maximum amount of core available to be accessible to the System Controller Unit (SCU).

STEP 3. Interleave available memory.

STEP 4. Designate the control processor.

STEP 5. Enable the ports that have main frame modules connected.

STEP 6. Group like active modules.

STEP 7. Insure that the SCU is online.

STEP 8. Set the "Alarm Disable" switch so that you will be notified when an over temperature condition occurs in the SCU.

SYSTEM CONFIGURATION

SCU-0 - 128K (Store A - 64K and Store B - 64 K)
IOM-0 - connected to Port Ø of SCU-0
IOM-1 - connected to Port 1 of SCU-0
PRO-0 - connected to Port 6 of SCU-0 (control processor)
PRO-1 - connected to Port 7 of SCU-0

EXHIBIT OPR020-1

System Controller Configuration Panel

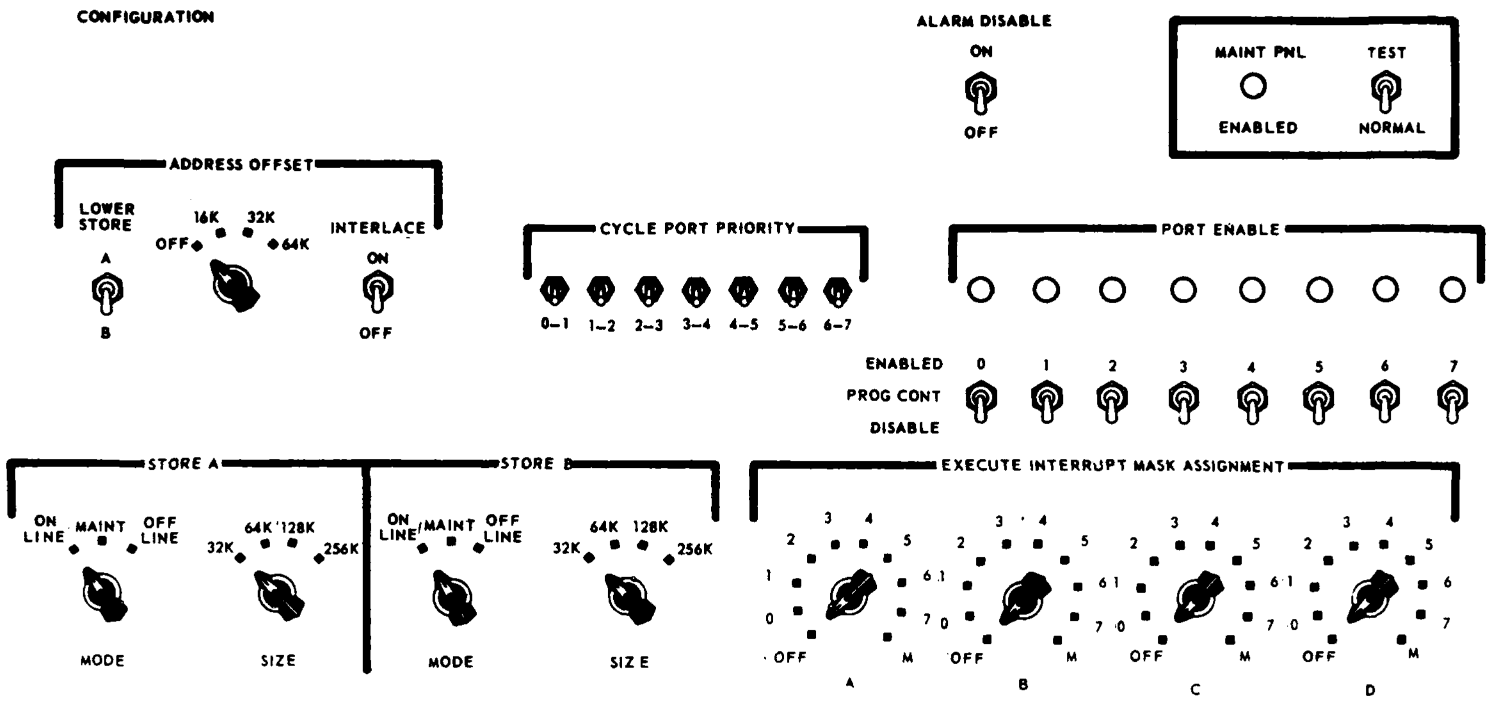
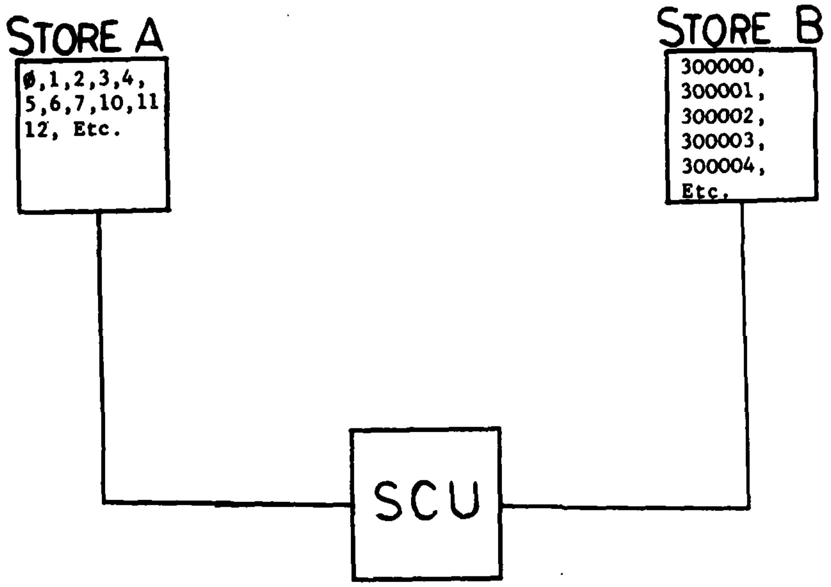
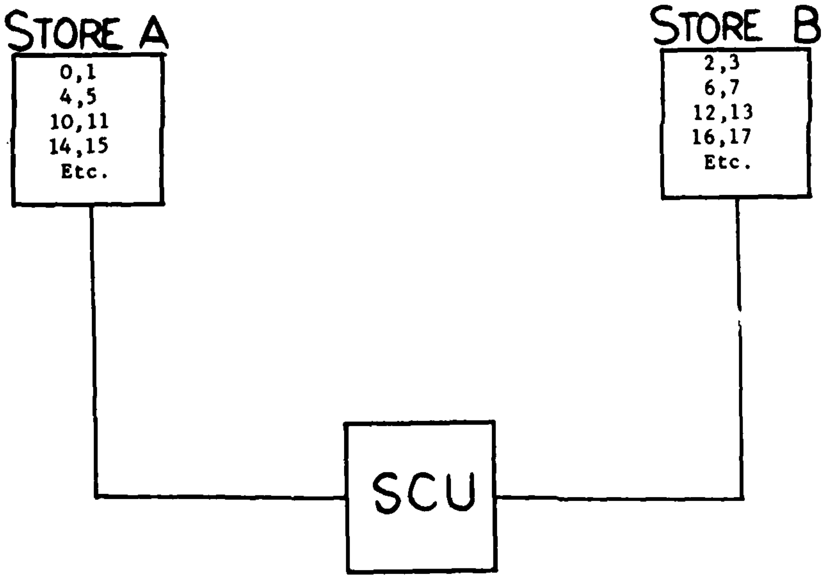


EXHIBIT OPR020-2



NOTE: "Interlace" switch set to the "OFF" position.



NOTE: "Interlace" switch set to the "ON" position.

EXHIBIT OPR020-3

LESSON OPR02O OUTLINE

SCU CONFIGURATION PANEL

A. The following five switches control the memory within the memory module:

1. Mode:

a. Purpose: Controls the accessibility by the System Controller Unit (SCU) of memory within each memory unit.

b. Functional Positions:

On Line
- normal operating position.
- allows the memory to be accessed by the SCU's ports.

Maint
- memory cannot be accessed by the SCU's ports.
- memory can be accessed by the SCU's maintenance panel test logic.

Off Line
- memory is not accessible to the SCU.
- memory is effectively removed from the system.

2. Size:

a. Purpose: Establishes the amount of memory that is accessible to the SCU.

b. Functional Positions:

Position	Memory Size
32K	32K
64K	64K
128K	96K, 128K
256K	160K, 192K, 224K, 256K

3. Lower Store:

a. Purpose: Selects the memory unit to which the lower addresses are to be routed.

b. Functional Positions: A and B.

NOTE: The "Size" switch to which the "Lower Store" switch is set establishes the address boundary between the two memory units (Store A and Store B).

4. Address Offset:

a. Purpose: Used in conjunction with other configuration switches to assign a faulty block of memory to a logically non-existent upper address range.

EXHIBIT OPR020-4

b. Functional Positions: OFF, 16K, 32K, and 64K.

5. Interlace:

a. Purpose: Enables interleaving between the two memory units.

b. Functional Positions:

On - interleaving is enabled.
- the "Lower Store" switch is disabled.

Off - interleaving is disabled.
- the "Lower Store" switch is enabled.

B. The following three groups of switches control the SCU's ports:

1. Execute Interrupt Mask Assignment:

a. Purpose: Used to designate the port of the SCU to which the control processor is connected.

b. Functional Switches: A, B, C, and D.

c. Functional Positions:
OFF - Unassigned.
Ø-7 - Assigned to port.
M - Assigned to maintenance panel.

NOTE: Only the rotary switch labeled "A" is used to designate the control processor. The remaining switches must be set to the "OFF" position.

2. Port Enable:

a. Purpose: Used to enable/disable specific ports of the SCU.

b. Functional Switches: Eight toggle switches that coincide with the eight ports.

c. Functional Positions:

Enable - port is enabled and able to communicate with an active main frame module.

Prog Cont - port is under program control and will be turned on and off by the proper bit in a program. (Not normally used in the WWMCCS community.)

Disable - port is disabled and unable to communicate with an active main frame module.

3. Cycle Port Priority:

a. Purpose: Used to group like active modules to assure equal access to memory connected to the same SCU.

b. Functional Switches: Seven toggle switches labeled: 0-1, 1-2, 2-3, 3-4, 4-5, 5-6, and 6-7.

c. Functional Positions:
On - two or more like modules are grouped.
Off - modules are not grouped.

C. The following two switches pertain to hardware problems:

1. Alarm Disable:

a. Purpose: used to enable/disable the alarm bell.

b. Functional Positions:
On - the alarm bell is disabled.
Off - the alarm bell is enabled.

2. Normal/Test:

a. Purpose: controls the operation of the maintenance panel test logic.

b. Functional Positions:
Normal - the maintenance panel test logic is disabled.
Test - the maintenance panel test logic is enabled.

OPR031 TEST INSTRUCTIONS

STEP 1. Obtain 2 copies of Exhibit OPR030-2.

STEP 2. Set all necessary switches on both IOM configuration panel exhibits to complement the system configuration described below.

STEP 3. Set the necessary switches to enable the system to load a startup deck from a desk of cards.

STEP 4. Return to a remote device and type "GET OPR 031 NEW".

SYSTEM MODULES:

1-SCU
2-IOMs

Physical cabling of the system modules:

IOM-0 - Port A cabled to Port 0 of SCU-0.
IOM-1 - Port B cabled to Port 1 of SCU-0.

The following peripheral devices are configured to the IOM's:

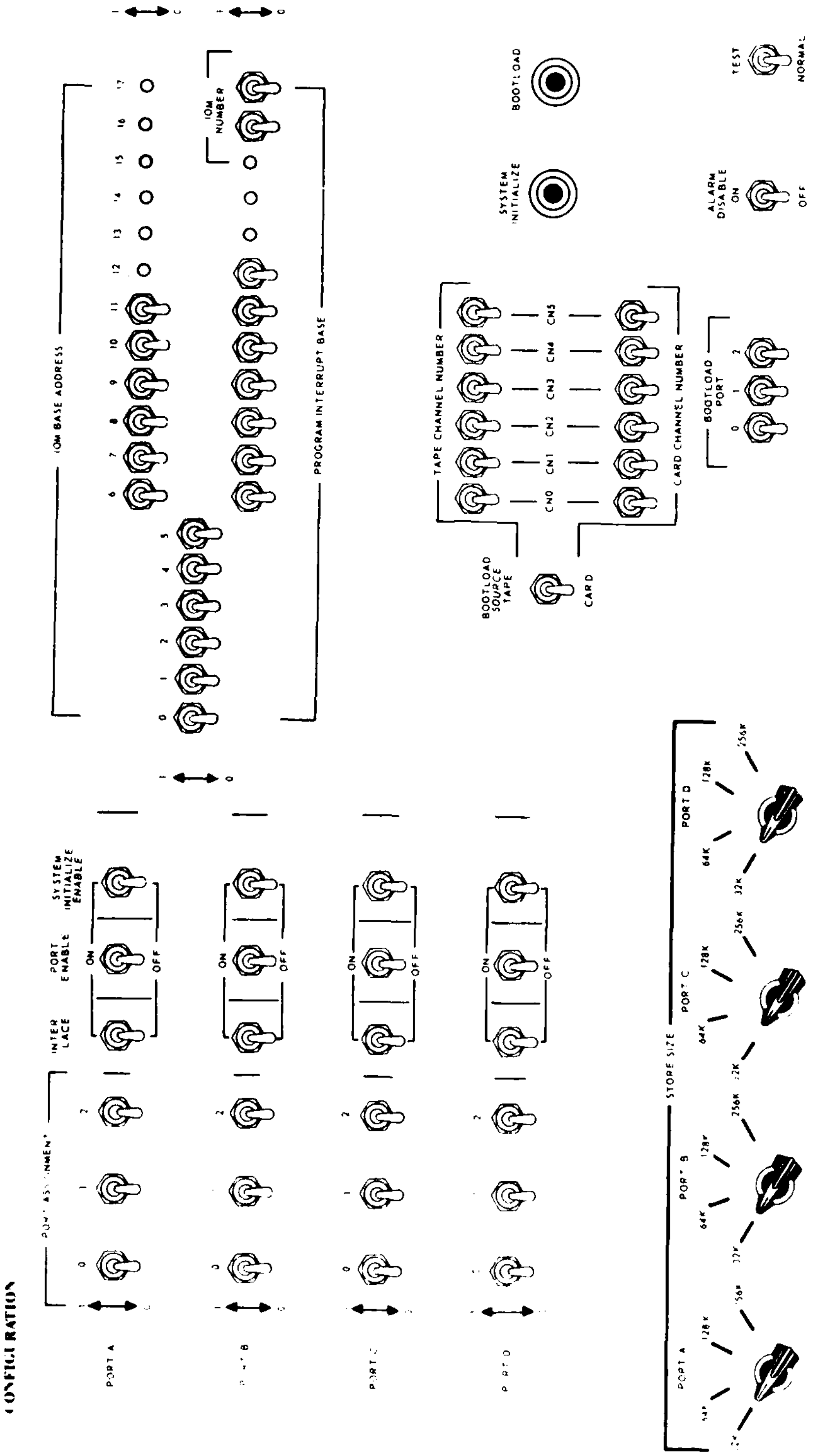
IOM-0:

1 Magnetic Tape Controller - Channel 12
1 Card Reader (CRZ201) - Channel 16

IOM-1:

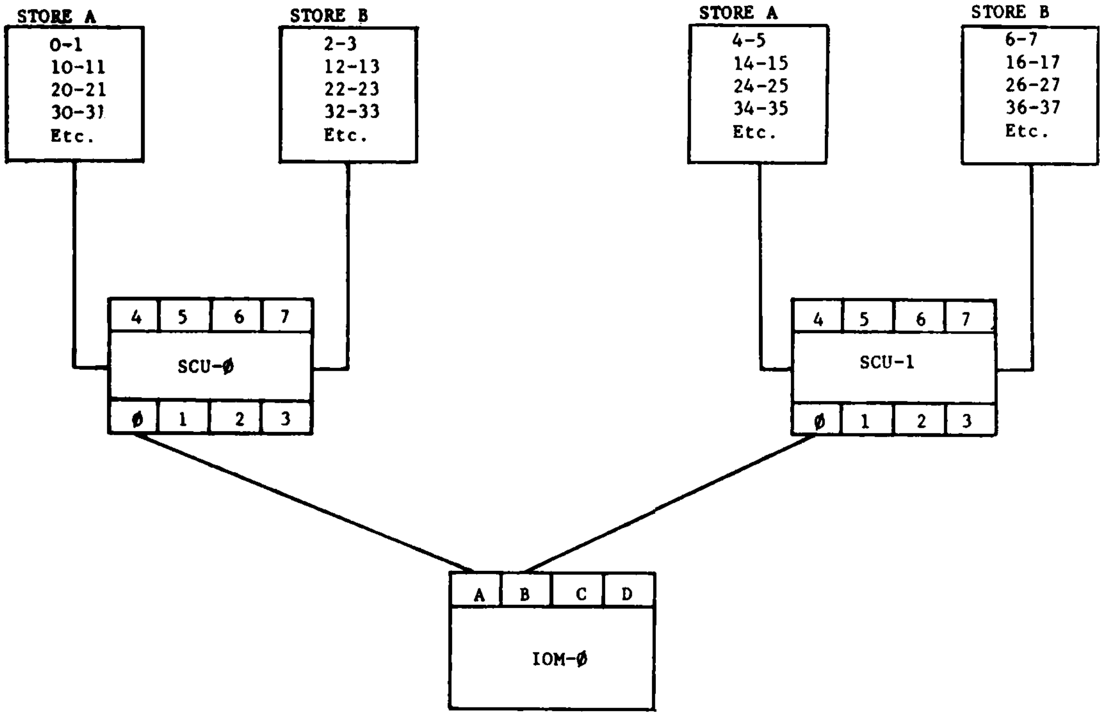
1 Magnetic Tape Controller - Channel 10
1 Card Reader (CRZ201) - Channel 12

EXHIBIT OPR030-1



IOM CONFIGURATION PANEL

EXHIBIT OPR030-2



SWITCH SETTINGS

SCU-0 Interlace Switch - ON
SCU-1 Interlace Switch - ON
IOM-0 Interlace Switches
Port A - ON
Port B - ON

EXHIBIT OPR030-3

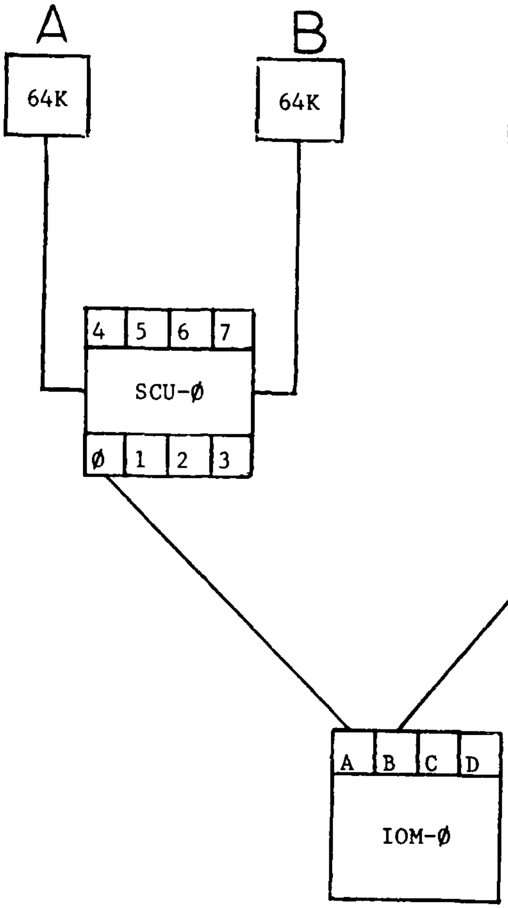


EXHIBIT OPR030-4

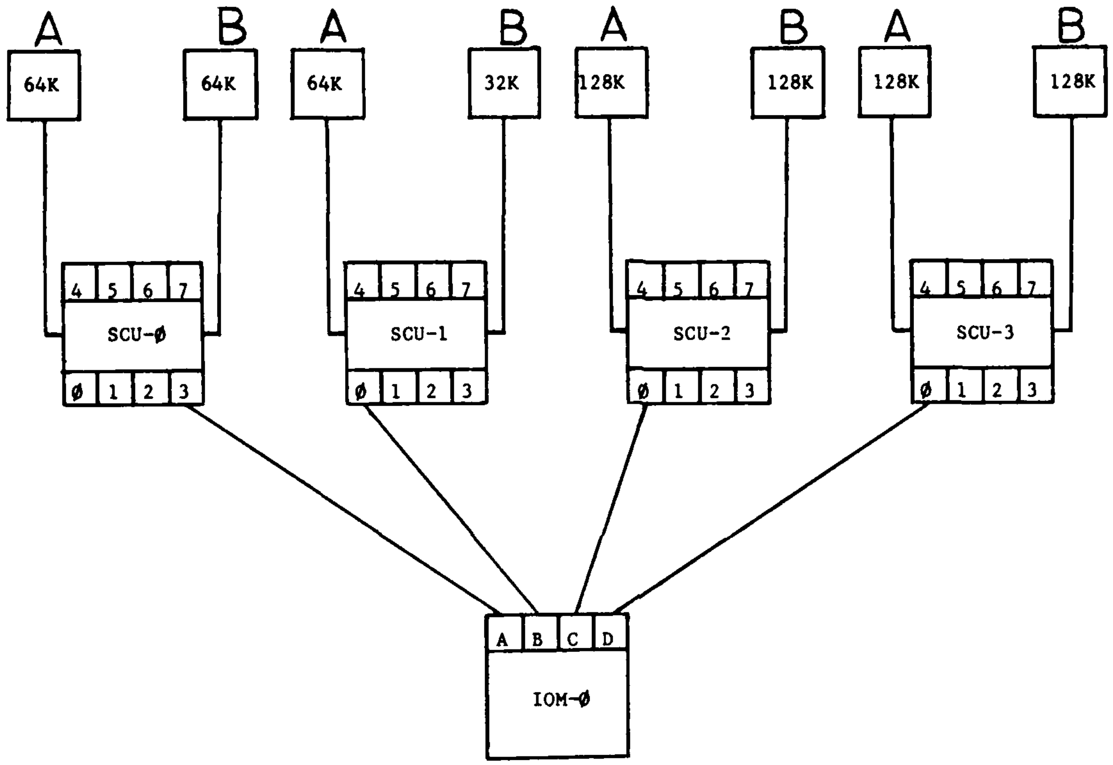


EXHIBIT OPR030-5

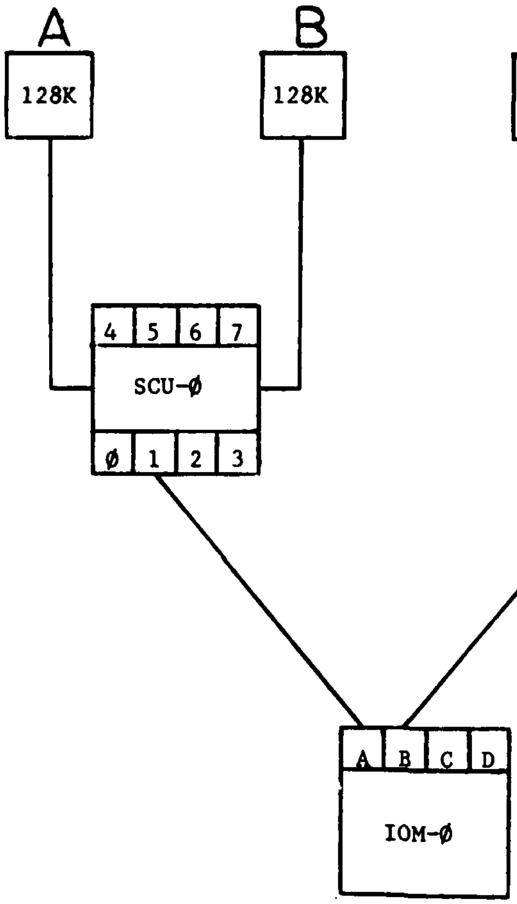
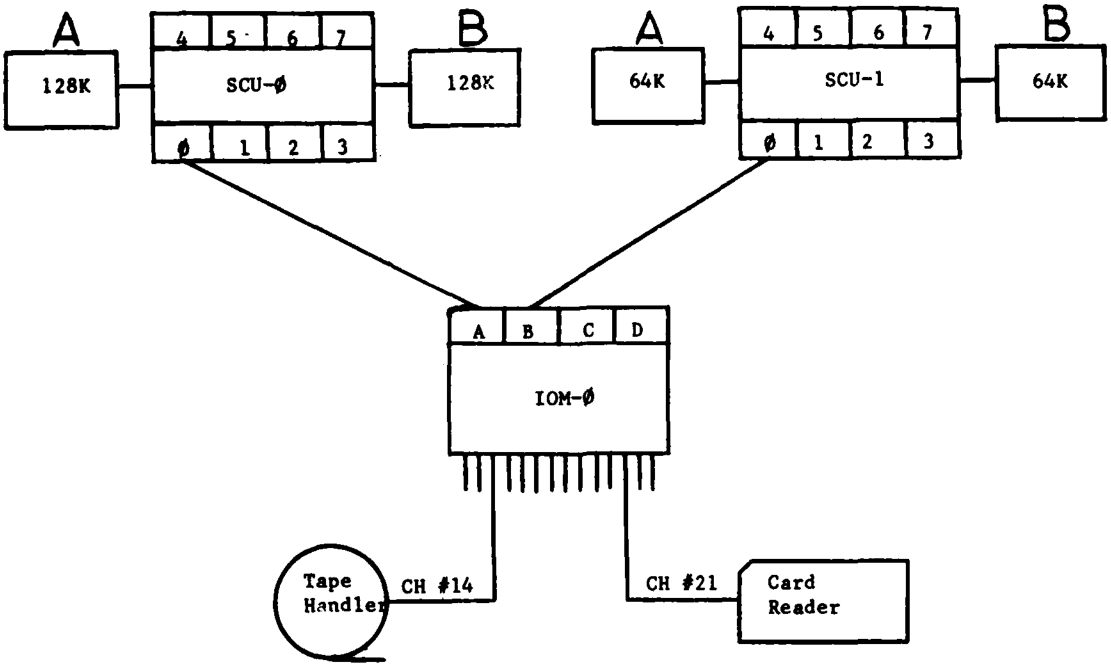


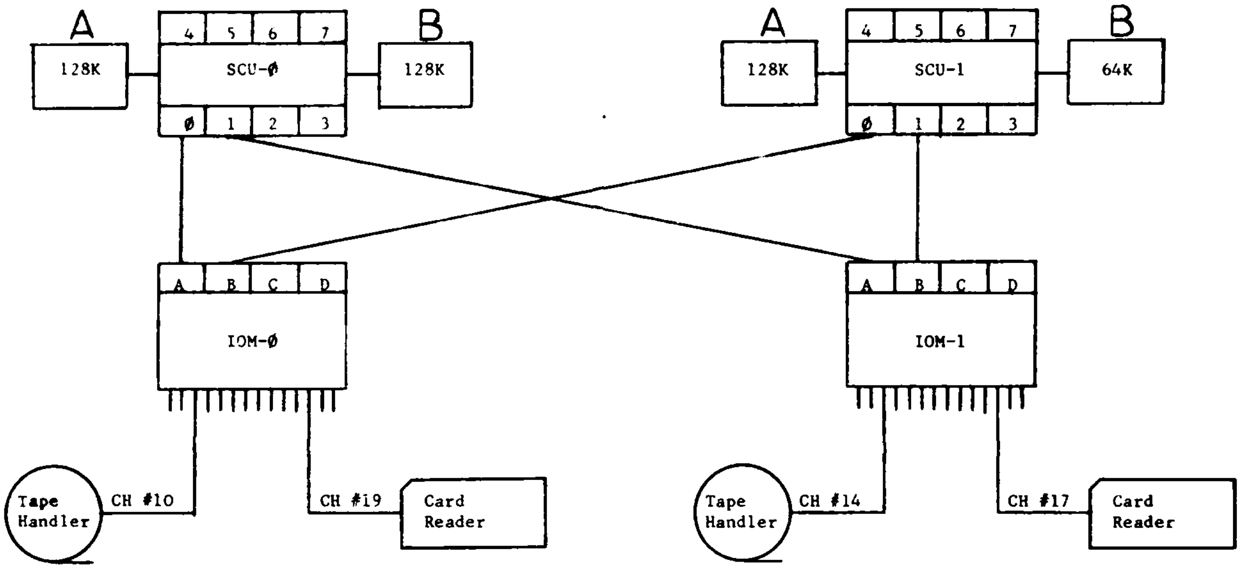
EXHIBIT OPR030-6



IOM-0 SWITCH SETTINGS

	STORE SIZE	PORT ASSIGNMENT	PORT ENABLE	INTERLACE	SYSTEM INITIALIZE ENABLE
PORT A	256K	000	ON	OFF	ON
PORT B	128K	010	ON	OFF	OFF

EXHIBIT OPR030-7



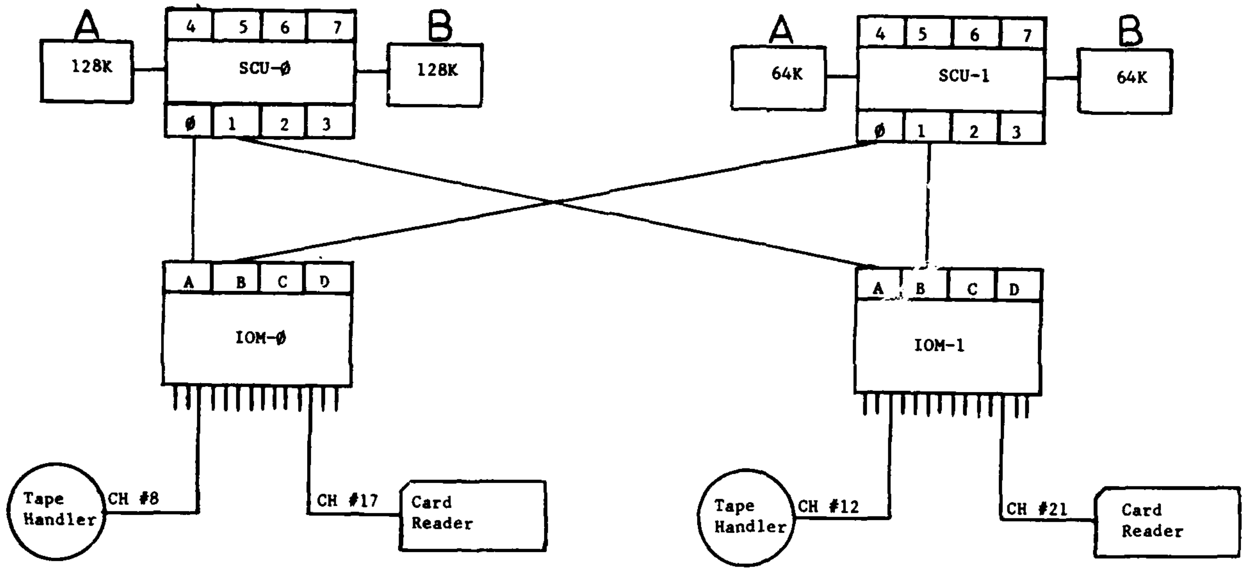
IOM-O SWITCH SETTINGS

	STORE SIZE	PORT ASSIGNMENT	PORT ENABLE	INTERLACE	SYSTEM INITIALIZE ENABLE
PORT A	256K	000	ON	OFF	ON
PORT B	256K	001	ON	OFF	OFF

IOM-1 SWITCH SETTINGS

	STORE SIZE	PORT ASSIGNMENT	PORT ENABLE	INTERLACE	SYSTEM INITIALIZE -NABLE
PORT A	256K	000	ON	OFF	ON
PORT B	256K	001	ON	OFF	OFF

EXHIBIT OPR030-8



IOM-O SWITCH SETTINGS

	STORE SIZE	PORT ASSIGNMENT	PORT ENABLE	INTERLACE	SYSTEM INITIALIZE ENABLE
PORT A	256K	000	ON	OFF	ON
PORT B	128K	010	ON	OFF	OFF

IOM-1 SWITCH SETTINGS

	STORE SIZE	PORT ASSIGNMENT	PORT ENABLE	INTERLACE	SYSTEM INITIALIZE ENABLE
PORT A	256K	000	ON	OFF	ON
PORT B	128K	010	ON	OFF	OFF

EXHIBIT OPR030-9

LESSON OPR030 OUTLINE

IOM CONFIGURATION PANEL

A. The IOM configuration panel contains the following five switches for each of the four input/output ports:

1. Store Size:

a. Purpose: To control the core memory size for each port.

b. Functional Positions:

Position	Total of SCU "Size" switches
32K	32K
64K	64K
128K	96K, 128K
256K	160K, 192K, 224K, 256K

c. Rule: Set equal to the total of the two SCU "Size" switches.

2. Port Assignment:

a. Purpose: In conjunction with the "Store Size" switch, will assign a group of memory addresses to be recognized by an IOM port.

b. Three toggle switches per port:

- binary representation of an octal number

Port Assignment switch	Ø	1	2
Binary value (if on)	4	2	1

- each toggle switch has two functional positions:

Ø (down) - the bit is off
1 (up) - the bit is active

c. Rule:
- actual memory size determines the order in which the "Port Assignment" switches are to be set.
- the port connected to the largest amount of memory must contain memory address ØØØØØØ.
- the position of the "Store Size" switches are used to determine the "Port Assignment" switch settings.
- divide the total of the previously established port's "Store Size" switch settings by the "Store Size" switch setting of the current port.
- the value will be represented by its binary configuration in the "Port Assignment" switches.

EXHIBIT OPR030-10

3. Interlace:

a. Purpose: Allows interleaving of memory addresses by two ports.

b. Two position toggle switch:

On (up) - interleaving is enabled.
Off (down) - interleaving is disabled.

c. Rule:

- two ports must be enabled.
- both ports must have the same size memory.
- both ports must be alphabetically adjacent.
- both port's "Port Assignment" switches Ø and 1 must be configured the same for each port.
- both port's "Port Assignment" switch 2 must be configured opposite of each other.

4. Port Enable:

a. Purpose: Logically connects the port to the System Controller Unit (SCU).

b. Two position toggle switch:
On (up) - port is connected to the SCU.
Off (down) - port is disconnected from the SCU.

c. Rule: Must be set to the "ON" position to allow the other port's switches to function properly.

5. System Initialize Enable:

a. Purpose: Allows the IOM to be initialized by a signal from the System Controller Unit (SCU).

b. Two position toggle switch:

On (up) - allows the IOM to be initialized.
Off (down) - the initialize signal is inhibited.

c. Rule: The port containing memory address ØØØØØØ must be set to the "ON" position.

B. The following four switches establish the parameters for the bootload sequence:

NOTE: In a multiple IOM system, these switches must be set to the same positions as IOM-Ø's on each of the additional IOM's.

1. Bootload Source:

a. Purpose: Selects the source of the bootload.

b. Two position toggle switch:

Tape - bootload source is magnetic tape.
Card - bootload source is cards.

2. Card Channel Number:

a. Purpose: Defines the channel of IOM-Ø to which the card reader that contains the bootload instructions is connected.

b. Six toggle switches:
- binary representation of an octal number.
- each toggle switch has two positions:

Ø (down) - the bit is off.
1 (up) - the bit is active.

c. Rule:
- the channel number is indicated on the $ IOM card within the Hardware Configuration Section of the startup deck.
- the channel number is always decimal and must be converted to octal.
- the binary configuration of the octal number is set in these switches.

NOTE: Present system restrictions limit the channel number to octal 37 (decimal 31).

3. Tape Channel Number:

a. Purpose: Defines the channel of IOM-Ø to which the magnetic tape handler that contains the bootload instructions is connected.

b. Six toggle switches:
- binary representation of an octal number.
- each toggle switch has two positions:

Ø (down) - the bit is off.
1 (up) - the bit is active.

c. Rule:
- the channel number is indicated on the $ IOM card within the Hardware Configuration Section of the startup deck.
- the channel number is always decimal and must be converted to octal.
- the binary configuration of the octal number is set in these switches.

NOTE: Present system restrictions limit the channel number to octal 37 (decimal 31).

4. Bootload Port:

a. Purpose: Defines the port number of the System Controller Unit (SCU) that controls memory address ØØØØØØ to which IOM-Ø is connected.

b. Three toggle switches:
- binary representation of the octal number for the SCU's port number.
- each toggle switch has two positions:

Ø (down) - the bit is off. 1 (up) - the bit is active.

C. The following two switches are used to initiate the bootload sequence:

1. System Initialize:

a. Purpose: Use to transmit a system initialize signal to the SCU.

b. Pushbutton (pressed) - transmits a system initialize signal to the SCU. The SCU transmits the initialize signal to all active modules.

NOTE: This switch must be pressed prior to beginning the bootload sequence. The initialize signal is used to set all active main frame modules to a known state so that the bootload sequence can begin.

2. Bootload:

a. Purpose: To initiate the bootload sequence.

b. Pushbutton (pressed) - the bootload sequence is initiated.

D. The following three switches allow you to interface the IOM with the system's software:

1. IOM Base Address:

a. Purpose: Used to set the base address of the IOM.

b. 18-bit word:

bits Ø-5 - toggle switches always set to zeroes.

bits 6-11 - toggle switches set to the binary configuration of the first two digits for the IOM's base address octal number.

bits 12-17 - internally wired to zeroes.

c. Rule:

IOM-Ø's base address: always octal number 14ØØØ.
IOM-1 through IOM-3's base address: add 6ØØ octal locations to the previous IOM's base address.

2. **Program Interrupt Base:**

a. Purpose: Used to set the Program Interrupt Base Address of the IOM.

b. 18-bit word:

bits ∅-5 - toggle switches always set to zeroes.
- also common to the IOM Base Address.
bits 6-11 - set to the binary configuration for the first two digits of the address (13)4∅.
bits 12-14 - set to the binary configuration for the third digit of the address 13(4)∅.
bits 15-17 - labeled "IOM Number."

NOTE: The program Interrupt Base Address for all IOM's is 1340.

3. **IOM Number:**

a. Purpose: Used to indicate the number of the IOM within the system.

b. Three bits:

bit 15 - internally wired to zero.

bit 16-17 - two toggle switches:

IOM Number	Binary Configuration
∅	∅∅
1	∅1
2	1∅
3	11

E. The following two switches pertain to hardware problems:

1. **Alarm Disable:**

a. Purpose: Used to enable/disable the alarm bell.

b. Two position toggle switch:

- On (up) - the alarm bell is disabled.
- Off (down) - the alarm bell is enabled.

2. **Normal/Test:**

a. Purpose: Controls the operation of the maintenance panel test logic.

b. Two position toggle switch:

Normal (down) - the maintenance panel test logic is disabled.
Test (up) - the maintenance panel test logic is enabled.

OPR040 TEST INSTRUCTIONS

STEP 1. Set all of the necessary switches on each of the DIA configurations panels to enable both of the DATANET 355's to function within the system.

STEP 2. Set the appropriate switches to show that 355-Ø's DIA channel number is 15.

STEP 3. Set the appropriate switches to show that 355-1's DIA channel number is Ø4.

STEP 4. Set the appropriate switches to ensure that the 355 computer mailbox address is 454 for both DATANET 355's.

STEP 5. Return to lesson OPR040 by typing "?GET OPR040 OLD".

System Modules:

1-SCU
2-IOM's
2-355's

Physical Cabling of the System Main Frame Modules:

IOM-0 PORT A cabled to PORT Ø of SCU-Ø
IOM-1 PORT B cabled to PORT 1 of SCU-Ø

The Following Peripheral Devices are Configured to the IOM's:

IOM-Ø
Channel 17 - DATANET 355-Ø

IOM-1
Channel 13 - DATANET 355-1

EXHIBIT OPR040-1

DIRECT INTERFACE ADAPTER (DIA) CONFIGURATION PANEL

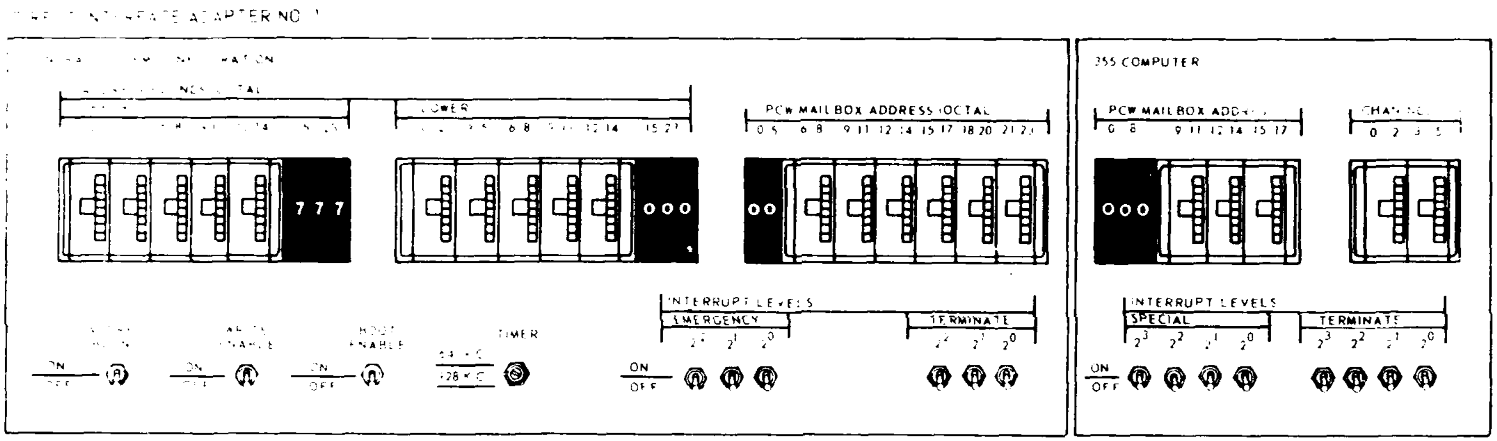


EXHIBIT OPR040-2

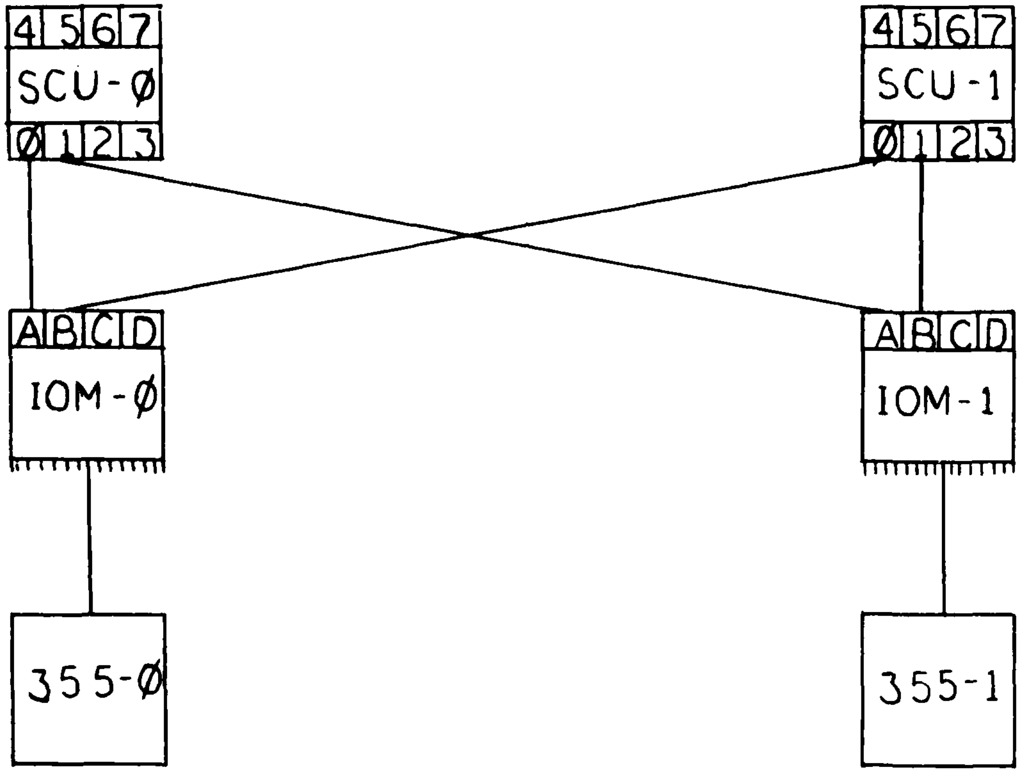


EXHIBIT OPR040-3

LESSON OPR040 OUTLINE

355 DIA CONFIGURATION PANEL

A. CENTRAL SYSTEM CONFIGURATION

1. Address Bounds (Octal):

a. Upper - 77777

b. Lower - 00000

2. PCW Mailbox Address (Octal):

a. Purpose: Used to set the base address of the Datanet 355.
b. Six thumbwheel switches: set to represent the base address.
c. Rule:

355-0 - add 600 octal locations to the last IOM's base address.
355-1 through 355-2 - add 100 octal locations to the previous 355's base address.

3. Address Bounds - Off

4. Write Enable - On

5. Boot Enable - On

6. Timer - Off

7. Interrupt Levels

a. Emergency - 111 (7)
b. Terminate - 111 (7)

B. 355 COMPUTER

1. PCW Mailbox Address (Octal):

a. Set to a value in the range 454-500.
b. Normally set to the value "454".

2. Chan No:

a. Set to a Channel number not in use.
b. Normally set to the value "04".

3. Interrupt Levels:

a. Special - 0011 (3)

b. Terminate - 0010 (2)

EXHIBIT OPR040-4

CPR050 TEST INSTRUCTIONS

STEP 1. Set all of the necessary switches on each of the Processor configuration panels to enable both the Processors to function within the system.

STEP 2. Set the appropriate switch to prevent the alarm bell from ringing for an over temperature condition on PRO-0.

STEP 3. Set the appropriate switch to allow the alarm bell to ring when an over temperature condition occurs in PRO-1.

STEP 4. Set the appropriate switch to allow the maintenance personnel to use the Maintenance Panel Test Logic on PRO-1.

STEP 5. Return to lesson OPR050 by typing "?GET OPR050 OLD".

SYSTEM MODULES:

1-SCU (128K)
2-IOMs
2-355s
2-PROs

Physical cabling of the system main frame modules:

IOM-0 Port A cabled to Port 0 of SCU-0.
IOM-1 Port B cabled to Port 1 of SCU-0.
PRO-0 Port C cabled to Port 6 of SCU-0. (Control Processor)
PRO-1 Port D cabled to Port 7 of SCU-0.

The following peripheral devices are configured to the IOMs:

IOM-0:

Datanet 355-0 - Channel 17

IOM-1:

Datanet 355-1 - Channel 13

EXHIBIT OPR050-1

CP6000 CONFIGURATION PANEL

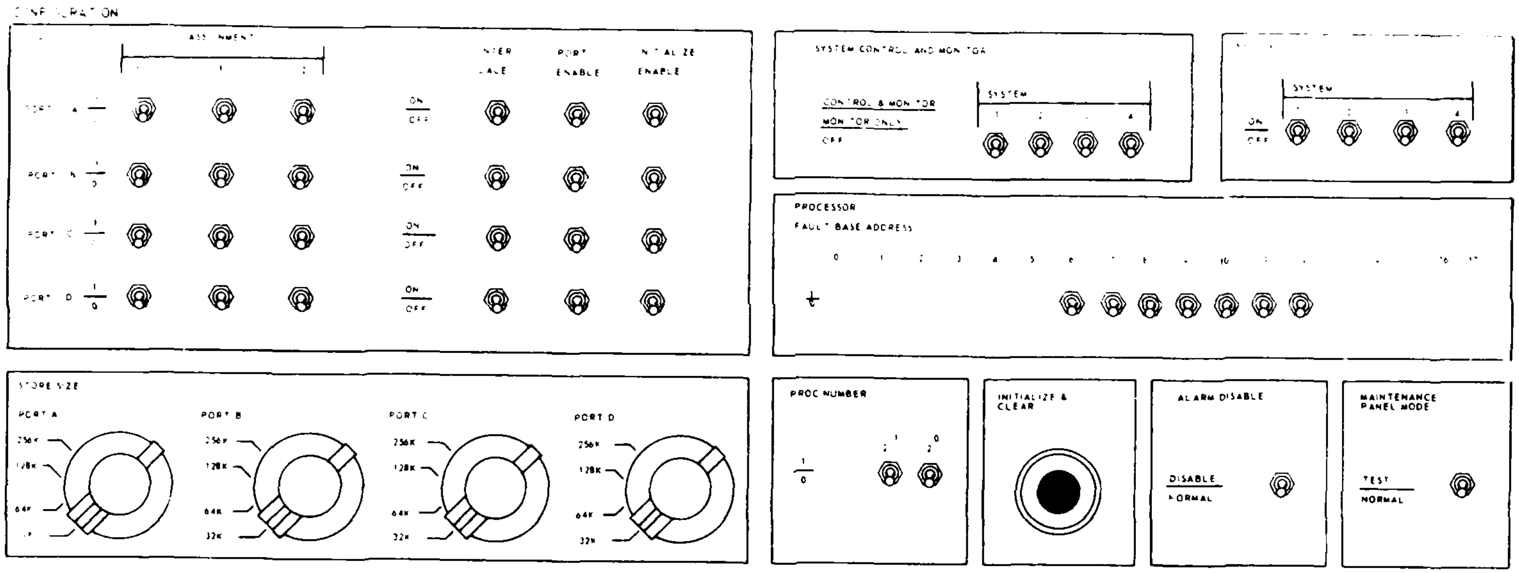
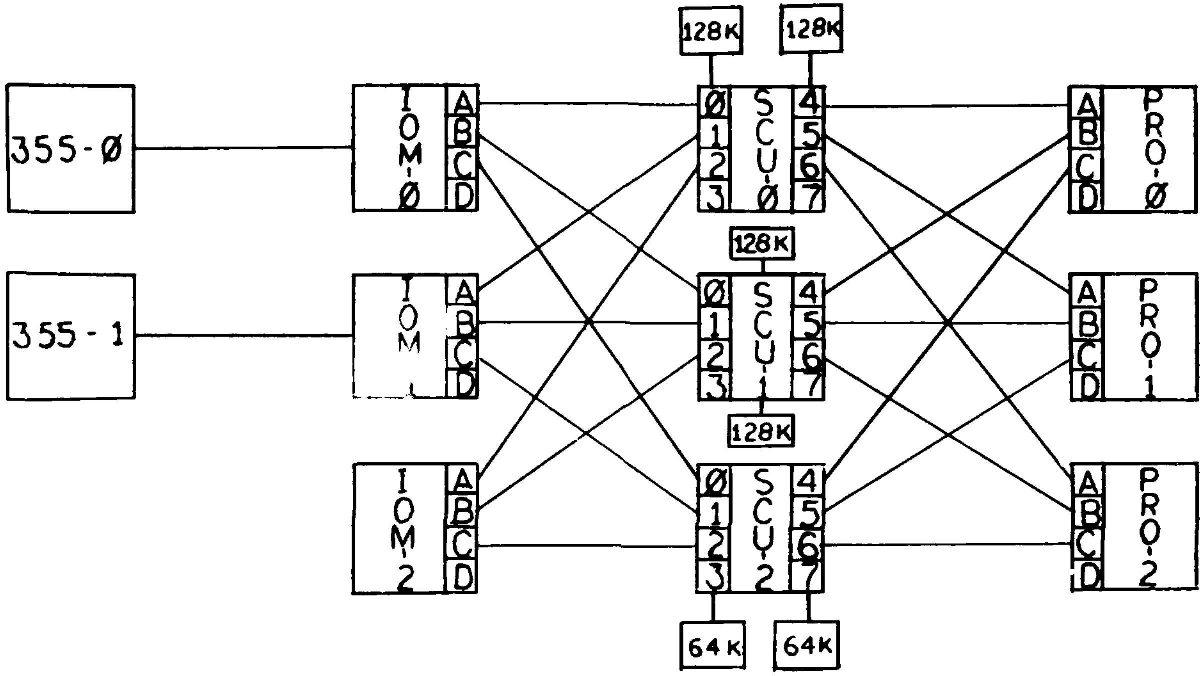


EXHIBIT OPR050-2



IOM-0 SWITCHES	PORT A	PORT B	PORT C	PORT D
STORE SIZE	256K	256K	128K	32K
PORT ASSIGNMENT	000	001	100	000
INTERLACE	ON	ON	OFF	OFF
PORT ENABLE	ON	ON	ON	OFF
SYSTEM INITIALIZE ENABLE	ON	OFF	OFF	OFF

IOM-1	PORT A	PORT B	PORT C	PORT D
STORE SIZE	256K	256K	128K	32K
PORT ASSIGNMENT	000	001	100	000
INTERLACE	ON	ON	OFF	OFF
PORT ENABLE	ON	ON	ON	OFF
SYSTEM INITIALIZE ENABLE	ON	OFF	OFF	OFF

IOM-2 SWITCHES	PORT A	PORT B	PORT C	PORT D
STORE SIZE	256K	256K	32K	128K
PORT ASSIGNMENT	000	001	000	100
INTERLACE	ON	ON	OFF	OFF

IOM-2 SWITCHES (CONT)	PORT A	PORT B	PORT C	PORT D
PORT ENABLE	ON	ON	OFF	ON
SYSTEM INITIALIZE ENABLE	ON	OFF	OFF	OFF

EXHIBIT OPR050-3

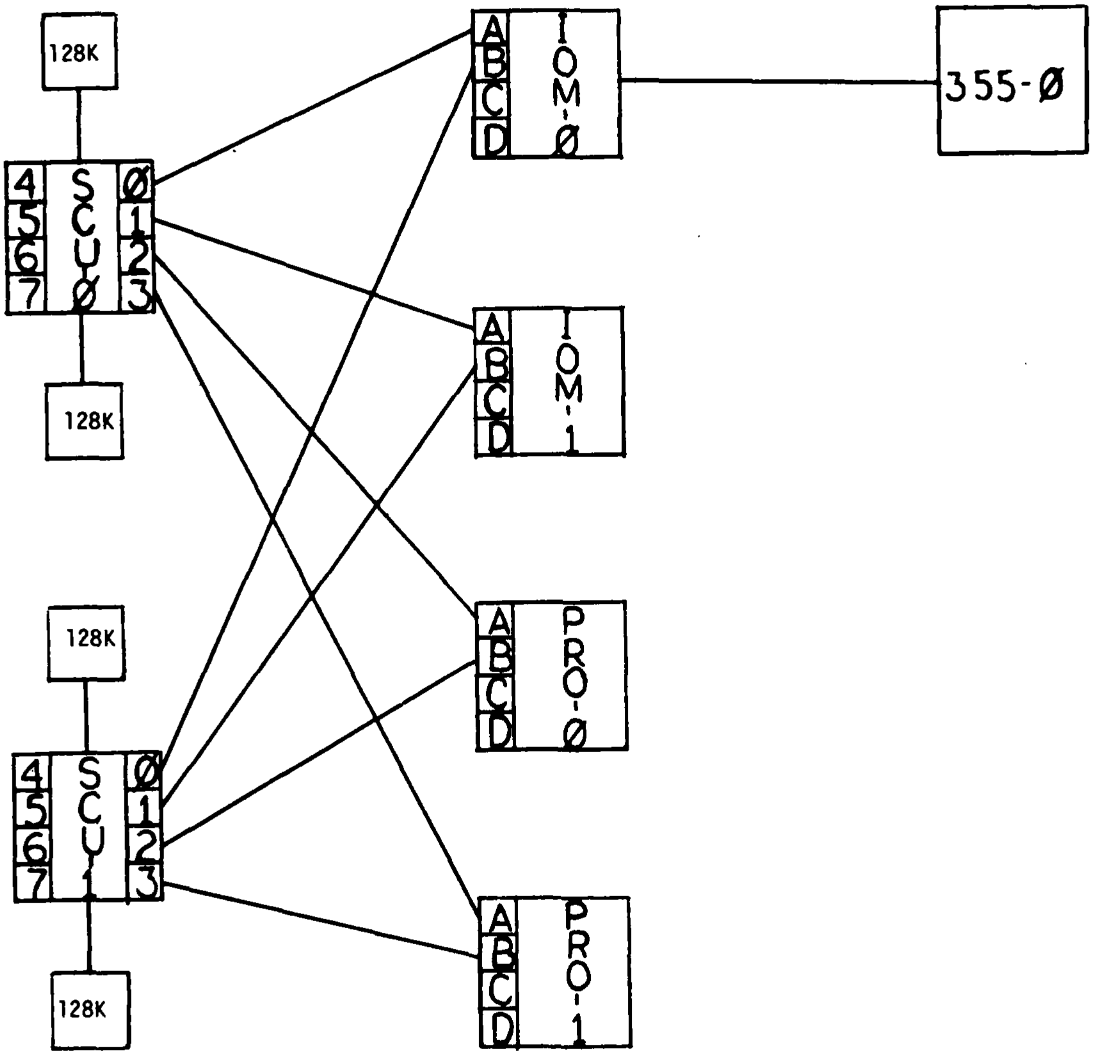


EXHIBIT OPR050-4

LESSON OPR050 OUTLINE

PROCESSOR CONFIGURATION PANEL

A. The Processor configuration panel contains the following five switches for each of the four input/output ports:

1. Store Size:

a. Purpose: To control the core memory size for each port.

b. Functional Positions:

Position	Total of SCU "Size" switches
32K	32K
64K	64K
128K	96K, 128K
256K	160K, 192K, 224K, 256K

c. Rule: Set equal to the total of the two SCU "Size" switches.

2. Port Assignment:

a. Purpose: In conjunction with the "Store Size" switch, will assign a group of memory addresses to be recognized by a Processor port.

b. Three toggle switches per port:

- binary representation of an octal number

Port Assignment switch	∅	1	2
Binary value (if on)	4	2	1

c. Rule:

- actual memory size determines the order in which the "Port Assignment" switches are to be set.
- the port connected to the largest amount of memory must contain memory address ∅∅∅∅∅∅.
- the position of the "Store Size" switches are used to determine the "Port Assignment" switch settings.
- divide the total of the previously established port's "Store Size" switch settings by the "Store Size" switch setting on the current port.
- the value will be represented by its binary configuration in the "Port Assignment" switches.

EXHIBIT OPR050-5

3. Interlace:

a. Purpose: Allows interleaving of memory addresses by two ports.

b. Two position toggle switch:

On (up) - interleaving is enabled.
Off (down) - interleaving is disabled.

c. Rule:

- two ports must be enabled.
- both ports must have the same size memory.
- both ports must be alphabetically adjacent.
- both port's "Port Assignment" switches Ø and 1 must be configured the same for each port.
- both port's "Port Assignment" switch 2 must be configured opposite of each other.

4. Port Enable:

a. Purpose: Logically connects the port to the System Controller Unit (SCU).

b. Two position toggle switch:

On (up) - port is connected to the SCU.
Off (down) - port is disconnected from the SCU.

c. Rule: Must be set to the "ON" position to allow the other port's switches to function properly.

5. Initial'ze Enable:

a. Purpose: Allows the Processor to be initialized by a signal from the System Controller Unit (SCU).

b. Two position toggle switch:

On (up) - allows the Processor to be initialized.
Off (down) - the initialize signal is inhibited.

c. Rule: The port containing memory address ØØØØØØ must be set to the "ON" position.

B. The following two switches interface the Processor with the system's software:

1. Processor Fault Base Address Switches:

a. Purpose: Sets the fault base address of the Processor.

b. 18-bit word:

bits 0-5 - preset to zeroes.
bits 6-12 - toggle switches set to the binary configuration of the first three digits of the Processor's fault base address.
bits 13-17 - preset to zeroes.

c. Computing Processor fault base addresses:

PRO-0 - Add 100 octal locations to the last Datanet 355's PCW Mailbox Address.

PRO-1, PRO-2 and PRO-3

- Add 40 octal locations to the previous Processor's Fault Base Address.

2. Proc Number:

a. Purpose: Designates the Processor assignment number within a multiprocessor system.

b. Binary configuration:

Processor	Processor Assignment Number	Binary Configuration
PRO-Ø	Ø	ØØ
PRO-1	1	Ø1
PRO-2	2	1Ø
PRO-3	3	11

C. The following two switches pertain to hardware problems:

1. Alarm Disable:

   a. Purpose: Used to enable/disable the alarm bell.

   b. Two position toggle switch:

      On (up) - the alarm bell is disabled.
      Off (down) - the alarm bell is enabled.

2. Maintenance Panel Mode:

   a. Purpose: controls the operation of the maintenance panel test logic.

   b. Positions:

      Normal - the maintenance panel test logic is disabled.
      Test - the maintenance panel test logic is enabled.

NOTE: In either position, the Processor is still able to function within the system.

OBJECTIVES

When you have completed the exercises in this exhibit, you will be able to:

1. Apply power to each hardware module.

2. Configure each hardware module for specified system configuration.

PROCEDURES

Accomplish each exercise in this exhibit as directed by your training monitor. Use any training literature, including the procedures presented in lessons OPR010 through OPR050 to accomplish these exercises. Each exercise will be accomplished in accordance with local standards and procedures. You will be spot-checked by your training monitor during your performance of the exercises.

EQUIPMENT

A Honeywell Series 6000 Computer System.

EXERCISE 1

Perform configuration of the hardware modules by completing the following tasks:

1. Apply power to each hardware module.

2. Configure each hardware module by setting the appropriate switches on the configuration portion of the maintenance panel of each module.

EXERCISE 2

Perform reconfiguration of the hardware modules by completing the following tasks:

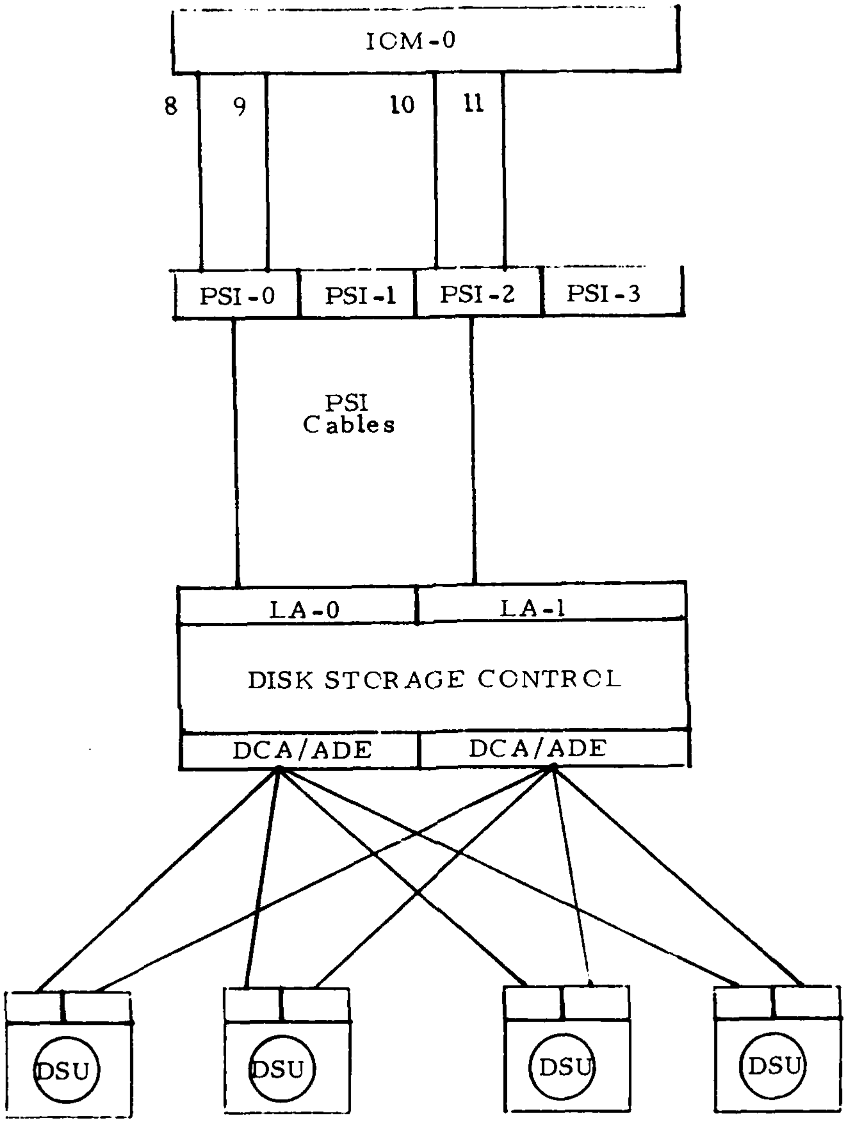
1. Remove a faulty block of memory using the System Controller Unit "Address Offset" switch.

2. Remove a Memory Unit from the system.

3. Remove an Input/Output Multiplexer (IOM) from the system.

4. Remove a Processor from the system.

SINGLE MPC/DUAL CABLE DISK SUBSYSTEM
(DSS181)



DCA/ADE - Disk Control Adaptor/Additional Drive Electronics.

DSU - Disk Storage Units.

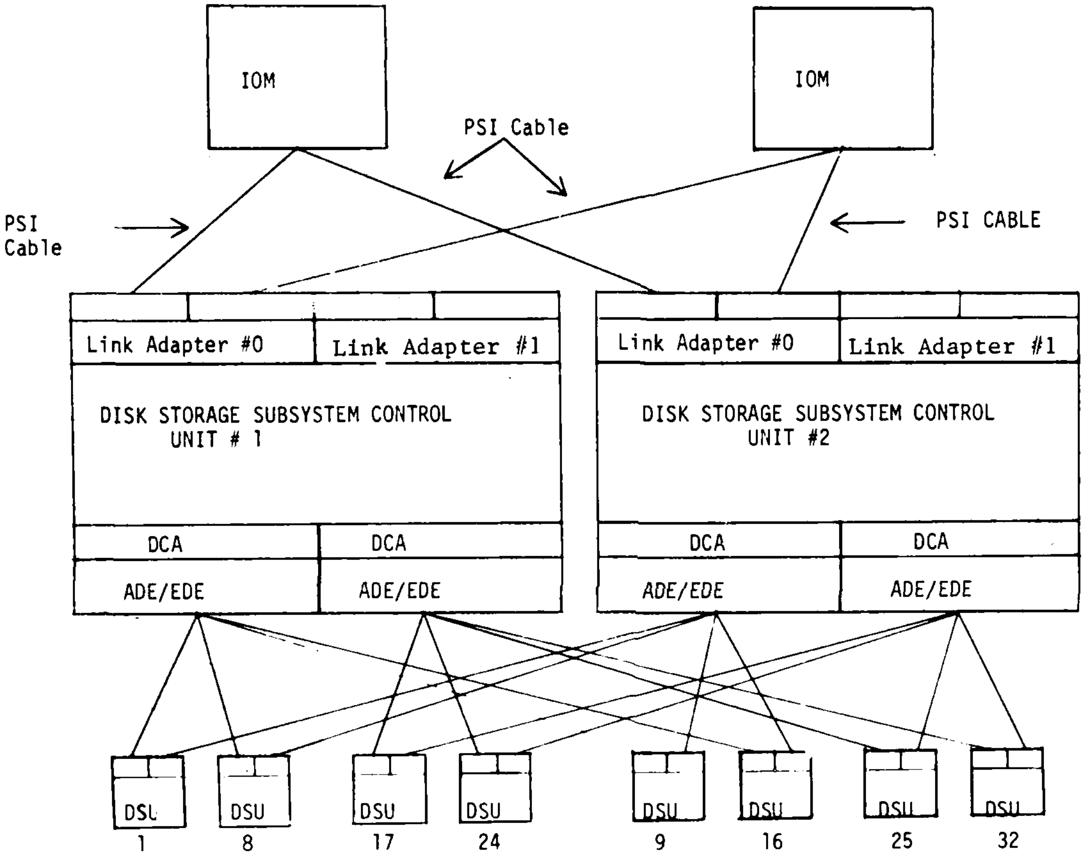
Note 1: The DCA/ADE connects the DSC to the Disk Storage Untis.

Note 2: Observe that two PSI cables connect the four IOM channels to the Disk Storage Controller.

FORMAT: $ XBAR IOM-0, PUB-8, PUB-9, PUB-10, PUB-11

EXHIBIT OPR060-1

DUAL CONTROL SUBSYSTEM - DSS181

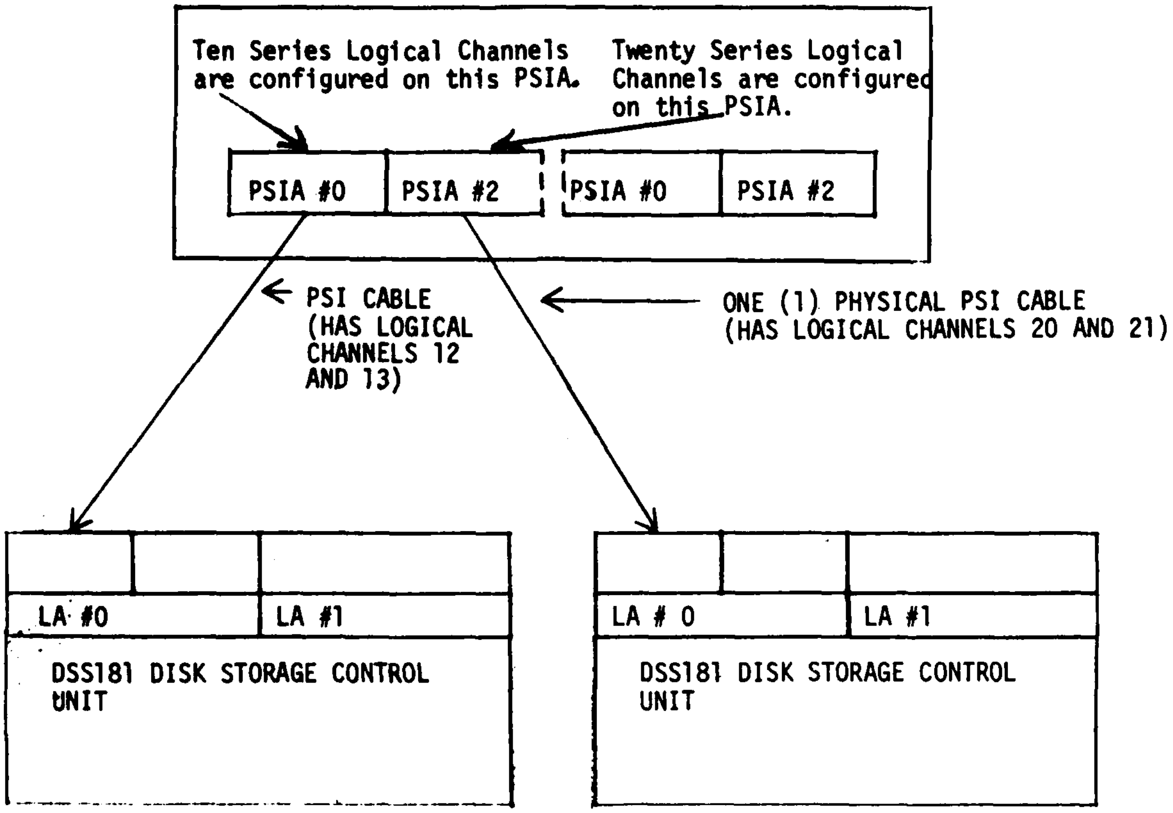


Note 1: Observe that there are two (DUAL) Disk Storage Subsystems.

EXHIBIT OPR060-2

PSI/PSIA ARRANGEMENT - DUAL CONTROL DSS181 SUBSYSTEM

(IOM - 0)


Ten Series Logical Channels are configured on this PSIA.
Twenty Series Logical Channels are configured on this PSIA.
PSIA #0
PSIA #2
PSIA #0
PSIA #2
PSI CABLE (HAS LOGICAL CHANNELS 12 AND 13)
ONE (1) PHYSICAL PSI CABLE (HAS LOGICAL CHANNELS 20 AND 21)
LA #0
LA #1
DSS181 DISK STORAGE CONTROL UNIT
LA # 0
LA #1
DSS181 DISK STORAGE CONTROL UNIT

LA - LINK ADAPTER

NOTE 1: Observe that only the primary PSIAs are indicated. Hence PSIAs number 1 and 3 are not shown here.

NOTE 2: Two (DUAL) Disk Subsystems are configured.

SAMPLE HARDWARE CONFIGURATION SECTION

```
1      8       16

$CONFIG
$      SYID    AIR FORCE-1
$      TRACE   0,0
$      DATE    080875
$      MCT-0   128,PORT-1,IOM-0,PORT-4,355-0.
$      ETC     PORT-6,PRO-0
$      IOM-0   PUB-27,355-0,LINES-75
$      IOM-0   PUB-9,TAPE,UNITS-8,
$      ETC     UNIT-1,
$      ETC     UNIT-2,1T1,
$      ETC     UNIT-3,1T2,
$      ETC     UNIT-4,1T3,
$      ETC     UNIT-5,1T4, ASA9,
$      ETC     UNIT-6,1T5,
$      ETC     UNIT-7,1T6,
$      ETC     UNIT-8,1T7,
$      IOM-0   PUB-11,READER
$      IOM-0   PUB-12,DISC*180,UNITS-4,LK-355,CH-3,
$      ETC     STRING-0,UNIT-1,DP1,
$      ETC     STRING-0,UNIT-2,DP2.
$      ETC     STRING-1,UNIT-3,DP3,
$      ETC     STRING-1,UNIT-4,DP4,
$      IOM-0   PUB-13,DISC*270,UNITS-2,
$      ETC     UNIT-1,ST1,MODULES-2,UNIT-2,DC1,MODULES-3
$      IOM-0   PUB-14,DISC*270,UNITS-1,
$      ETC     UNIT-1,DS2,MODULES-4
$      IOM-0   PUB-15,PUNCH*300,PU2
$      IOM-0   PUB-30,PRINTER,PR1
$      IOM-0   PUB-31,CONSOLE,TY1,TY2,TY3,TY4
$      XBAR    IOM-0,PUB-9,PUB-26
$      XBAR    IOM-0,PUB-13,PUB-24
$      GCOSFIOLST1,DP1,DP2,DP3
$      AUTOLD  ST1,200
***EOF
```

EXHIBIT OPR060-4

NARRATIVE -- H6000 SIMULATED COMPUTER SYSTEM

System Identification: WWMCCS-1
TRACES: Include ALL TRACE types

Startup INFO ($ INFO Card) Parameters: See Items A, B, C below.

A. Sieve Limits

(1) Tapes - 3 (2) LLinks - 500 (3) PRC Time - 0399

B. Secure

C. SNEAKON and LUMP Files are on Disk Pack ST1.

Startup/Dump Program Response ($ ANSWER Card) Parameters:

ACTION	RESPONSE	ACTION	RESPONSE
Restart	No	Bootload	Yes
SSCLEAR	Yes	Initialize	Partial
INIT	No	SCF	No

System Modules: One SCU (128K - Stores A and B 64K Each)
Two IOM's
Two Processors
Control Processor is PRO-0

Physical Cabling of System Modules:

Port A of IOM-0 is cabled to Port 0 of the SCU
Port B of IOM-1 is cabled to Port 1 of the SCU
Port C of PRO-0 is cabled to Port 6 of the SCU
Port D of PRO-1 is cabled to Port 7 of the SCU

The following peripheral devices are connected to IOM-0:

MICROPROGRAMMABLE PERIPHERAL CONTROLLER NUMBER ZERO (MPC-0)

PUB 8 - One DSS181 Disk Subsystem with DUAL CABLES
Dual Channel Crossbarred
Three PERM Packs
Three Removable (RMVBL) Packs

Disk Pack Unit Number	Logical Name	Status
Unit 1	ST1	PERM
Unit 2	DP1	PERM
Unit 3	DP2	PERM
Unit 4	DP3	REMOVABLE
Unit 5	DP4	REMOVABLE
Unit 6	DP5	REMOVABLE

MICROPROGRAMMABLE PERIPHERAL CONTROLLER NUMBER ONE (MPC-1)

PUB 12 - One MTS5ØØ Magnetic Tape Subsystem with DUAL CABLES
Dual Channel Crossbarred
Two 7-Track Tape Units
Four 9-Track Tape Units

Tape Unit Number	Logical Name	Status
Unit 1	1TØ	7 Track
Unit 2	1T1	7 Track
Unit 3	1T2	9 Track
Unit 4	1T3	9 Track
Unit 5	1T4	9 Track
Unit 6	1T5	9 Track

The following additional peripherals are connected to IOM-0:

Channel Number	Logical Name	Number Devices/Type
PUB 14	PR2	One Printer*3ØØ (PRT3ØØ)
PUB 15	CP1	One Punch (CPZ2Ø1)
PUB 16	CR1	One Reader (CRZ1Ø1)
PUB 17		One FNP 355-Ø, Lines-12
PUB 3Ø	PR1	One Printer (PRT2Ø1)
PUB 31	TY1,TY2,TY3,TY4	One Control Console

The following peripheral devices are connected to IOM-1:

MICROPROGRAMMABLE PERIPHERAL CONTROLLER NUMBER TWO (MPC-2)

PUB 8 - One DSS181 Disk Subsystem
Single Channel Crossbarred
Two PERM Packs
Four Removable (RMVBL) Packs

Disk Pack Unit Number	Logical Name	Status
Unit 1	2P1	PERM
Unit 2	2P2	PERM
Unit 3	2P3	REMOVABLE
Unit 4	2P4	REMOVABLE
Unit 5	2P5	REMOVABLE
Unit 6	2P6	REMOVABLE

MICROPROGRAMMABLE PERIPHERAL CONTROLLER NUMBER THREE (MPC-3)

PUB 10 - One MTS5ØØ Magnetic Tape Subsystem
Single Channel Crossbarred
Six 9-Track Tape Units

Tape Unit Number	Logical Name	Status
Unit 1	2TØ	9 Track
Unit 2	2T1	9 Track
Unit 3	2T2	9 Track
Unit 4	2T3	9 Track
Unit 5	2T4	9 Track
Unit 6	2T5	9 Track

The following additional peripherals are connected to IOM-1:

Channel Number	Logical Name	Number Devices/Type
PUB 12	CP2	One Punch (CPZ2Ø1)
PUB 13	CR2	One Reader (CRZ2Ø1)
PUB 18		One FNP 355-1, Lines 1Ø
PUB 3Ø	PR3	One Printer*3ØØ (PRT3ØØ)
PUB 31		One Console

The following additional system definitions apply:

A. STARTUP FILES: Device names ST1, DP1, DP2, 2P1, and 2P2 must contain files that the Startup Program will use. The scan for these files will be ST1, DP1, DP2, 2P1 and then 2P2.

B. AUTOLD FILE: Reserve 2ØØ LLinks of disk file space on device ST1. The AUTOLD file will be placed on ST1 to provide automatic reboot of the system.

CARD DEFINITION - H6000 SIMULATED COMPUTER SYSTEM

Hardware Configuration ($CONFIG) Section

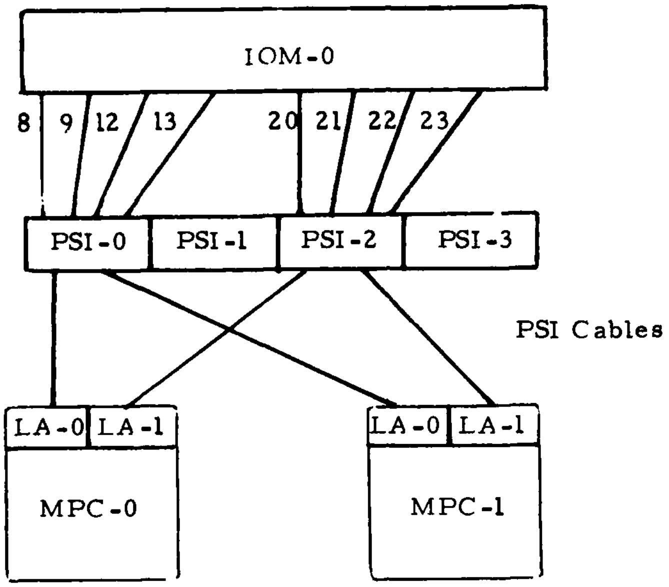
```
1      8       16

$CONFIG
$      SYID    WWMCCS-1
$      TRACE   0.0
$      ANSWER  RESTART/NO,SSCLEAR/YES,BOOTLOAD/YES,
$      ETC     INITIALIZE/PARTIAL,INIT/NO,SCF/NO
$      DATE    031975
$      MCT-0   128,PORT-0,IOM-0,PORT-1,IOM-1,
$      ETC     PORT-6,PRO-0,PORT-7,PRO-1
$      IOM-0   PUB-8,DISC*181,UNITS-6,
$      ETC     UNIT-1,ST1,
$      ETC     UNIT-2,DP1,
$      ETC     UNIT-3,DP2,
$      ETC     UNIT-4,DP3,RMVBL,
$      ETC     UNIT-5,DP4,RMVBL,
$      ETC     UNIT-6,DP5,RMVBL
$      IOM-0   PUB-12,TAPE*MPC9,UNITS-6,
$      ETC     UNIT-1,1T0,TAPE*MPC7,
$      ETC     UNIT-2,1T1,TAPE*MPC7,
$      ETC     UNIT-3,1T2,
$      ETC     UNIT-4,1T3,
$      ETC     UNIT-5,1T4,
$      ETC     UNIT-6,1T5,
$      IOM-0   PUB-14,PRINTER*300,PR2,DEDICATED,TRAIN-2
$      IOM-0   PUB-15,PUNCH*300,CP1
$      IOM-0   PUB-16,READER*200,CR1
$      IOM-0   PUB-17,355-0,LINES-12
$      IOM-0   PUB-30,PRINTER,PR1
$      IOM-0   PUB-31,CONSOLE,TY1,TY2,TY3,TY4
$      IOM-1   PUB-8,DISC*181,UNITS-6,
$      ETC     UNIT-1,2P1,
$      ETC     UNIT-2,2P2,
$      ETC     UNIT-3,2P3,RMVBL,
$      ETC     UNIT-4,2P4,RMVBL,
$      ETC     UNIT-5,2P5,RMVBL,
$      ETC     UNIT-6,2P6,RMVBL,
$      IOM-1   PUB-10,TAPE*MPC9,UNITS-6,
$      ETC     UNIT-1,2T0,
$      ETC     UNIT-2,2T1,
$      ETC     UNIT-3,2T2,
$      ETC     UNIT-4,2T3,
$      ETC     UNIT-5,2T4,
$      ETC     UNIT-6,2T5,
$      IOM-1   PUB-12,PUNCH*300,CP2
$      IOM-1   PUB-13,READER*200,CR2
$      IOM-1   PUB-18,355-1,LINES-10
$      IOM-1   PUB-30,PRINTER*300,PR3,TRAIN-2
```

EXHIBIT OPR060-6

```
1      8       16

$      IOM-1   PUB-31,CONSOLE
$      XBAR    IOM-0,PUB-8,PUB-20,
$      ETC     IOM-0,PUB-9,PUB-21
$      XBAR    IOM-0,PUB-12,PUB-22,
$      ETC     IOM-0,PUB-13,PUB-23
$      XBAR    IOM-1,PUB-8,PUB-9
$      XBAR    IOM-1,PUB-10,PUB-11
$      MPC-0   SIZE-4,PSI-0,IOM-0,
$      ETC     PUB-8,PUB-9,
$      ETC     PSI-2,IOM-0,PUB-20,PUB-21
$      MPC-1   SIZE-4,PSI-0,IOM-0,
$      ETC     PUB-12,PUB-13,
$      ETC     PSI-2,IOM-0,
$      ETC     PUB-22,PUB-23
$      MPC-2   SIZE-4,PSI-0,ICM-0,
$      ETC     PUB-8,PUB-9
$      MPC-3   SIZE-4,PSI-0,IOM-1,
$      ETC     PUB-10,PUB-11
$      GCOSFIL ST1,DP1,DP2,2P1,2P2
$      AUTOLD  ST1,200
$      INFO    SLTAPE/3,SLINKS/500,SLTIME/0399
$      INFO    SNEAKON/ST1,LUMP/ST1,SECURE
***EOF
```

EXHIBIT OPR060-6 (Cont)

(DUAL MPC/DUAL CABLE CONFIGURATION)



(DUAL MPC/SINGLE CABLE CONFIGURATION)

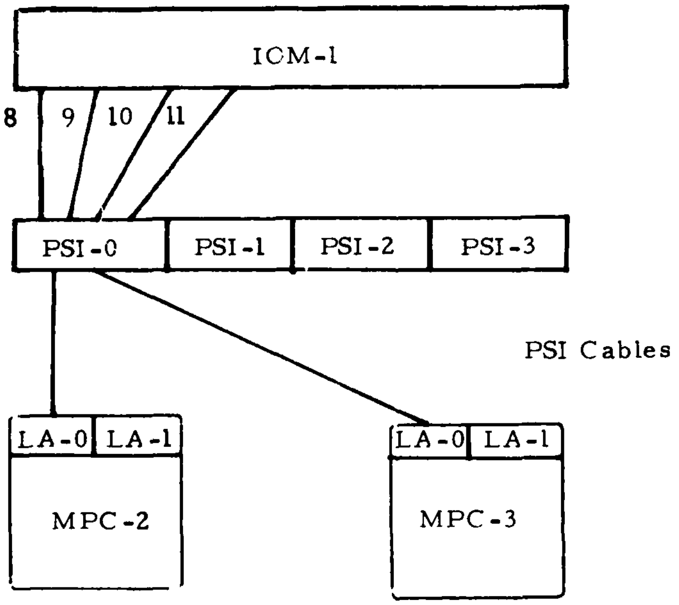


EXHIBIT OPR060-7

OBJECTIVES

When you have completed this exercise, you will be able to construct and keypunch the $CONFIG Section of a system startup deck.

PROCEDURES

Accomplish this exercise as directed by your training monitor. Use any training literature, including the information presented in lesson OPR060 to accomplish this exercise. This exercise will be accomplished in accordance with local standards and procedures. You may obtain assistance from your training monitor on the more difficult parts during your performance of this exercise.

EQUIPMENT

A keypunch machine.

EXERCISE

Construct and keypunch the $CONFIG section of the startup deck for your site's Series H6000 Computer System.

EXHIBIT OPR060-8

DEVICE INITIALIZATION SECTION-SIMULATED COMPUTER

```
1      8       16

$INITIALIZE
$      INIT    ST1,DP1,DP2,DP3,DP4,DP5
$      INIT    2P1,2P2,2P3,2P4,2P5,2P6
$      OBJECT  H6000J6,001       CON          MPCB
       .
       .       MPC BOOTLOAD DECK
       .
$      DKEND
$      OBJECT  HMPCJ1.001        DAT          M181
       .
       .       DSS181 FIRMWARE DECK
       .
$      DKEND
$      OBJECT  HMPCJ1.003        DAT          M500
       .
       .       MTS500 FIRMWARE DECK
       .
$      DKEND
$      READIN  1T0
***EOF
```

Note 1: Firmware decks for the remaining MPC-Driven subsystems shown in Exhibit OPR060-6 are read in from tape unit 1T0.

Note 2: Only one MPC Bootload Deck is required.

EXHIBIT OPR070-1

FILE EDIT SECTION - SIMULATED COMPUTER

```
1      8       16
$EDIT
$      FILDEF  ST1,GCOS-WMIXUSE,300/0,SYS,1T1
$      FILDEF  ST1,PWIN-USE,500/0,SYS,*
$      FILDEF  ST1,SOFTW-SYSLID,800/0,RDM,*
$      FILDEF  ST1,GCOS-HI-USE,800/0,SYS,*
$      FILDEF  DP1,GCOS-LO-USE,1800/0,SYS,*
$      FILDEF  DP2,T-AND-D,900/0,SYS,*
$      FILDEF  DP2,TSS-SUB-SYS,990/0,SYS,*
$      FILDEF  2P1,SOFTW-PRIME,1520/0,SYS,*
$      FILDEF  2P1,SOFTW-SECOND,2400/0,SYS,*
$      FILDEF  2P2,DMS-SOFTW,3000/0,SYS,*
$      FILDEF  DP1,FORTLIB,50/0.
$      ETC     RDM,*,LABEL/TSS-FORTLIBE
$      FILDEF  ST1,LUMP,120
$      FILDEF  2P2,BACKDOOR,12
$      FILDEF  2P2,SYOU1,3600
$      FILDEF  2P2,SYOU2,3600
$      SSFILE  ST1,MAX/35,MIN/10,.TASK/10,.TRANS/10,
$      ETC     .EXPRS/10/20,.HOLD/10,.NORM/10/20
***EOF
```

Note 1: Tape unit 1T1 of IOM-0, channel 12, contains the Total System Tape.

Note 2: All mass storage devices used in this section must have their logical names defined on the $ GCOSFIL card. Reference Exhibit OPR060-6.

EXHIBIT OPR070-2

SOFTWARE CONFIGURATION, PROGRAM PATCH AND GCOS MODULES SECTIONS

SIMULATED COMPUTER

```
1      8       16
$FILES
$      PFILES  FORTLIB
$      PFILES  BACKDOOR,LUMP
$      PFILES  DMS-SOFTW
$      SYSTEM  GCOS-WMIXUSE,GCOS-HI-USE,
$      ETC     GCOS-LO-USE,T-AND-D,SOFTW-PRIME,
$      ETC     TSS-SUB-SYSP-SOFTW-SECOND,DMS-SOFTW
$      LIBRARY RDM,SOFTW-SYSLIB
$      SYSOUT  SYOU1,SYOU2
$      SAVE    LUMP
$      ACCOUNT RMV,1T5,IDS,BUFSIZ/400
$      ACCBUF  ,,36
***EOF

$PATCH
***EOF

$LOAD
   .
   .   OBJECT DECKS
   .
***EOF
***EOF
```

Note 1: If Time-Sharing FORTRAN is to be run, its files must be specified on a $ PFILES card.

Note 2: The $ ACCBUF card shows that:
A. All record types will be buffered.
B. All record type dispositions can be changed by the operator.

EXHIBIT OPR070-3

MPC CONTROLLER MAINTENANCE PANEL

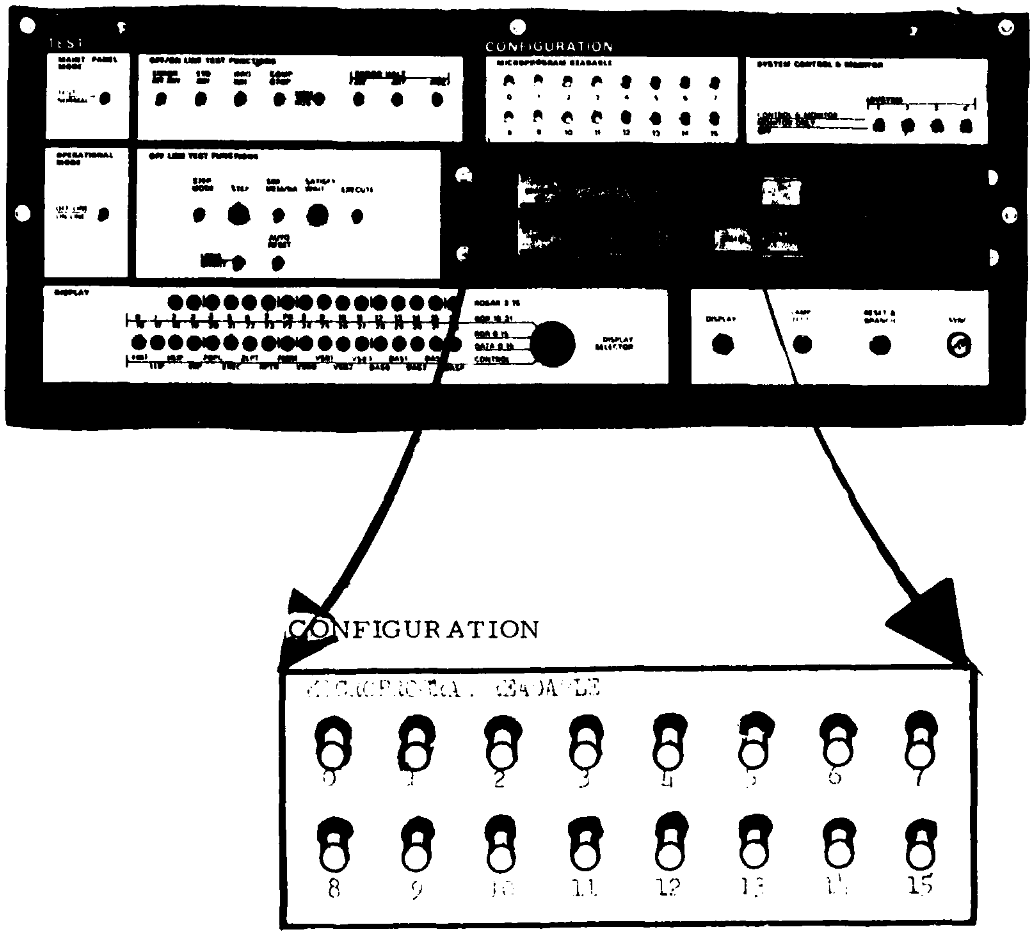
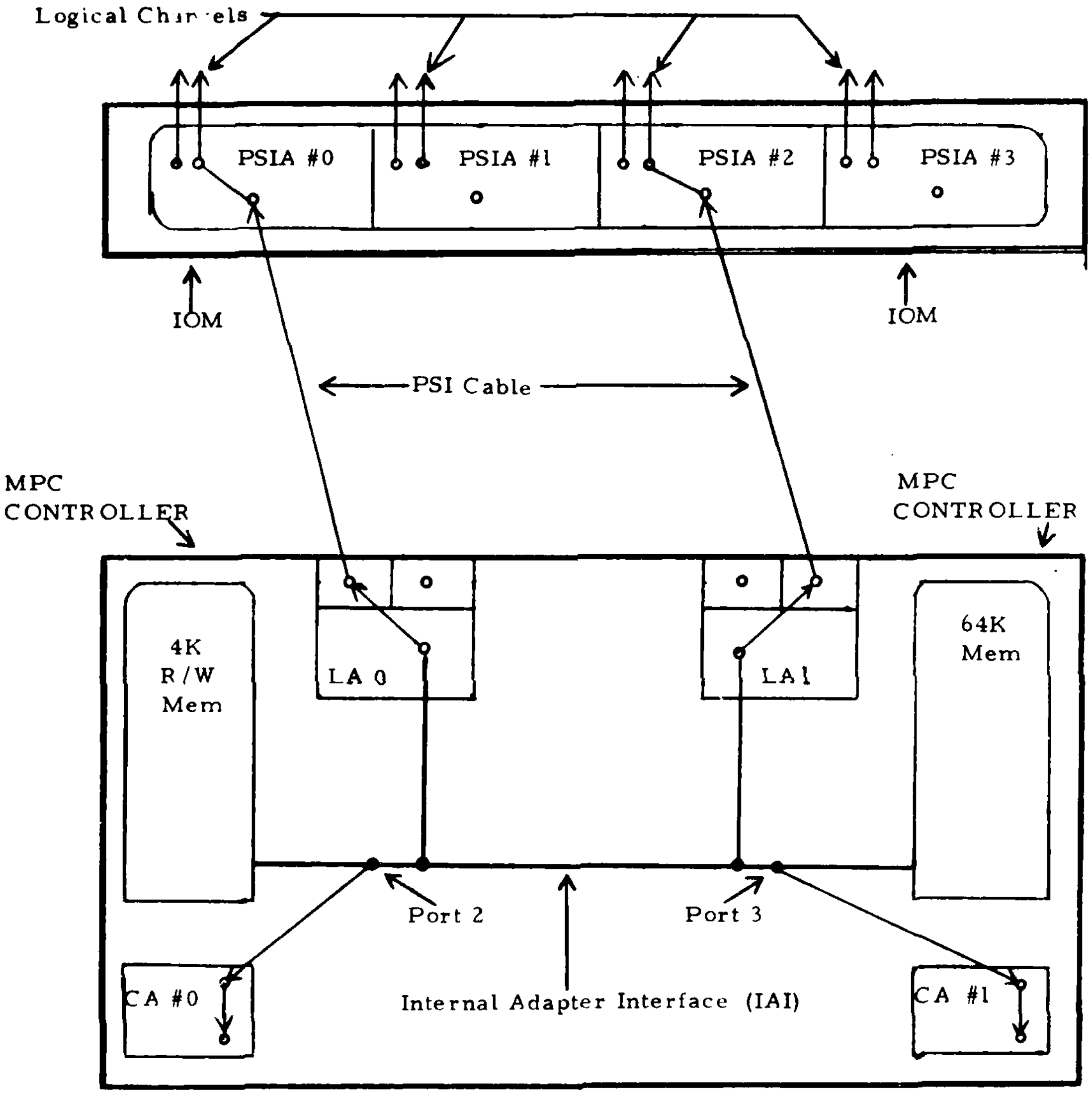


EXHIBIT OPR070-4

MPC CONTROLLER - CARD READER SUBSYSTEM



Note 1: Link Adapter (LA) - Connects PSI to MPC controller.

Note 2: Internal Adapter Interface (IAI) - Handles all data, addresses and control informaticn.

Note 3: Controller Adapter - Connects the MPC to the device.

EXHIBIT OPR070-5

OBJECTIVES

When you have completed the exercise to this exhibit, you will be able to operate and claim peripheral equipment.

PROCEDURES

Accomplish each exercise to this exhibit as directed by your training monitor. Use any available reference material to accomplish these exercises. Each exercise must be accomplished in accordance with local standards and procedures. You will be spot-checked by your training monitor during your performance of the exercises.

EQUIPMENT

Honeywell Series 6000 Computer System.

EXERCISE 1

Perform operation of the card reader by correctly accomplishing the following tasks:

1. Power ON.
2. Load a job deck.
3. Remove a card jam.
4. Clear a Read Alert condition.
5. Clean the card reader.

EXERCISE 2

Perform operation of the card punch by correctly accomplishing the following tasks:

1. Power ON.
2. Load blank cards.
3. Clear a card jam.
4. Clean the card punch.

EXERCISE 3

Perform operation of the disk pack drives by correctly completing the following tasks:

1. Power ON the controller and each disk pack drive.
2. Mount a disk pack.
3. Dismount a disk pack.
4. Clean a disk pack.

EXHIBIT OPR080-1

EXERCISE 4

Perform operation of the magnetic tape subsystem by correc:ly completing the following tasks:

1. Power ON the controller and each tape unit.
2. Mount and load a tape.
3. Unload and dismount a tape.
4. Clean a tape unit.

EXERCISE 5

Perform operation of the printer by correctly accomplishing the following tasks:

1. Power ON.
2. Load and adjust forms.
3. Install a VFU loop.
4. Install a printer ribbon.
5. Clean a printer.

ABBREVIATIONS AND VARIABLES

STANDARD ABBREVIATIONS

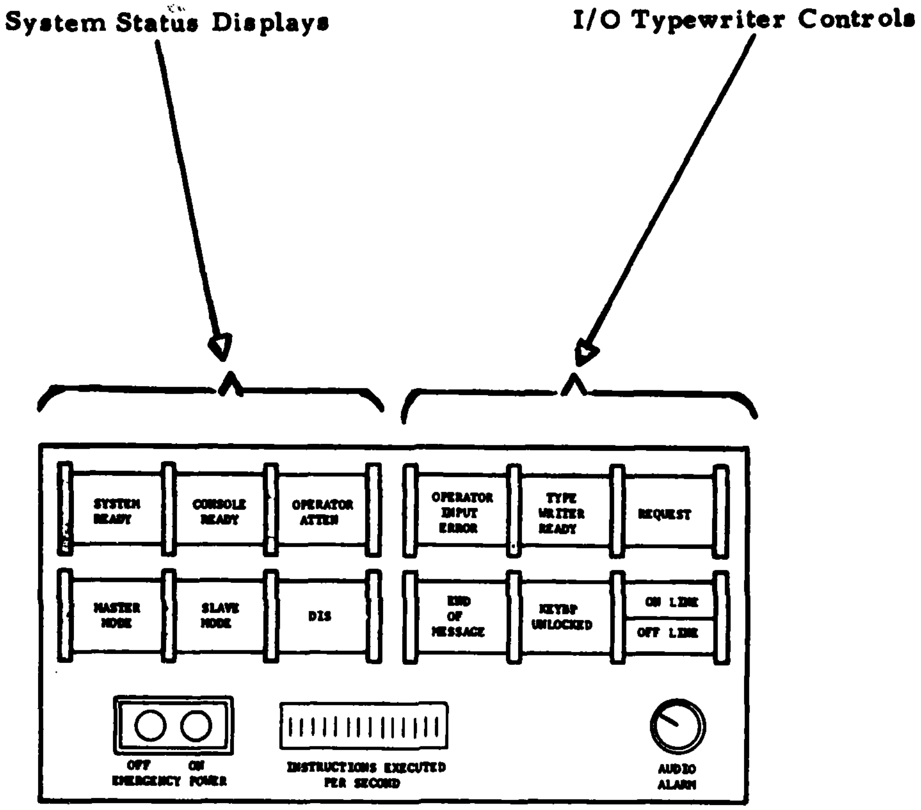
Abbreviation	Meaning
ATT	Attention Major Status
BSY	Channel Busy Major Status
CP	Card Punch
CR	Card Reader
DLT	Deleted
DMT	Dismount
DP	Disk Pack Subsystem
DS	Disk Subsystem
FYI	For Your Information
LRT	Data Alert Major Status
MNT	Mount
MT	Magnetic Tape Subsystem
PN	Card Punch
PR	Printer Subsystem
PT	Perforated Tape Subsystem
PUB	Peripheral Unit Buffer
RDY	Ready
SCC	System Control Center
TRAP	Transaction Processing Application Program
TPE	Transaction Processing Executive
TSS	Time Sharing System
TY	Console

VARIABLES

Variable	Meaning
aa	Activity Number
cc	Channel (PUB) Number
dd	Device Number
dd	Day
fc	File Code
i	IOM Number
id	Identification
mm	Month
pp	Program Number
rrrr or rrrrr	Reel Number
ss	Sequence Number
sssss	Job Sequence Number (SNUMB)
Sid	Station Identification
tt:tt	Time in Hours and Minutes
tt.ttt	Time in Hours and Thousandths of an Hour
ttt	Time in Thousandths of an Hour
tt,t.tt, or tt.tt	Time Hours and Hundredths of an Hour
Uuu	Job Urgency
xyyzz	Alternate iccdd
yy	Year
355-n	DATANET 355/6600 Processor Number (0-3)

EXHIBIT OPR090-1

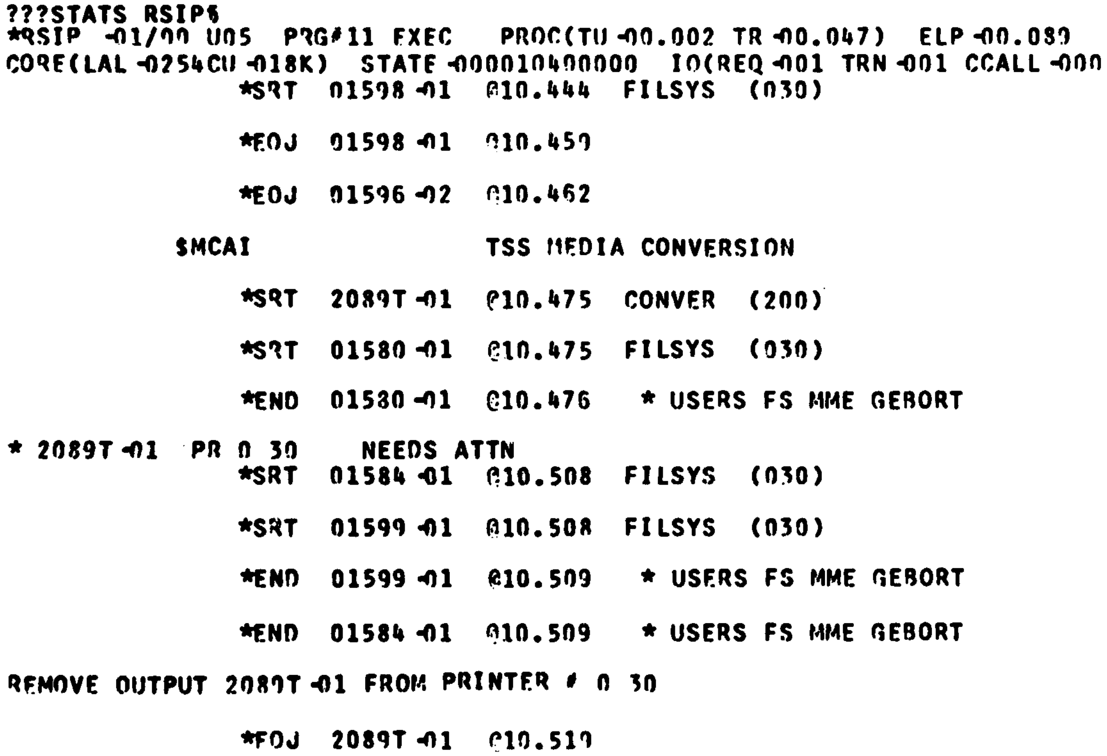
SYSTEM CONSOLE



CONSOLE OPERATOR'S PANEL

EXHIBIT OPR090-2

SYSTEM CONSOLE LISTING



```
???STATS RSIP$
*RSIP =01/00 U05  PRG#11 EXEC   PROC(TU=00.002 TR=00.047)  ELP=00.080
CORE(LAL=0254CU=018K)  STATE=000010400000  IO(REQ=001 TRN=001 CCALL=000
                   *SRT  01598=01  @10.444  FILSYS  (030)

                   *EOJ  01598=01  @10.450

                   *EOJ  01596=02  @10.462

            $MCAI                  TSS MEDIA CONVERSION

                   *SRT  2089T=01  @10.475  CONVER  (200)

                   *SRT  01580=01  @10.475  FILSYS  (030)

                   *END  01580=01  @10.476   * USERS FS MME GEBORT

* 2089T=01   PR 0 30     NEEDS ATTN
                   *SRT  01584=01  @10.508  FILSYS  (030)

                   *SRT  01599=01  @10.508  FILSYS  (030)

                   *END  01599=01  @10.509   * USERS FS MME GEBORT

                   *END  01584=01  @10.509   * USERS FS MME GEBORT

REMOVE OUTPUT 2089T=01 FROM PRINTER # 0 30

                   *EOJ  2089T=01  @10.510
```

EXHIBIT OPR090-3

CONSOLE LISTING-INPUT VERBS

```
???LINES6

CSS 0 IOM 0 CH   17
       IDL   AB
       DAC   TI
       TSS   CN   CO   CM   TK   TJ   6370   TZ   C5


???LIST NAME6
   NAMED DEVICES
ST1   0 08 01        DS1   0 08 01        DS2   0 08 02        DS3   0 08 03
DS4   0 08 04        DS5   0 08 05        DS6   0 08 06        CR1   0 12 00
1T0   0 15 00        1T1   0 15 01        1T2   0 15 02        1T3   0 15 03
PR1   0 30 00        TY1   0 31           TY2   0 31           TY3   0 31
TY4   0 31
           *RDY   CR   0 12 00


???LIST SCHED6
MAX JOBS=35 MIN=10
.EXPRS RUN 20/02 CAT 0390/0000
.HOLD  RUN 00/00 CAT 0390/0000
.NORM  RUN 20/00 CAT 0390/0000
.TASK  RUN 10/00 CAT 0000/0000
.TRANS RUN 10/00 CAT 0000/0000

           *RDY   CR   0 12 00

???LIST SCHED 006
MAX JOBS=35 MIN=10
.EXPRS RUN 20/02 CAT 0390/0000

           *RDY   CR   0 12 00


???LIST RLSED6
RELEASED DEVICES
     0 08 05          0   13U           0   16U           0   18U
 0   19U              0   20U           0   21U           0   22U
 0   23U              0   24U           0   25U           0   26U
 0   27U              0   28U           0   29U
```

EXHIBIT OPR090-4

CONSOLE LISTING

```
???PSTATS 3 PSTATS 0306
                    *END  01697-01  @14.535   * USERS FS MME GEBORT

*030    PRT201 @14.540 #CON=01280 #ERR=0 E/R=0/64 -ASND/$$SYOT
 PR1


???LSTCR6
TOTAL  192K GCOS 36K MXSLV 156K

   036K-079K TSS    3+40K
   079K-090K $RTIN 2+09K
   106K-112K $LOGN 1+05K
   112K-126K $PALC 1+13K
   127K-132K VIDEO 1+04K
   132K-135K HEALS 1+02K
   135K-153K $SYOT 1+17K
   153K-185K 6386T 2+30K

@090K 016K @126K 001K @185K 007K


  ???LSTAL6
                *GET P#14 S#2708T TAPE#00908 RING-IN/9TRK
                              TAPE#00909 RING-IN/9TRK



  WAITING MEDIA
  2695T-01 U05 IDTI
  SWAPPED OUT
  HEALS-01 U05
  IN EXECUTION
  TSS   -01 U61       VIDEO-01 U60
  IN LIMBO
  2685T-01 U05 IDTI 2708T-01 U05 IDCO
                * PLEASE, FIX SYSOUT DEVICE CP 0 14
```

EXHIBIT OPR090-5

CONSOLE LISTING

```
???LSTWT6
                    *SRT  01056-01  @13.555  FILSYS  (030)

                    *END  01056-01  @13.555   * USERS FS MME GEBORT

*SYOT SPACE @13.556(TOTAL=0120 AVAIL=0377 #ERRS=0
 *WAITING TO PRINT
 01056
 *WAITING TO PUNCH
 2694T




???LSTRE6
NO SPECIAL REQUESTS TO GEOT



???CANCL PURGE ALL6

???CZNCL PURGE6

WHAT?
CANCL PURGE6
            * 0-15-02 DMT PURG


???LSTRE6

            SPECIAL REQUESTS TO GEOT
PURGE ALL           CANCEL
```

EXHIBIT OPR090-6

OBJECTIVES

When you have accomplished the exercise in this exhibit, you will be able to:

1. Identify the three types of system console messages.
2. Communicate with the H6000 computer system via the system console.
3. Use the Console Typewriter Messages reference manual to determine the appropriate operator action for system console messages.

PROCEDURES

Accomplish the exercise as directed by your training monitor. Use any available reference materials, including the procedures presented in the course up to this point, to accomplish the exercise.

EQUIPMENT

Honeywell Series H6000 Computer System

EXERCISE

Perform operation of the system console for a period of four hours. During this time, perform the following task:

1. Power ON.
2. Input one console verb.
3. Clear a locked keyboard condition.
4. Use the OPERATOR INPUT ERROR pushbutton to reaccomplish the input of an incorrect verb/message.

Part 1

STARTUP AND DUMP

1. What console command re-initializes the system when GCOS is in core memory?

2. What three responses are made to the following message?

   *BOOT SOURCE: CARD, AUTO?

3. What response to the following message causes the jobs in the system to be restarted?

   *RESTART?

4. What response to the *EDIT? message processes each card of this section individually?

5. What action is taken to correct the condition specified by the following message?

   *EDIT CARD BELOW SPECIFIES FILE IN UNKNOWN FORMAT

   $FILDEF ST1,GCOS,WMIXUSE,300/0,SYE,1T3

6. What must be done to correct the condition indicated by the following message?

   *FAULT VECTOR SWITCHES SET TOO LOW

7. When response causes the options of the $ Answer card to be performed when the following message is outputted?

   *CHANGE?

8. What tasks must be performed to rectify the condition indicated?

   *$GCOSFILE CARD IS MISSING

9. Enter the response to the following message that bootloads the 181 Disk controller.

   *MSS181 ON Ø Ø8 BOOTLOAD? (SYS ID NAME XXXXXX)

EXHIBIT OPR100-1

10. What response to the *SCF Continuation? message tells the system to restart the statistical collection file?

11. What response to the following message causes the jobs in the system scheduler to be restarted?

*SYSTEM SCHEDULER CLEAR?

12. Change the following date in the computer to July 15, 1975.

*DATE 041875

13. What response to the TIME? question sets the system clock to midnight?

14. What response to the *INITIALIZE? console message causes startup to bypass mass storage initialization?

15. What verb is used to cause the system to perform a master mode dump?

16. What response causes the dump program to use the options specified in the $ Answer card when the following message is displayed on the console?

***DUMP (O)PTIONS OR (D)EFAULT

17. What action should be taken after the following message is displayed?

*END,RESTART TABLES SAVED

18. What response to the following message produces a hard-copy of a master-mode dump?

*PRINT OR TAPE/REEL#, DENSITY

Part 2

SYSTEM SCHEDULER

1. What input message is used to inform the System Scheduler to delete program 56989 from its catalog?

2. Your system has just been rebooted with system scheduler restart. Notify the system scheduler to restart normal scheduling of all jobs.

3. What input message displays a list of all the jobs known to the system scheduler in .NORM (USER) class? The .NORM (USER) class can also be represented by the numeric value 01.

4. What input message is used to notify the system "to prepare" to purge SNUMB 2468T from the system scheduler?

5. When the following output message is generated on the console, what response must you enter "to execute" the previously given system scheduler purge?

```
*JPURGE 2468T READY
```

Part 3

INPUTS - EXCEPTION PROCESSOR

1. What are the "Explicit" exception processor messages that can be inputted to a user program?

2. What are the "Implicit" exception processor messages that can be inputted to a user program?

3. What are the "Implicit" exception processor messages that apply to I/O Devices?

4. What does the following message indicate about the removable Disk Pack?

```
*ATT I ØØXX DP Ø Ø8 Ø1 1895T R?
```

5. What is wrong with the console device when the following message is outputted?

```
*ATT I ØØXX TY Ø 31 32222 RM?
```

6. What does the following message indicate?

```
*ATT I Ø1XX CR Ø 16 1621T RDY?
```

7. What action must you take to correct the attention condition given below?

```
*ATT I Ø1XX CR Ø 16 1621T
```

8. What must you do to correct the following attention condition?

```
*ATT I Ø1XX PR Ø 14 39742 RU?
```

9. When you get the following message, what is wrong with the tape subsystem?

```
*ATT I Ø2XX MT Ø 12 Ø1 9522T R?
```

10. What is wrong with the following indicated I/O Device?

```
*ATT T 2ØXX CP Ø 15 Ø ØØ 79543 B?
```

11. What has occurred at the card reader?

```
*ATT T 5ØXX CR 1 13 ØØ 9215T U?
```

12. What is the meaning of the following output message?

```
*FYI (25/64) THRESHOLD ON Ø 12 Ø1
```

EXHIBIT OPR100-1 (Cont)

Part 4

TIME-SHARING SYSTEM

1. Notify all Remote Terminals Users that TSS will go off the air at 1200 hrs.

2. Check the status of TSS.

3. Inform TSS that you desire to send a message to all users.

4. Display the TSS message on the console that's sent to all users at log-on time.

5. Cancel the previously inputted TSS request.

6. Set the maximum size of TSS to 40K words.

Part 5

CORE ALLOCATOR (GPOP)

1. Enter the command to execute SNUMB 53278 from the system console.

2. What command must be used to check the status of SNUMB 4792T?

3. What device must be checked when the following message is shown?

   *CSS-Ø IS DOWN PLEASE INVESTIGATE

4. Limit the size of a slave program so that it will use 32768 words of core memory.

5. Change the tape Sieve Limit so that only three (3) units will be allocated for use.

6. What console command causes Datanets 355-Ø and 355-1 to accept no further calls from their respective remote terminals?

7. Determine the areas of memory that have been released and are being tested by TOLTS.

8. What console verb causes the following job statistics to be displayed on the console?

   a. Number of Jobs in execution.
   b. Number of Jobs waiting for SYSOUT.
   c. Number of Jobs to be executed.
   d. Totals for Jobs cataloged by System Scheduler.

9. List the jobs that have a status that pertains to the peripheral allocator.

10. Look at Exhibit OPR060-6, IOM-0, Pub-12, Tape Unit 1T2. What console command is entered to exchange tape unit 2T4 for 1T2 when unit 1T2 is not working properly?

11. Determine the unavailable LLINKS on disk pack DP1, shown on Exhibit OPR060-6.

12. Start the IMCV program that uses a 9-track tape located on u 1T3.

13. What response must you key in on the console to cause the IMCV Program to omit SNUMB 13578 from the above tape when the following output message is generated and the tape has been mounted?

```
*TAPES-IMCV MNT Ø-12-Ø4
*OPTIONS
```

14. Determine the status of Channel 8 on IOM-Ø.

15. List the remote jobs known to the system.

16. Display a mass storage resource list on the system console.

17. What console command is entered that causes allocation of Disk Pack DP5, IOM-Ø, Pub-8, to be moved to IOM-1, Pub-8, unit 5?

18. Display the present dispositions for all SCF record types on the system console.

19. Kill Job Sequence Number 1579T.

20. What input verb was keyed in to cause the following message to be displayed on the console?

```
*NAME TABLE FULL
```

21. Remove the dedication from the PRT300 printer shown on Exhibit OPR060-6, IOM-0, Pub 14.

22. Set the URGENCY for SNUMB 6478T to 50.

23. Display a list on the console of all input messages known to the system.

24. What console command is entered to release the following job for allocation when the specified media is available?

```
*GET P#16 S#2435T TAPE#00612 RING-IN/9TRK
```

EXHIBIT OPR100-1 (Cont)

25. Determine if there are any areas of memory being tested by the Total On-Line Test System (TOLTS).

26. ABORT - SNUMB 7788T and produce a dump regardless of user specified options in the $ EXECUTE card.

27. Display a listing of the system's tape configuration on the console.

28. ABORT SNUMB 82975 and do not produce a program dump.

29. Set the time in the computer to 1400 hours.

30. What console commands turns on the system trace function?

31. Display a listing on the console of the system's current hardware configuration.

32. Remove the name from the punch shown on Exhibit OPR060-6, IOM-1, Pub-12.

33. What input verb is used to display the contents of memory cell 777060 on the system console?

34. What type of privilege is granted to the job shown in the following message when the run verb is entered?

*S#355LD ASKS PRIVITY, ID=OPNSVTIL..RUN?

Part 6

SYSTEM OUTPUT CONVERSION (SYSOUT)

Enter the message that will accomplish each of the actions listed.

1. Terminate the current SYSOUT collector tape as soon as possible.

2. Eliminate all of the output for Job Sequence Number 7632T.

3. Repunch the cards for SNUMB 3479T.

4. Reprint the reports for SNUMB 22222.

5. Redirect all SYSOUT to remote printer AA.

6. Place SYSOUT information on magnetic tape, and also produce a hard copy of this data.

7. You've just been notified that there's a storm in the immediate area of your site. You want to save all system output. Have all SYSOUT copied on magnetic tape as well as hard copy production.

8. Remove SNUMB 64325 when it is in SYSOUT.

9. Have the printer output for SNUMB 1246T printed ahead of other SYSOUT data.

10. Cancel the previous copy request to SYSOUT that affects all output to remote AA.

11. Cancel the redirect to remote AB.

CONSOLE MAINTENANCE PANEL

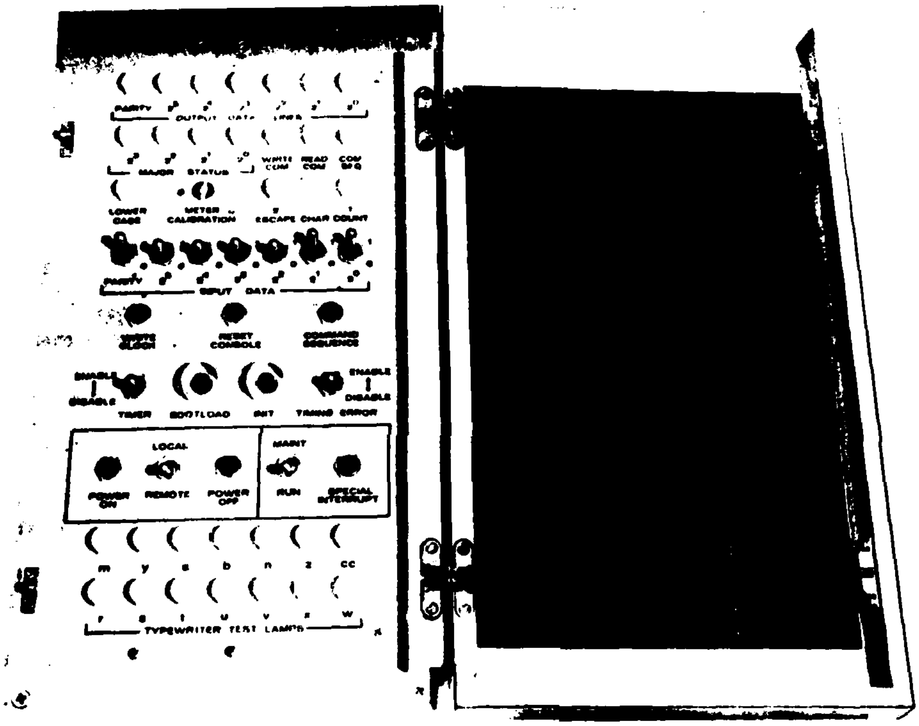


EXHIBIT OPR100-2

BASIC FLOW - PROCESSOR MAINTENANCE PANEL PROCEDURE TO EXECUTE A MASTER- MODE DUMP

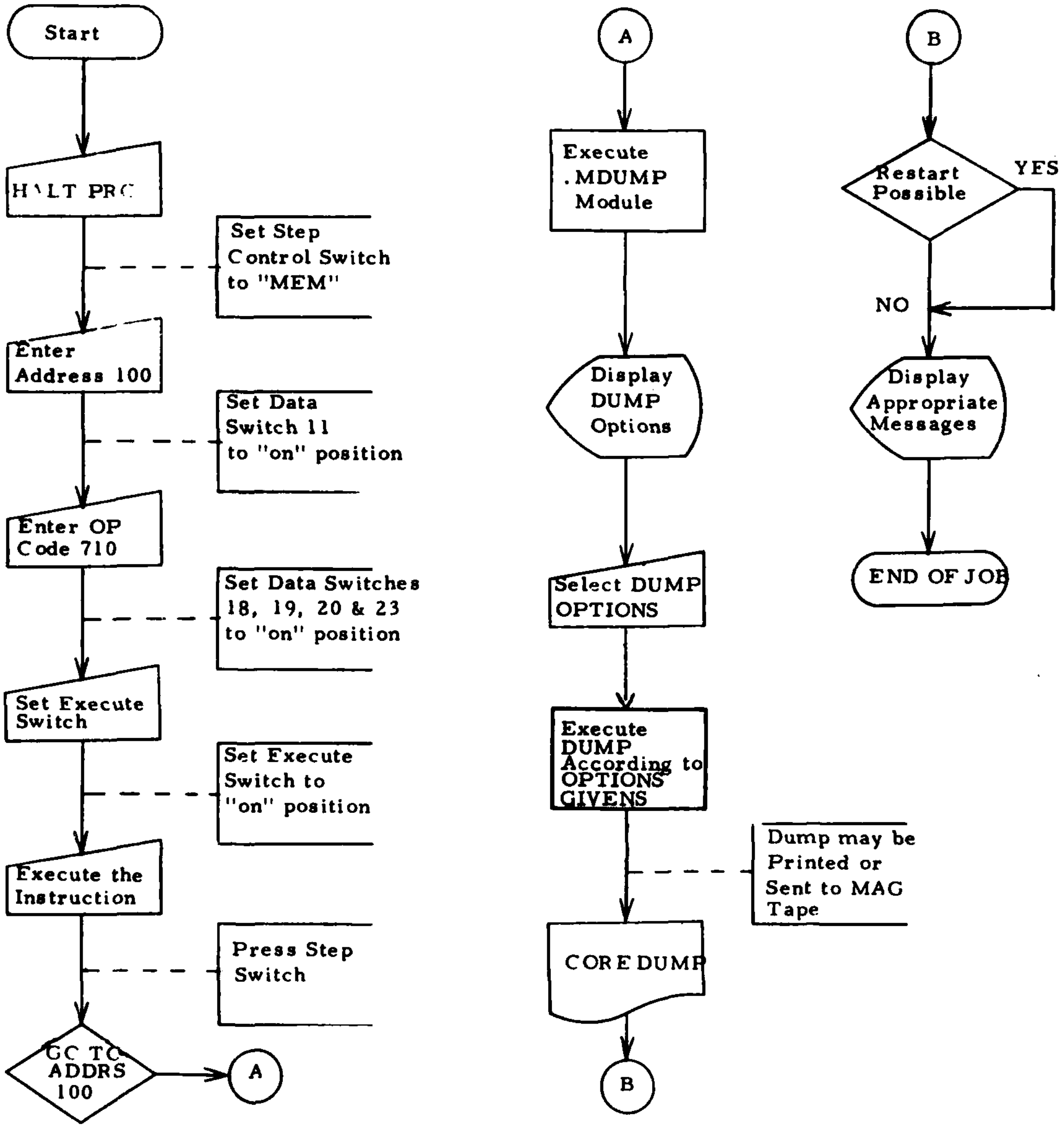

Start
H^LT PRC
Set Step Control Switch to "MEM"
Enter Address 100
Set Data Switch 11 to "on" position
Enter OP Code 710
Set Data Switches 18, 19, 20 & 23 to "on" position
Set Execute Switch
Set Execute Switch to "on" position
Execute the Instruction
Press Step Switch
GC TC ADDRS 100
A
A
Execute .MDUMP Module
Display DUMP Options
Select DUMP OPTIONS
Execute DUMP According to OPTIONS GIVENS
Dump may be Printed or Sent to MAG Tape
CORE DUMP
B
B
Restart Possible
YES
NO
Display Appropriate Messages
END OF JOB

EXHIBIT OPR100-3

PROCESSOR MAINTENANCE PANEL

(UPPER HALF)

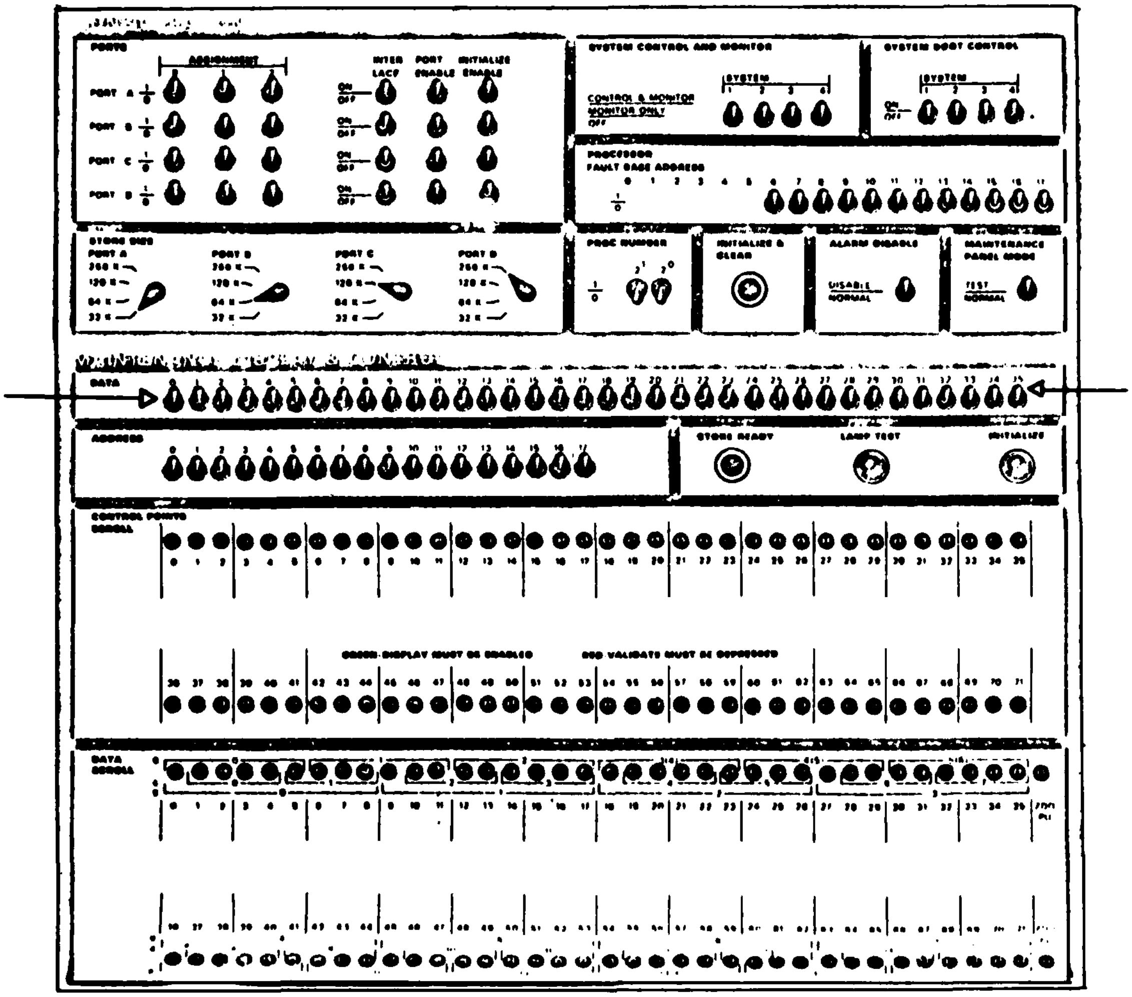


EXHIBIT OPR100-4

PROCESSOR MAINTENANCE PANEL

(LOWER HALF)

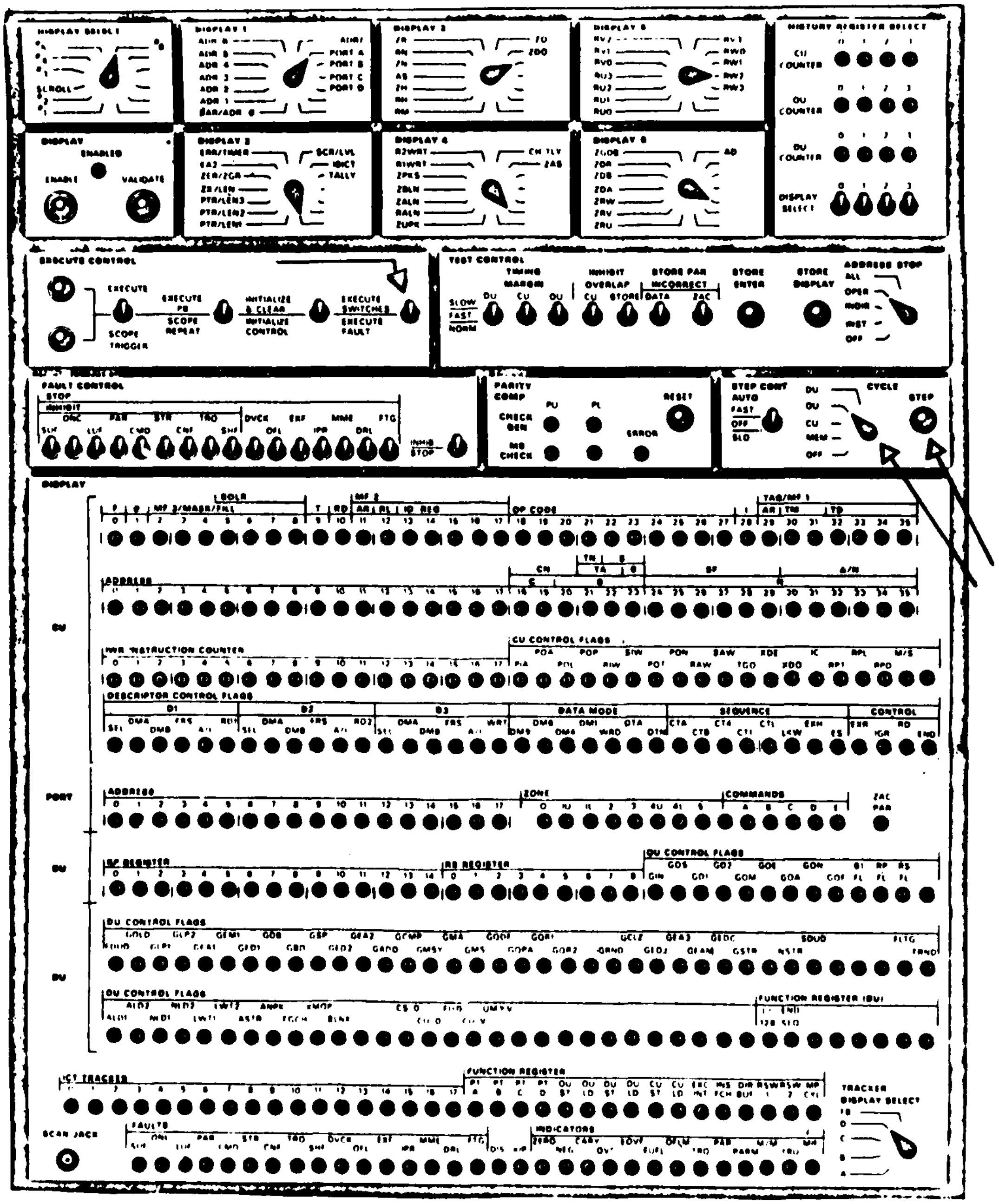
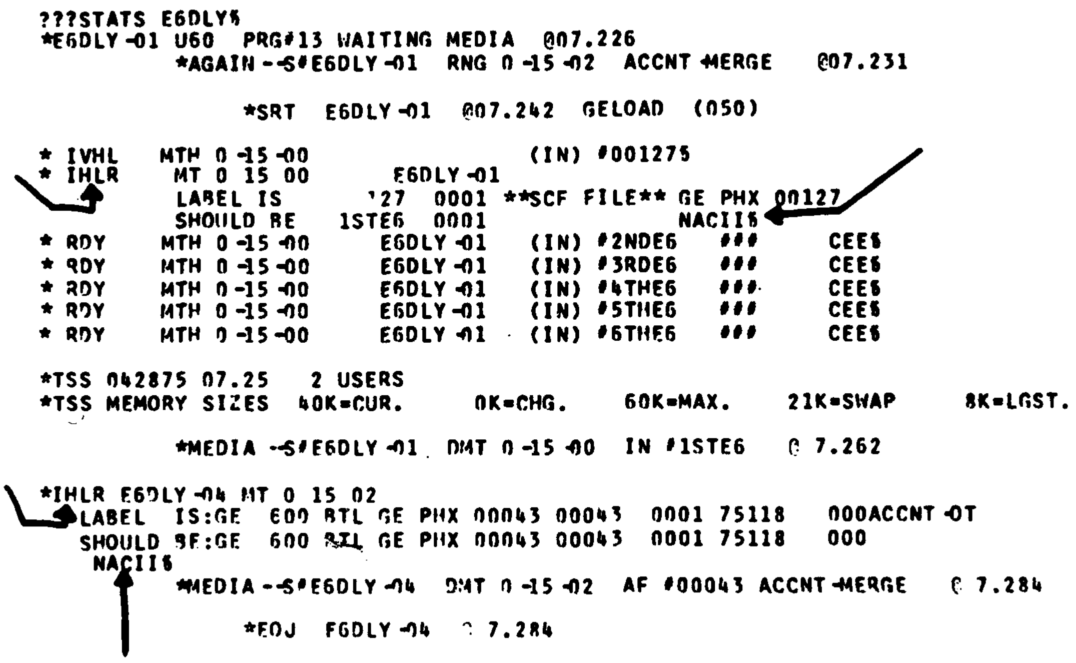


EXHIBIT OPR100-5

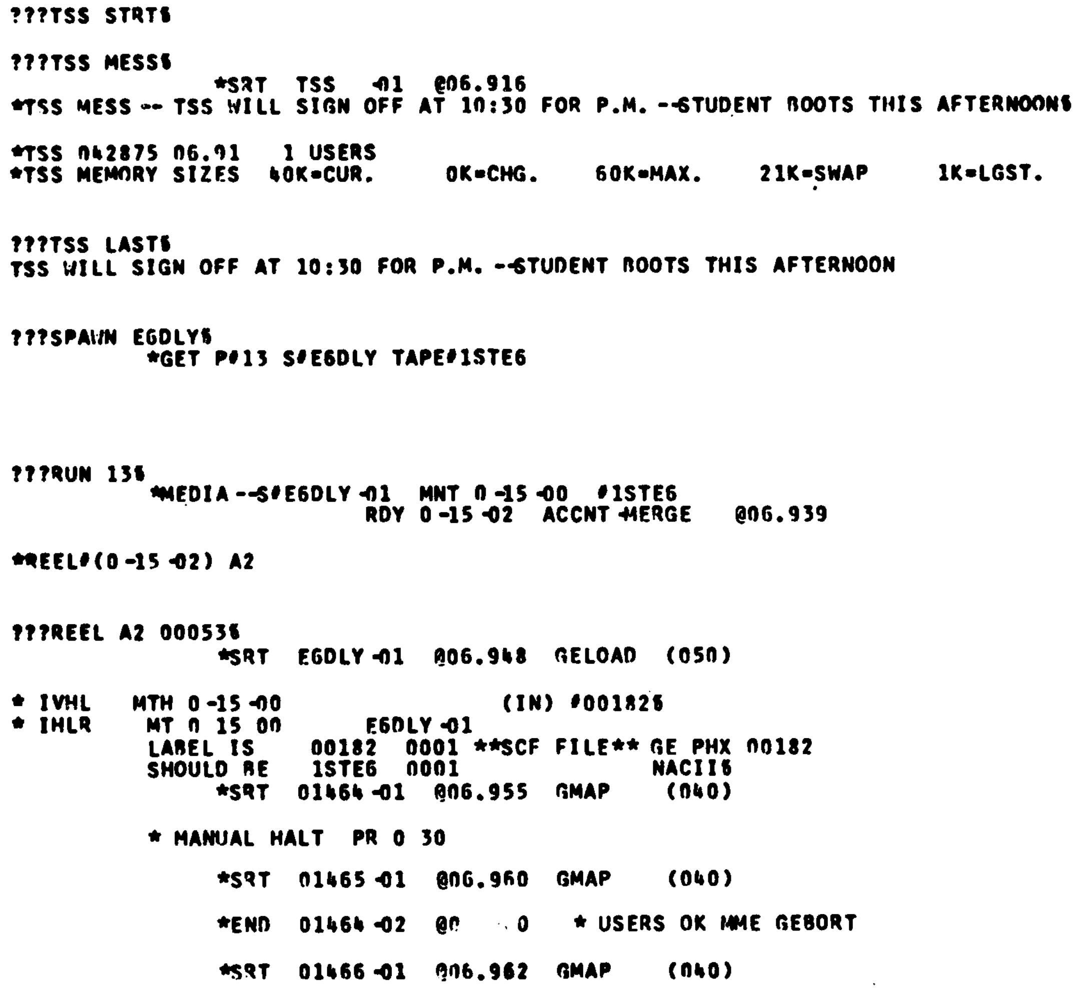
CONSOLE LISTING



```
???STATS E6DLY$
*E6DLY-01 U60  PRG#13 WAITING MEDIA  @07.226
          *AGAIN--S#E6DLY-01  RNG 0-15-02  ACCNT-MERGE   @07.231

               *SRT  E6DLY-01  @07.242  GELOAD  (050)

* IVHL    MTH 0-15-00                   (IN) #001275
* IHLR     MT 0 15 00        E6DLY-01
          LABEL IS        '27  0001 **SCF FILE** GE PHX 00127
          SHOULD BE   1STE6  0001                NACII$
* RDY     MTH 0-15-00       E6DLY-01   (IN) #2NDE6   ###      CEE$
* 9DY     MTH 0-15-00       E6DLY-01   (IN) #3RDE6   ###      CEE$
* 3DY     MTH 0-15-00       E6DLY-01   (IN) #4THE6   ###      CEE$
* RDY     MTH 0-15-00       E6DLY-01   (IN) #5THE6   ###      CEE$
* RDY     MTH 0-15-00       E6DLY-01   (IN) #6THE6   ###      CEE$

*TSS 042875 07.25   2 USERS
*TSS MEMORY SIZES  40K=CUR.     0K=CHG.    60K=MAX.    21K=SWAP     8K=LGST.

          *MEDIA --S#E6DLY-01  DMT 0-15-00  IN #1STE6   @ 7.262

*IHLR E6DLY-04 MT 0 15 02
   LABEL  IS:GE  600 BTL GE PHX 00043 00043  0001 75118   000ACCNT-OT
   SHOULD 9E:GE  600 BTL GE PHX 00043 00043  0001 75118   000
    NACII$
          *MEDIA--S#E6DLY-04  DMT 0-15-02  AF #00043 ACCNT-MERGE   @ 7.284

               *EOJ  FGDLY-04  @ 7.284
```

EXHIBIT OPR100-6

CONSOLE LISTING



```
???TSS STRT$

???TSS MESS$
                    *SRT  TSS  -01  @06.916
*TSS MESS -- TSS WILL SIGN OFF AT 10:30 FOR P.M. --STUDENT ROOTS THIS AFTERNOON$

*TSS 042875 06.91   1 USERS
*TSS MEMORY SIZES  40K=CUR.    0K=CHG.    60K=MAX.    21K=SWAP     1K=LGST.


???TSS LAST$
TSS WILL SIGN OFF AT 10:30 FOR P.M. --STUDENT ROOTS THIS AFTERNOON


???SPAWN EGDLY$
             *GET P#13 S#E6DLY TAPE#1STE6



???RUN 13$
             *MEDIA --S#E6DLY -01  MNT 0 -15 -00  #1STE6
                                 RDY 0 -15 -02  ACCNT -MERGE   @06.939

*REEL#(0 -15 -02) A2


???REEL A2 00053$
                   *SRT  EGDLY -01  @06.948  GELOAD  (050)

* IVHL   MTH 0 -15 -00                    (IN) #00182$
* IHLR    MT 0 15 00        E6DLY -01
          LABEL IS    00182  0001 **SCF FILE** GE PHX 00182
          SHOULD BE   1STE6  0001                NACII$
                *SRT  01464 -01  @06.955  GMAP    (040)

          * MANUAL HALT  PR 0 30

                *SRT  01465 -01  @06.960  GMAP    (040)

                *END  01464 -02  @0     0    * USERS OK MME GEBORT

                *SRT  01466 -01  @06.962  GMAP    (040)
```

EXHIBIT OPR100-7

SYSTEM REBOOT SEQUENCE - MESSAGES

```
*END, RESTART TABLES SAVED

*BOOT SOURCE:  CARD, AUTO? AUTOb̸
*CONTROL CHG AT CONFIG,INITIALIZE,EDIT,FILES,PATCH OR LOAD?b̸
*CHANGE? YESb̸
*RESTART? YESb̸
*SCF CONTINUATION? YESb̸
*DATE  000000 ? 170875b̸
*TIME 00.000?  11:10b̸
*INITIALIZE?b̸
*EDIT?b̸
*SYSTEM SCHEDULER CLEAR?NOb̸

*  TAPE 01 ON 0 12 RELEASED

*  UNIT 04 DSPK 0 08 RELEASED

*  UNIT 05 DSPK 0 08 RELEASED

*SYSTEM SCHDLR JOBS TO RE-INPUT
1786T-1  2095T-B
*CANNOT READ ALL OF SYS SCHEDLR FILE

*JOBS THAT CANNOT BE RESTARTED, RE-INPUT

NONE

*JOBS THAT ARE FINISHED, ARE IN SYSOUT

NONE

*JOBS THAT ARE BEING RESTARTED

PFILE PPRMP PPRNT

*DATE 170875     TIME  11.200

          *SYSOUT FOUND:  0120 BLINKS, 3 PRINTERS, 2 PUNCHES, 22 REMOTES

          *SYSOUT FOUND:  A BACKDOOR FILE, TOO

???JRUN RESTb̸

???LSTALb̸
```

NOTE 1: The underlined entires are the operator inputs.

EXHIBIT OPR100-8

OBJECTIVES

When you have completed the exercises in this exhibit, you will be able to:

1. Monitor the entire operation of the H6000 system via the system console, including:

   a. System startup.
   b. Remote processing systems.
   c. Normal batch processing requirements.
   d. Dump procedures.

2. Direct team members to take action on all peripherals.

3. Determine possible cause of machine malfunctions and initiate corrective action.

PROCEDURES

Accomplish each exercise in this exhibit as directed by your training monitor. Use any available reference material to accomplish these exercises. Each exercise must be accomplished in accordance with local standards and procedures. You may obtain assistance from your training monitor on the more difficult parts during your performance of these exercises.

EQUIPMENT

Honeywell Series 6000 Computer System

EXERCISE 1

Execute a WARM BOOT by performing the following tasks:

1. Initiate the WARM BOOT from the system console.
2. Answer the bootload questions correctly.

EXERCISE 2

Perform a COLD BOOT by performing the following tasks:

1. Insure that all switches on the main frame modules are properly set.
2. Insure that no trouble conditions exist on the main frame modules.
3. Enter the Startup Deck from tape or cards (optional).
4. Initiate the COLD BOOT from the IOM or console maintenance panel (optional).
5. Answer the bootload questions correctly.

EXERCISE 3

Initiate and monitor remote processing subsystems by performing the following tasks:

1. Bootload the DATANET 355 FNP.
2. Open all communication lines.
3. "nlock a remote.
4. Start the time sharing subsystem.
5. Terminate the time sharing subsystem.
6. Close all communication lines.

EXERCISE 4

Monitor the operation of the H6000 System from the system console while performing as a team member for a period of 8 hours. During this period, you should perform the following tasks:

1. Set proper sieve limits as directed.
2. Monitor the system by properly entering verbs and responding to output messages.
3. Direct team members to perform peripheral operations as dictated by console messages.

EXERCISE 5

Execute master mode dumps by performing the following tasks:

1. Initiate a master mode dump from the system console.
2. Initiate a master mode dump from the Processor maintenance panel.

CONSOLE VERBS AND MESSAGES

Each of the following items will require a typewriter input or an operator action. If an item requires a typewriter input, print the typewriter input in the correct format in the space provided below the item. If the item requires operator action other than typewriter input, explain the necessary operator action. Use the $CONFIG section of the startup deck, provided in EXHIBIT OPR060-6, to formulate your correct response.

1. *ATT I 0440 CP 0 15 00 04005 RDY?

2. Name the reader connected to IOM-0 "RD-1".

3. Request the current status of the tape unit named 2T3.

4. *ATT I 20XR CR 0 13 00 04005 R?

5. Assign the tape unit named 1T1 to the system.

6. Dedicate the disk unit number 6 connected to IOM-0.

7. Remove the name from the printer connected to IOM-1.

8. Remove S#04005 from SYSOUT.

9. List the system counter totals.

10. Request that all SYSOUT be copied to mag tape.

11. *ATT T10XR MT 0 12 03 04005 R?

12. Request job status information about S#04005.

13. Release the printer connected to IOM-0 from the system.

14. Release 1024 words of core starting at octal location 140.

EXHIBIT OPR110-1

15. *ATT T 10XR CR 1 13 00 04005 RDY?

16. Abort PGM 04005 no dump is required regardless of user option.

17. List the special requests made to the system.

18. *LRT T 20XX PR 0 30 00 04005 R?

19. *MEDIA--S#4005 MNT 0 12 06 #00145 (SNAKE) @029518.

20. Name the third tape unit connected to IOM-1 "2T2".

21. List the systems mass storage resources on the console.

22. Abort job 04005 produce a dump regardless of user option.

23. Undedicate the printer connected to IOM-1.

24. Cancel the COPY ALL request.

25. Reprint all reports for S#04005.

26. Redirect all printed output to remote device BB.

27. Place all SYSOUT output on mag tape.

28. List all the jobs in the system.

29. *ATT I 01XR PR 0 30 00 04005 U?

30. Return 1024 words of core starting at address 140 to active allocation.

31. Abort PGM 04005 and produce a dump at user option.

32. Set the date in the system to 08 DEC 72.

33. Start the IMCV program on tape unit 2 connected to IOM-0.

34. Limit the size of slave programs to 32768 words.

35. List the named devices on the system.

36. Cancel the REDIRECT request from the printer to remote device BB.

37. Repunch all output from S#04005.

38. Limit the size of the time-sharing subsystem to 30K.

39. Connect all communication lines.

40. Display the current number of TS1 users.

41. Initiate the TS1 subsystem.

42. Disconnect all communication lines.

43. Allow no new TS1 users to sign on the system.

44. Terminate the TS1 subsystem.

45. Bootload the DATANET 355: 355LD is a permanent file.

46. Close out the current statistical collection file and open a new one.

47. Print the output of job 04005 before any other output.

48. Check the status of all remote lines.

49. List those jobs that are waiting in the peripheral allocator.

50. Job number 04005 needs to be processed with a higher priority than other jobs.

Insure that you have completed all 50 items. If all 50 items have been completed, please begin lesson OPR110 at this time. To begin lesson OPR 110, type "?GET OPR110 NEW".

EXHIBIT OPR110-1 (Cont)

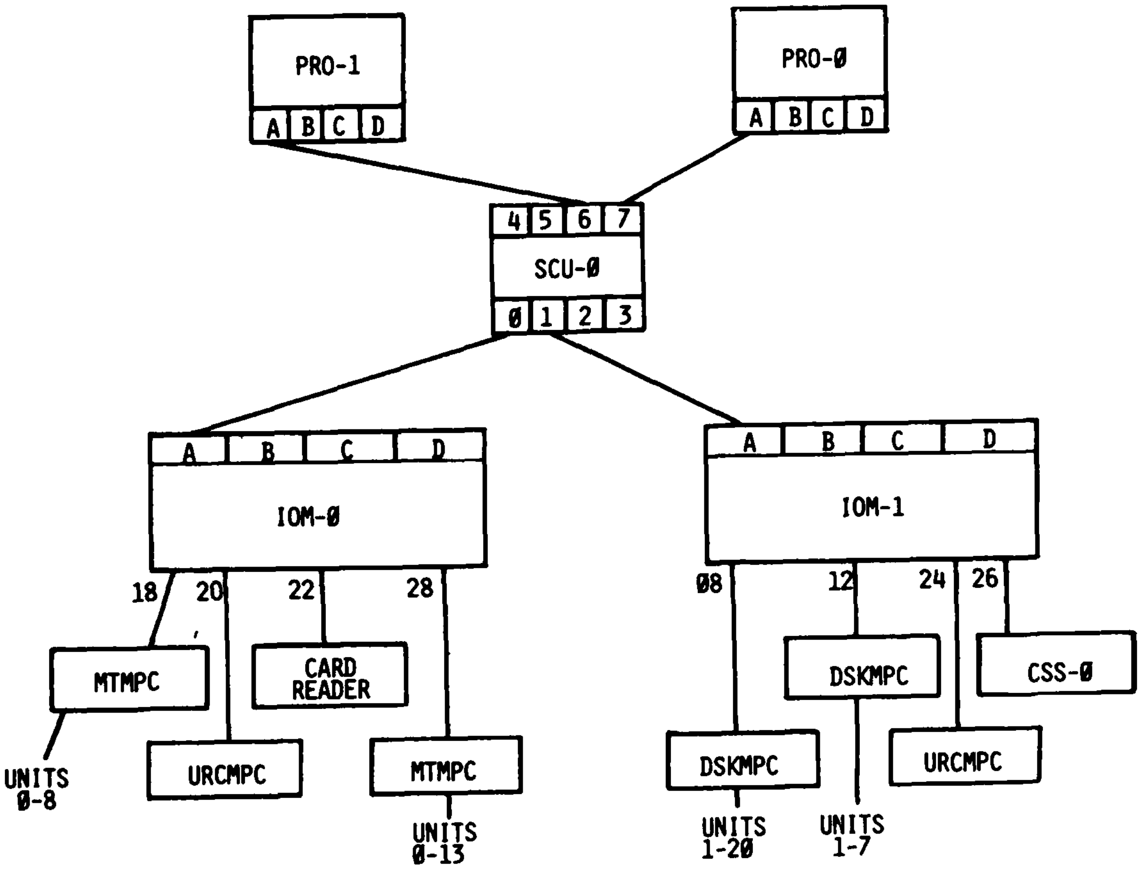


EXHIBIT OPR120-1

The trainee must perform all parts of the exercises listed below with only a spot check of completed work. Each exercise must be performed in accordance with local standards and procedures. The trainee should be allowed to use all available reference material to accomplish each exercise.

EXERCISE 1

Perform configuration of the hardware modules by completing the following tasks:

1. Apply power to each hardware module.
2. Configure each hardware module by setting the appropriate switches on the configuration portion of the maintenance panel of each module.

EXERCISE 2

Perform reconfiguration of the hardware modules by completing the following tasks:

1. Remove a faulty block of memory using the System Controller Unit "Address Offset" switch.
2. Remove a Memory Unit from the system.
3. Remove an Input/Output Multiplexer (IOM) from the system.
4. Remove a Processor from the system.

EXHIBIT TPT010-1

The trainee must perform all parts of the exercise listed below. The trainee should be able to do most parts of the task and may be provided help only on the most difficult parts of the task. This task must be performed in accordance with local standards and procedures. The trainee should be allowed to use all available reference material to complete the exercise.

EXERCISE

Construct and keypunch the $CONFIG Section of the startup deck for your site's Series 6000 Computer System.

EXHIBIT TPT020-1

The trainee must perform all parts of the exercises listed below with only a spot check of the completed work. Each exercise must be performed in accordance with local standards and procedures. The trainee should be allowed to use all available reference material to perform the exercises.

EXERCISE 1

Perform operation of the card reader for a minimum of one hour. During this time, the trainee should complete the following tasks:

1. Power ON.
2. Load a job deck.
3. Remove a card jam.
4. Clear a Read Alert condition.
5. Clean the card reader.

EXERCISE 2

Perform operation of the card punch for a minimum of one hour. During this time, the trainee should complete the following tasks:

1. Power ON.
2. Load blank cards.
3. Clear a card jam.
4. Clean the card punch.

EXERCISE 3

Perform operation of the disk pack drives for a minimum of one hour. During this time, the trainee should complete the following tasks:

1. Power ON controller and each disk pack drive.
2. Mount a disk pack.
3. Dismount a disk pack.
4. Clean a disk pack.

EXERCISE 4

Perform operation of the magnetic tape subsystem for a minimum of two hours. During this time, the trainee should complete the following tasks:

1. Power ON the controller and each tape unit.
2. Mount and load a tape.
3. Unload and dismount a tape.
4. Clean a tape unit.

EXHIBIT TPT030-1

EXERCISE 5

Perform operation of the printer for a minimum of two hours. During this time, the trainee should perform the following tasks:

1. Power ON.
2. Load and adjust forms.
3. Install VFU loop.
4. Install a printer ribbon.
5. Clean the printer.

EXHIBIT TPT030-1 (Cont)

The trainee must be able to perform all parts of the exercise listed below. The trainee may be provided help only on the hardest part of the exercise. Allow the trainee to use all available reference material during the performance of the exercise. The exercise must be performed in accordance with local standards and procedures.

EXERCISE

Perform operation of the system console for a period of four hours. During this training period, the trainee must perform the following tasks:

1. Power ON.
2. Input one console verb.
3. Clear a locked keyboard condition.
4. Use the OPERATOR INPUT ERROR pushbutton to reaccomplish the input of an incorrect verb/message.
5. Reply to an output message.

EXHIBIT TPT040-1

The trainee must be able to perform most parts in each of the exercises listed below. The trainee may be provided help only on the hardest part of each exercise. Allow the trainee to use all available reference material in the performance of each exercise. Each exercise must be performed in accordance with local standards and procedures.

EXERCISE 1

Execute a WARM BOOT by performing the following tasks:

1. Initiate the WARM BOOT from the system console.
2. Answer the bootload questions correctly.

EXERCISE 2

Execute a COLD BOOT by performing the following tasks:

1. Insure that all switches on the main frame modules are properly set.
2. Insure that no trouble conditions exist on the main frame modules.
3. Enter the Startup Deck from tape or cards (optional).
4. Initiate the COLD BOOT from IOM or console maintenance panel (optional).
5. Answer the bootload questions correctly.

EXERCISE 3

Initiate and monitor remote processing subsystems by performing the following tasks:

1. Bootload the DATANET 355 FNP.
2. Open all communication lines.
3. Unlock a remote.
4. Start the time sharing subsystem.
5. Terminate the time sharing subsystem.
6. Close all communication lines.

EXERCISE 4

Monitor the operation of the H6000 System from the system console while performing as a team member for a period of 8 hours. During this period, the trainee should perform the following tasks:

1. Set proper sieve limits as directed.
2. Monitor the system by properly entering verbs and responding to output messages.
3. Direct team members to perform peripheral operations as dictated by console messages.

EXHIBIT TPT050-1

EXERCISE 5

Execute master mode dumps by performing the following tasks:

1. Initiate a master mode dump from the system console.
2. Initiate a master mode dump from the processor maintenance panel.

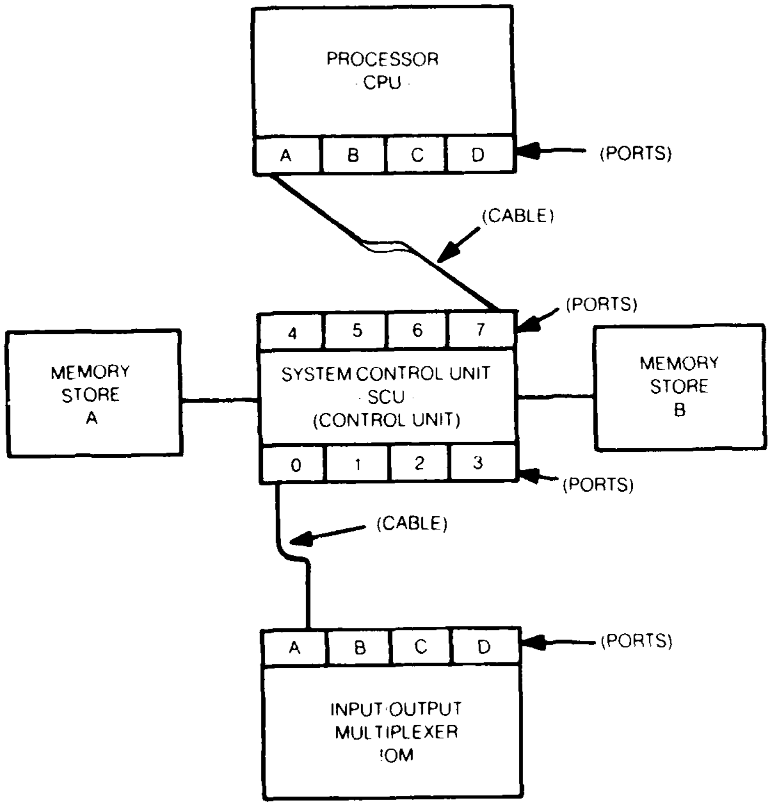


EXHIBIT OPR510-1

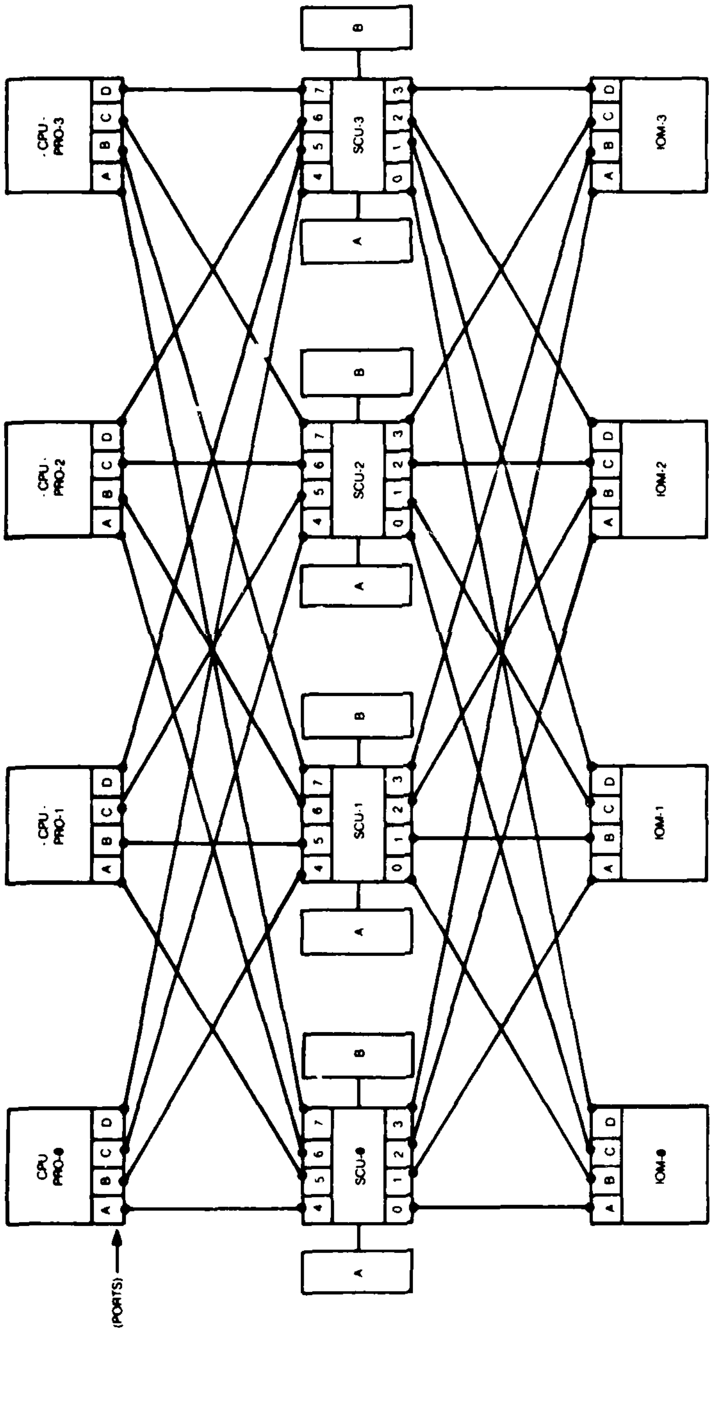


EXHIBIT OPR510-2

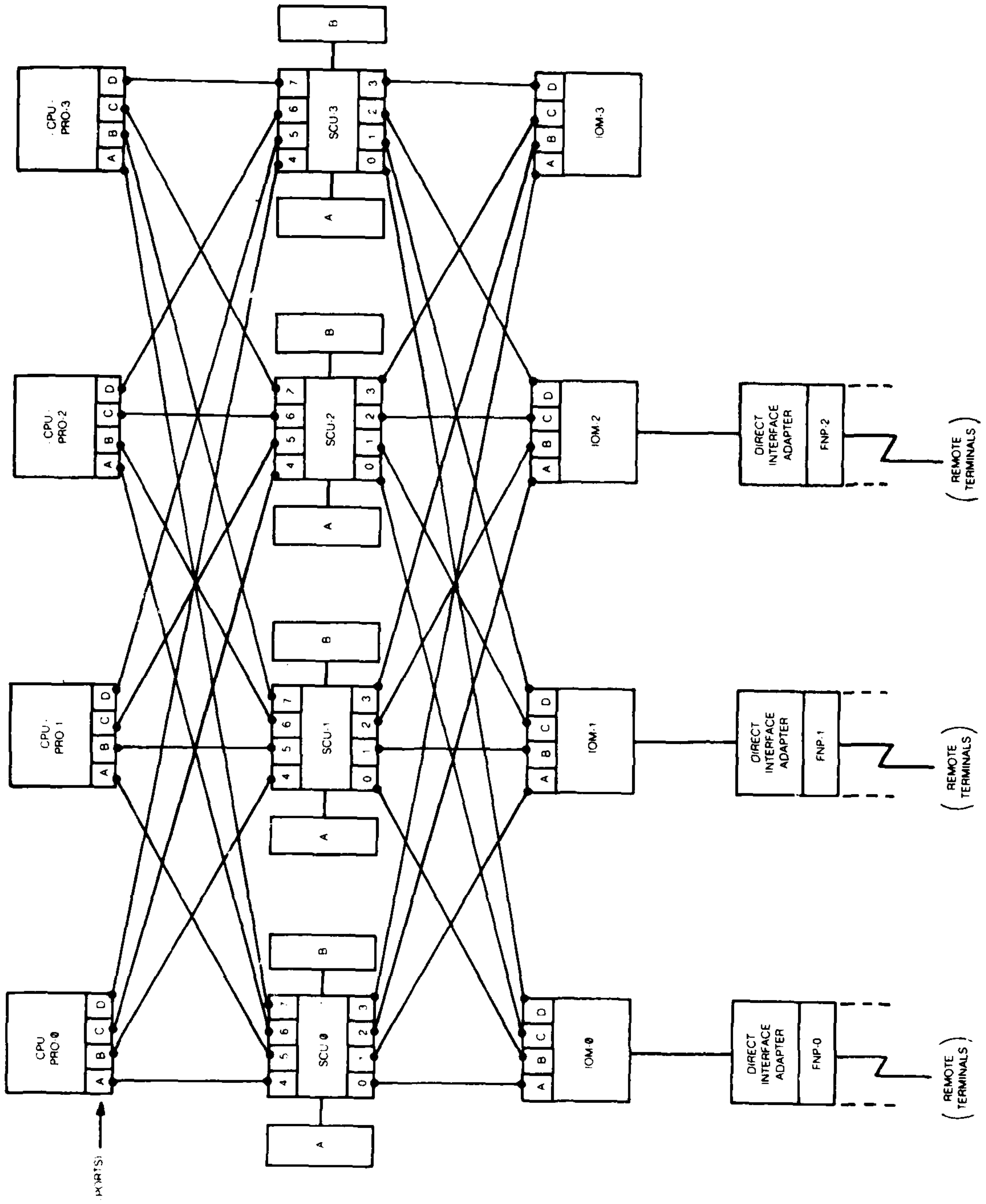


EXHIBIT OPR510-3

EXHIBIT OPR510-3

HARDWARE JOB FLOW - (BATCH)

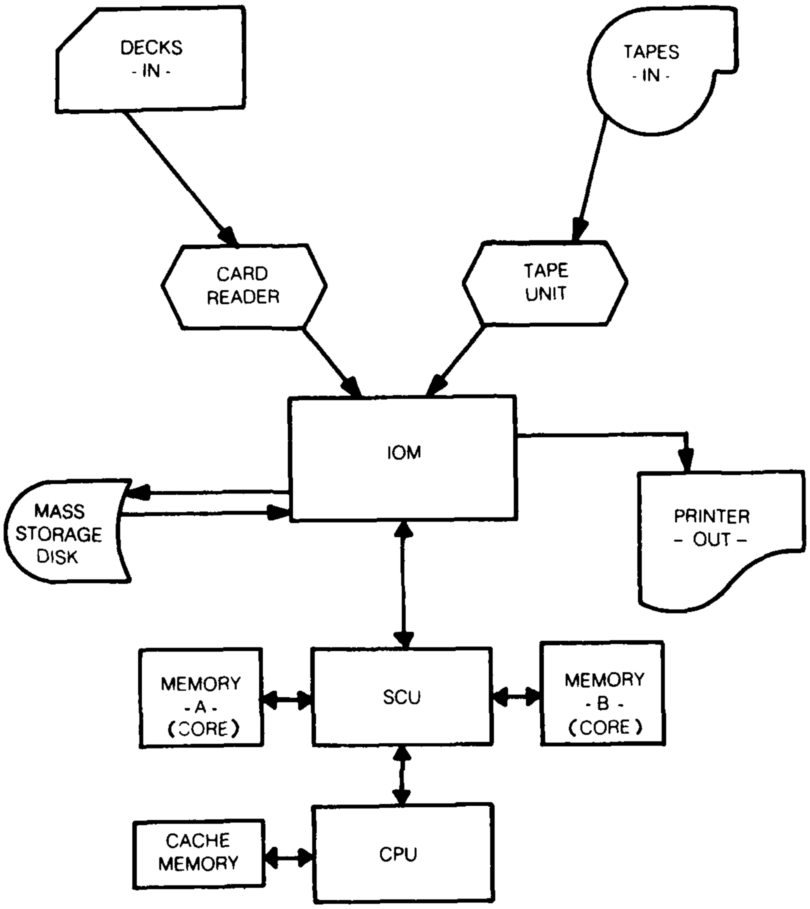


EXHIBIT OPR510-4

HARDWARE JOB FLOW - (REMOTE)

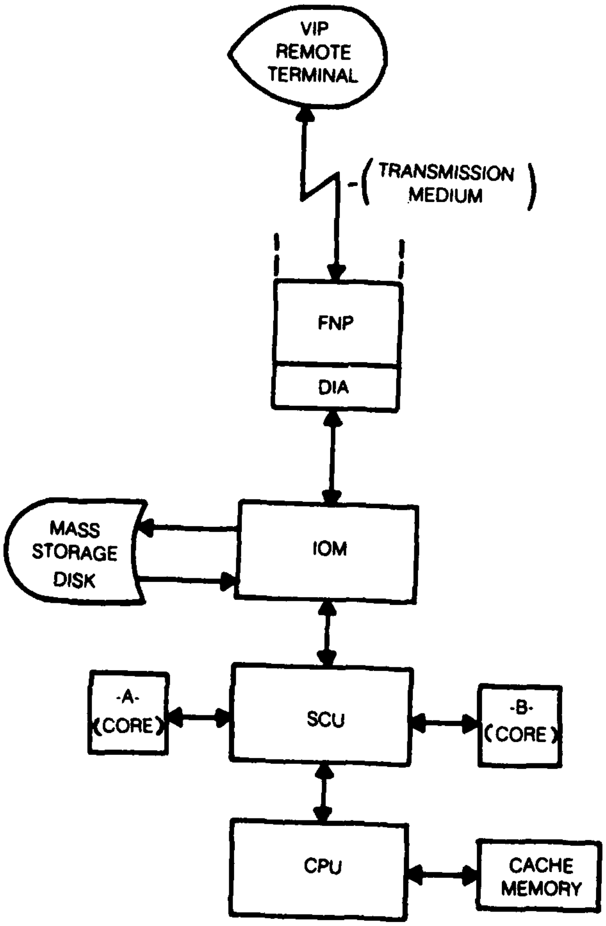


EXHIBIT OPR510-5

PRIVILEGED SLAVE PROGRAMS

PGM # (OCTAL)	SNUMB	NAME	REMARKS
1	$CALC	Core Allocator	(Always in memory)
2	$PALC	Peripheral Allocator	(Loaded as needed by System)
3	$SYOT	System Output Conversion	(Loaded as needed by System)
4	$RTIN	Job Scheduler and Remote Input	(Loaded as needed by System)
5	TS1	Time Sharing Supervisor	(Loaded by Operator Command)
6	$T/DS	Test and Diagnostics	(Loaded by System, or Operator)
7	$TRAX	Transaction Processor Executive	(Loaded by Operator Command)
10	$LOGN	Terminal Log on Supervisor	(Loaded by System)
11	NCP	Network Command Program (WIN)	(Loaded by Operator Command)
12	TLNT	Telenet (WIN)	(Loaded by Operator Command)
13	FTS	File Transfer System (WIN)	(Loaded by Operator Command)
15	$FSYS	File Management Supervisor	(Loaded by System)
72	$GEIN	System Input For Card Reader #6	(Loaded by System)
73	$GEIN	System Input For Card Reader #5	(Loaded by System)
74	$GEIN	System Input For Card Reader #4	(Loaded by System)
75	$GEIN	System Input For Card Reader #3	(Loaded by System)
76	$GEIN	System Input For Card Reader #2	(Loaded by System)
77	$GEIN	System Input For Card Reader #1	(Loaded by System)

EXHIBIT OPR520-1

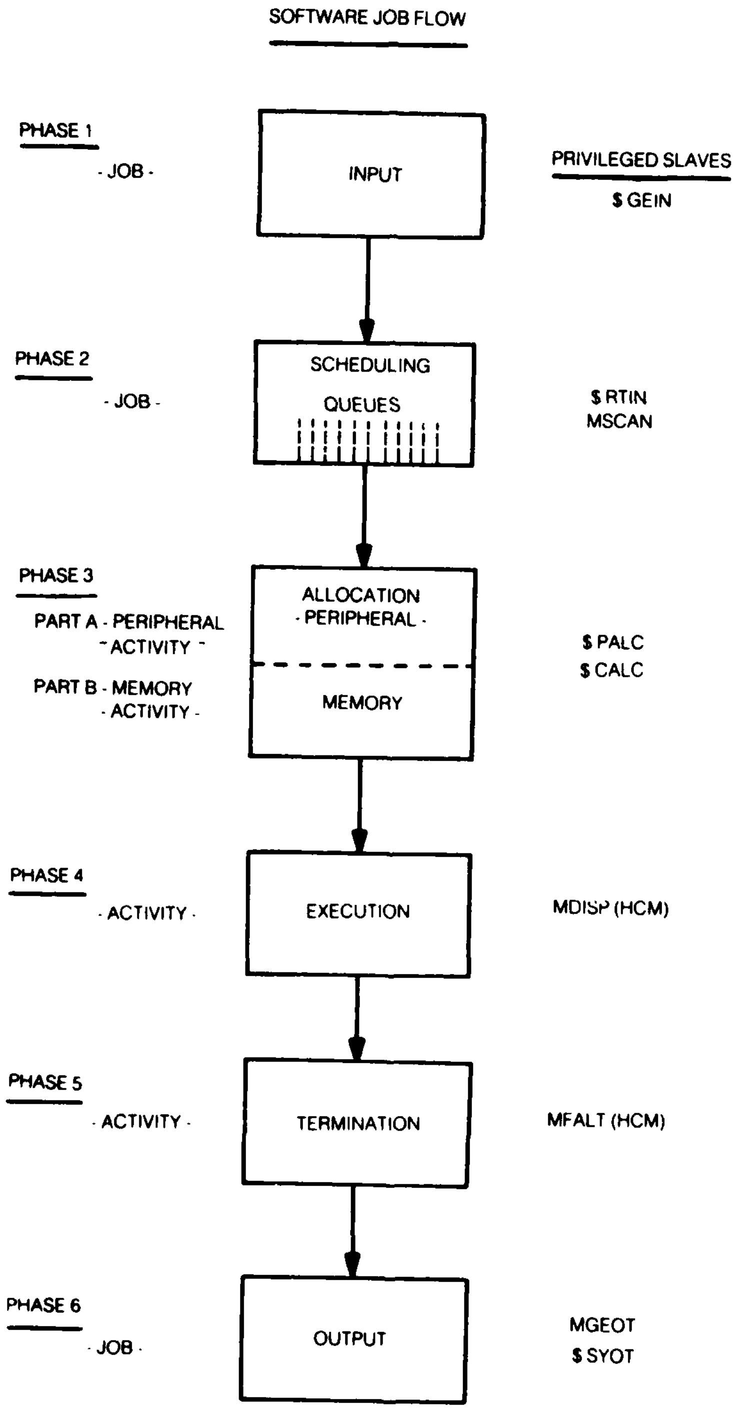


EXHIBIT OPR520-2

PHASE 1 INPUT

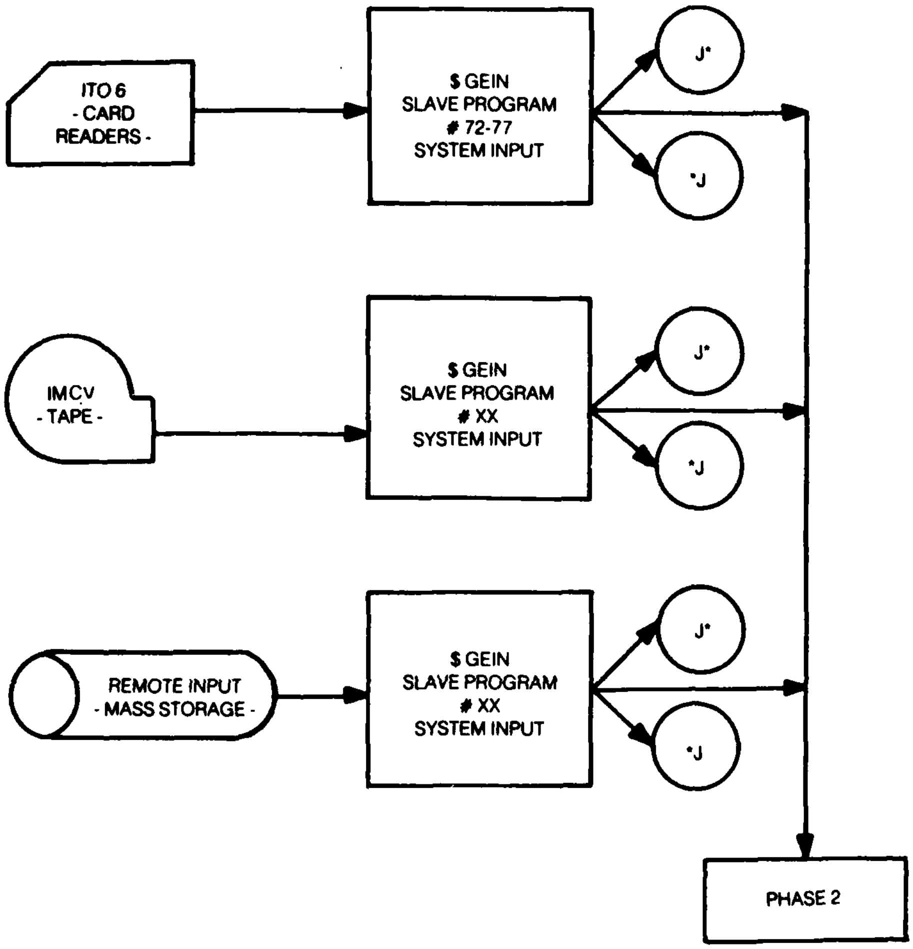


EXHIBIT OPR520-3

PHASE 2 SCHEDULING

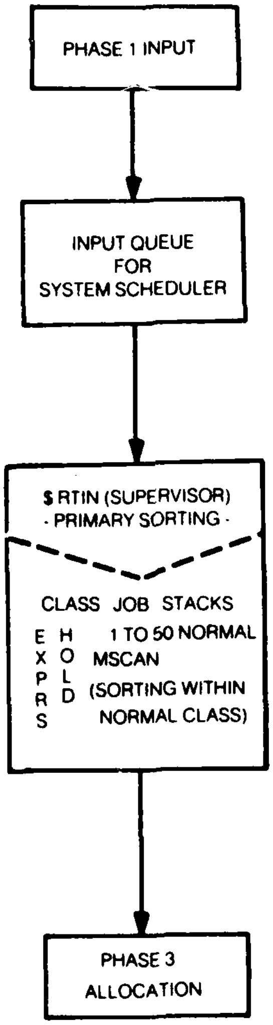


EXHIBIT OPR520-4

PHASE 3 ALLOCATION

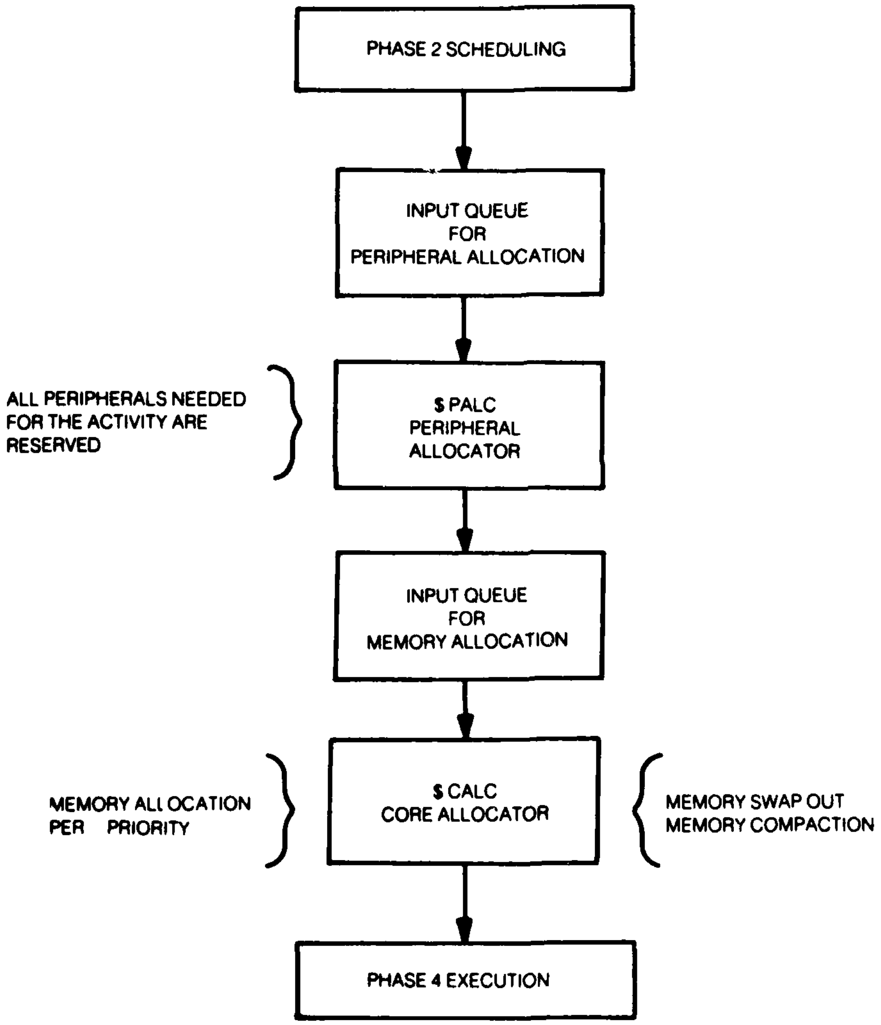

PHASE 2 SCHEDULING
INPUT QUEUE FOR PERIPHERAL ALLOCATION
ALL PERIPHERALS NEEDED FOR THE ACTIVITY ARE RESERVED
$ PALC PERIPHERAL ALLOCATOR
INPUT QUEUE FOR MEMORY ALLOCATION
MEMORY ALLOCATION PER PRIORITY
$ CALC CORE ALLOCATOR
MEMORY SWAP OUT MEMORY COMPACTION
PHASE 4 EXECUTION

EXHIBIT OPR520-5

PHASE 4 EXECUTION

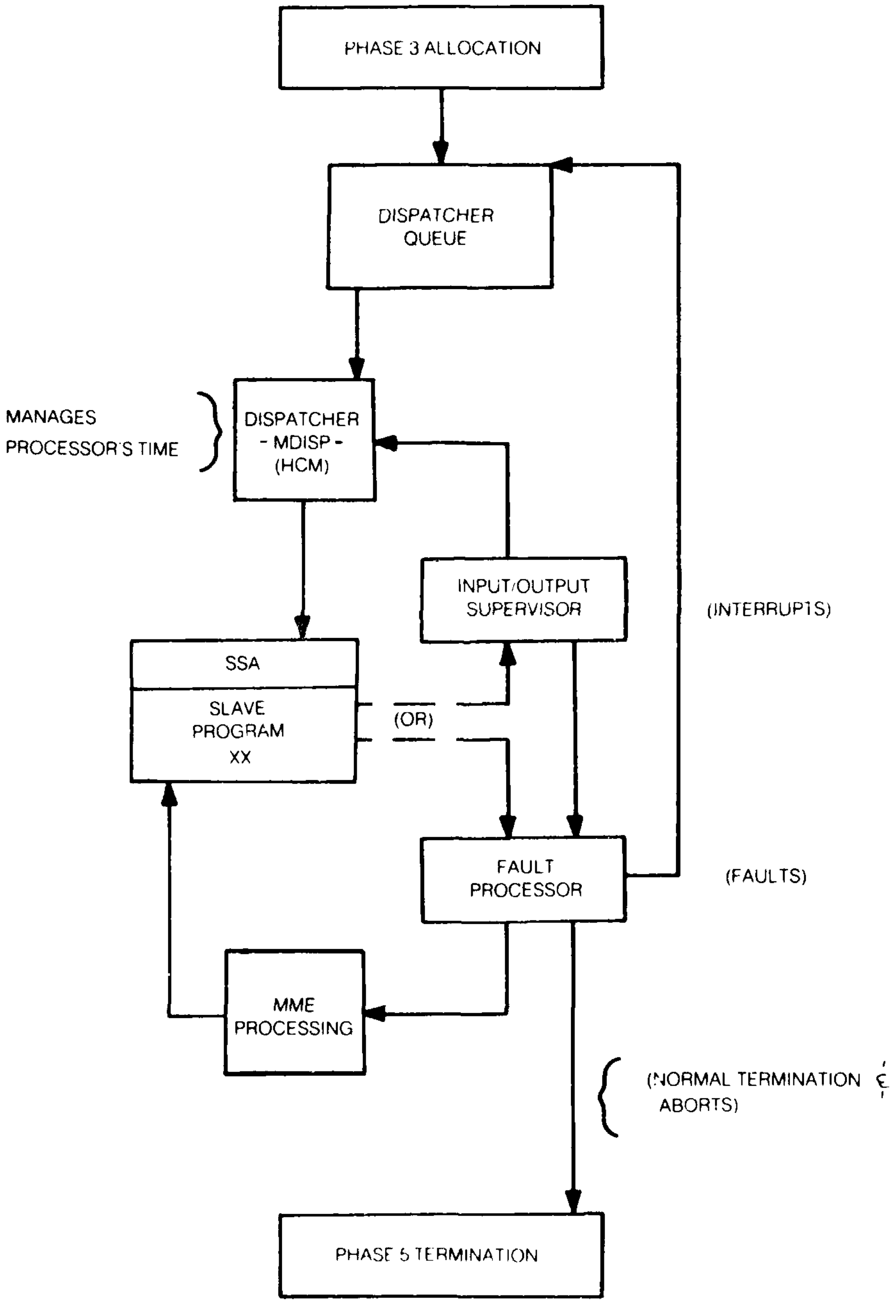


EXHIBIT OPR520-6

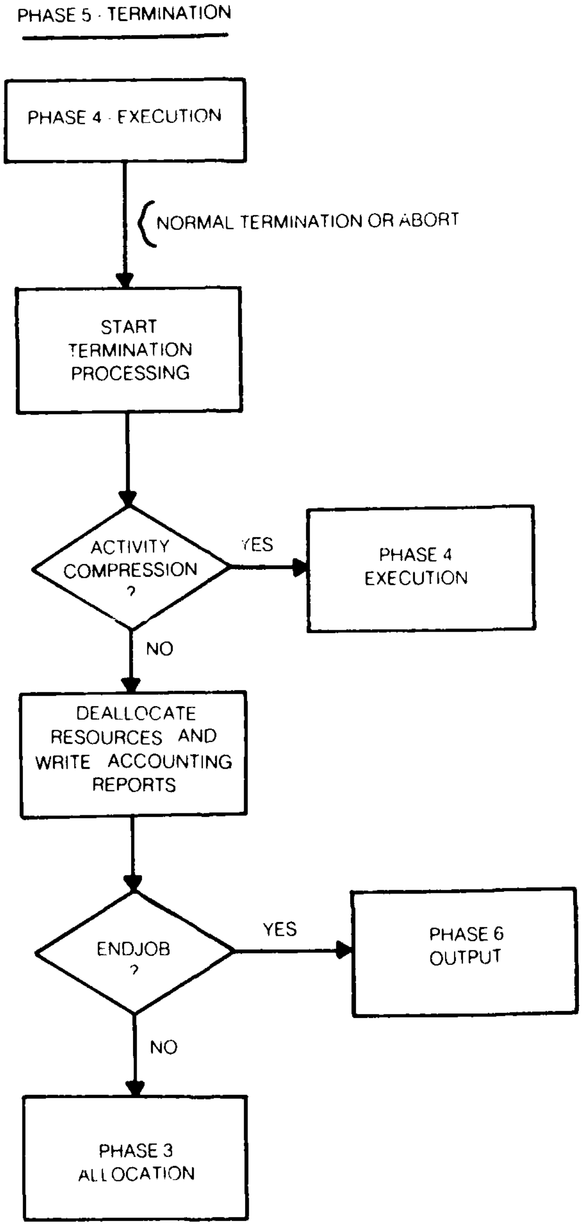


EXHIBIT OPR520-7

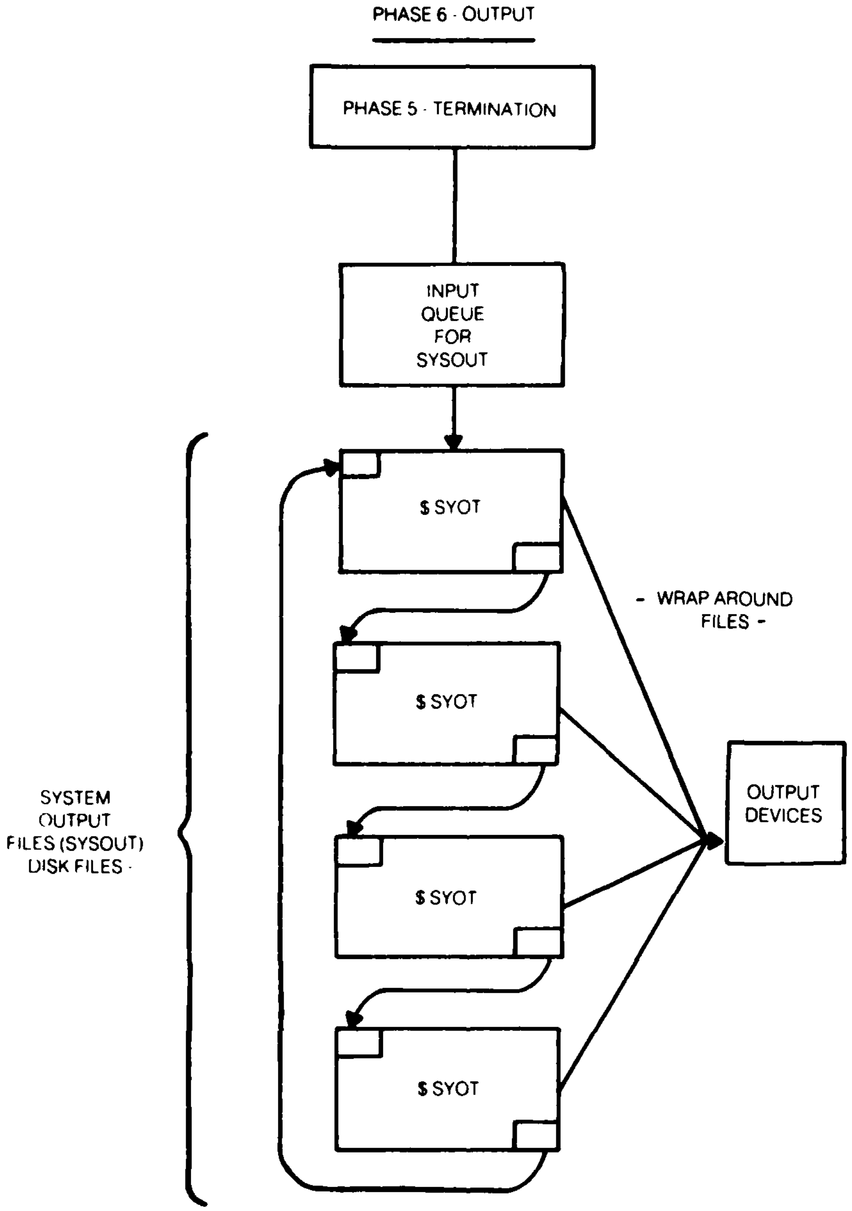
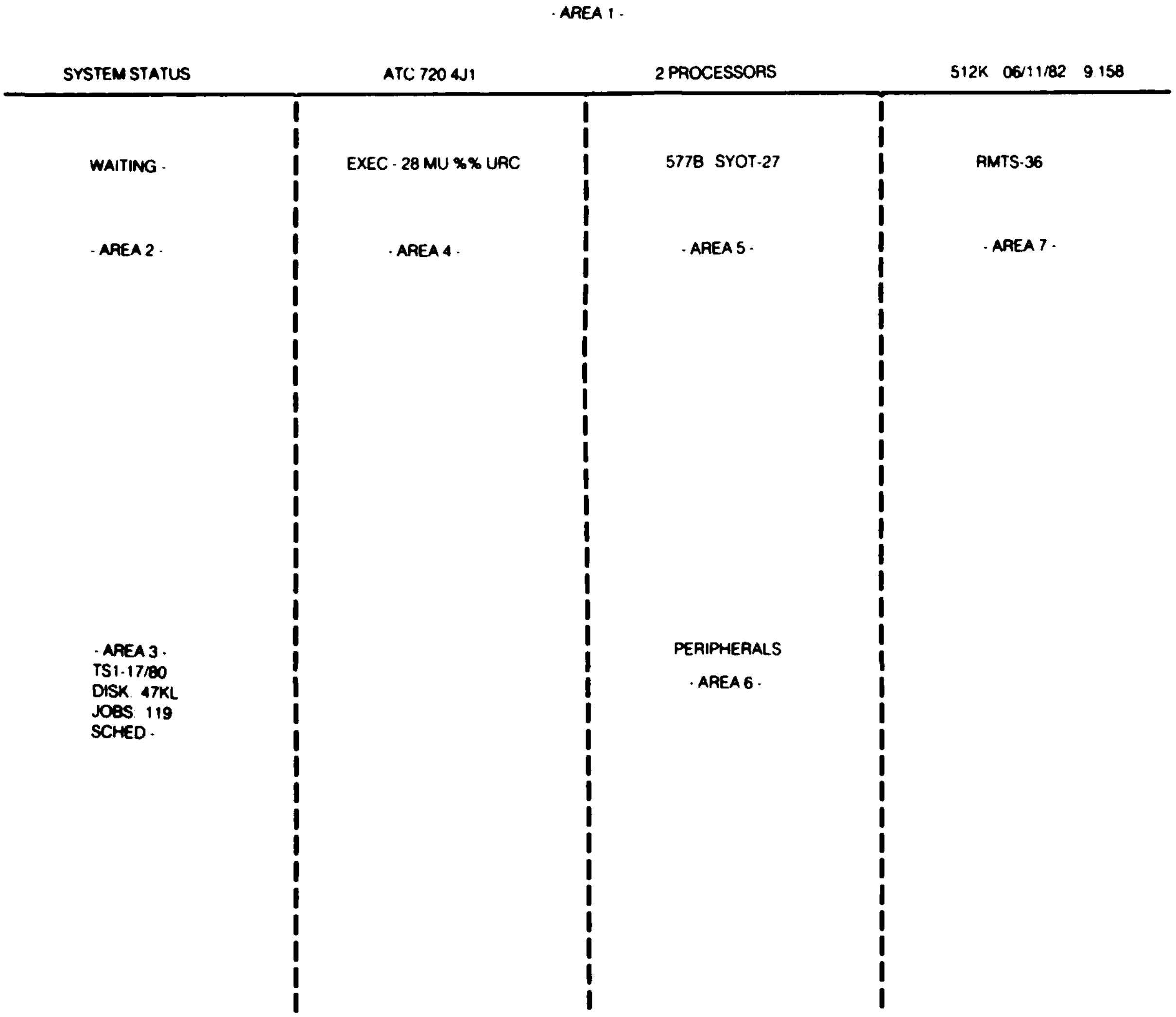


EXHIBIT OPR520-8

VIDEO



- AREA 1 -

SYSTEM STATUS	ATC 720 4J1	2 PROCESSORS	512K 06/11/82 9.158
WAITING -	EXEC - 28 MU %% URC	577B SYOT-27	RMTS-36
- AREA 2 -	- AREA 4 -	- AREA 5 -	- AREA 7 -
- AREA 3 - TS1-17/80 DISK 47KL JOBS 119 SCHED -		PERIPHERALS - AREA 6 -	

EXHIBIT OPR520-9

VIDEO: Area 2

```
waiting -
1  2839T-02  TP9
   3045T-06  Per
   3013T-01  HLD
59K  2929T-03 - COR
```

EXHIBIT OPR520-10

VIDEO: Area 3

```
TS1 - 17/80
DISK:  47KL
JOBS:  119
SCHED - 19
.HOLD  9
.ADX1  5-
.TAPE  5-
```

EXHIBIT OPR520-11

VIDEO: Area 4

```
EXEC - 15  mu %%  URG
IDLE:  229/120/
GCOS  53/28/
$PACL  22/Ø/Ø
$SYOT  30/1/62
$RTIN  17/1/Ø
TS1 -01  80/49/62
$LOGN  9/5/62
$FSYS  10/Ø/Ø
DMT    Ø1  4/SWP/Ø
658 [illegible] Ø1  19/5/32
DMSTA  8/SWP/Ø
$GEIN  11/Ø/61
*** MORE ***
```

EXHIBIT OPR520-12

VIDEO: Area 5

```
101B  SYOT - 9
+  00620
+  00622
*  00623
*  00624
*  00625
*  7990T  A5
*  8002T
*  00629
*  8019T
*  00631
```

EXHIBIT OPR520-13

VIDEO: Area 6

```
Peripherals

1T4 - RLSED
1T5 -
CP1 -
PR1 - $SYOT
RD1 - 6580T
RD2 -
RD3 -
RD4 - RLSED
PR2 - RLSED
```

EXHIBIT OPR520-14

VIDEO: Area 7

```
RMTS -        10

              R2 Idle
V             DØ Video
MCSK          CØ TS1
HNYCLASS4     D1 TS1
JOPS          G5 $LOGN
MCOM          C6 TS1
MCSK          B8 TS1
MCSK          B9 CNCT
MDOBBS        AU TS1
              R1 Idle
```

EXHIBIT OPR520-15

SELECTIVE INQUIRY SYSTEM (SIS) (H6000-CDT) COURSE EXHIBITS

```
$       PROGRAM RLHS,ON1,DUMP
$       LIMITS  ,20K,,8000
$       PRMFL   H*,R,R,cat/file string      (EDIT program)
$       PRMFL   RI,R,R,cat/file string      (SELECT program)
$       PRMFL   RS,R,R,cat/file string      (SORT program)
$       PRMFL   RR,R,R,cat/file string      (REPORT program)
$       SYSOUT  PA                          (printed report)
$       FILE    FA,X1S,5L                   (record output)
$       DATA    CC
        (directives follow this card)
```

EXHIBIT SIS010-1

FIELD NAME	POSITIONS	LENGTH
Employee Social Security Number	1 - 9	9
Department Number	10-11	2
Employee Name	12-31	20
Employee Address	32-51	20
City/State/Zip code	52-71	20
Hourly Rate	72-75	4
Sex	76	1
Marital Status*	77	1

*CODES

M	Married
S	Single
D	Divorced
W	Widowed

EXHIBIT SIS030-1

PHASE I OVERVIEW

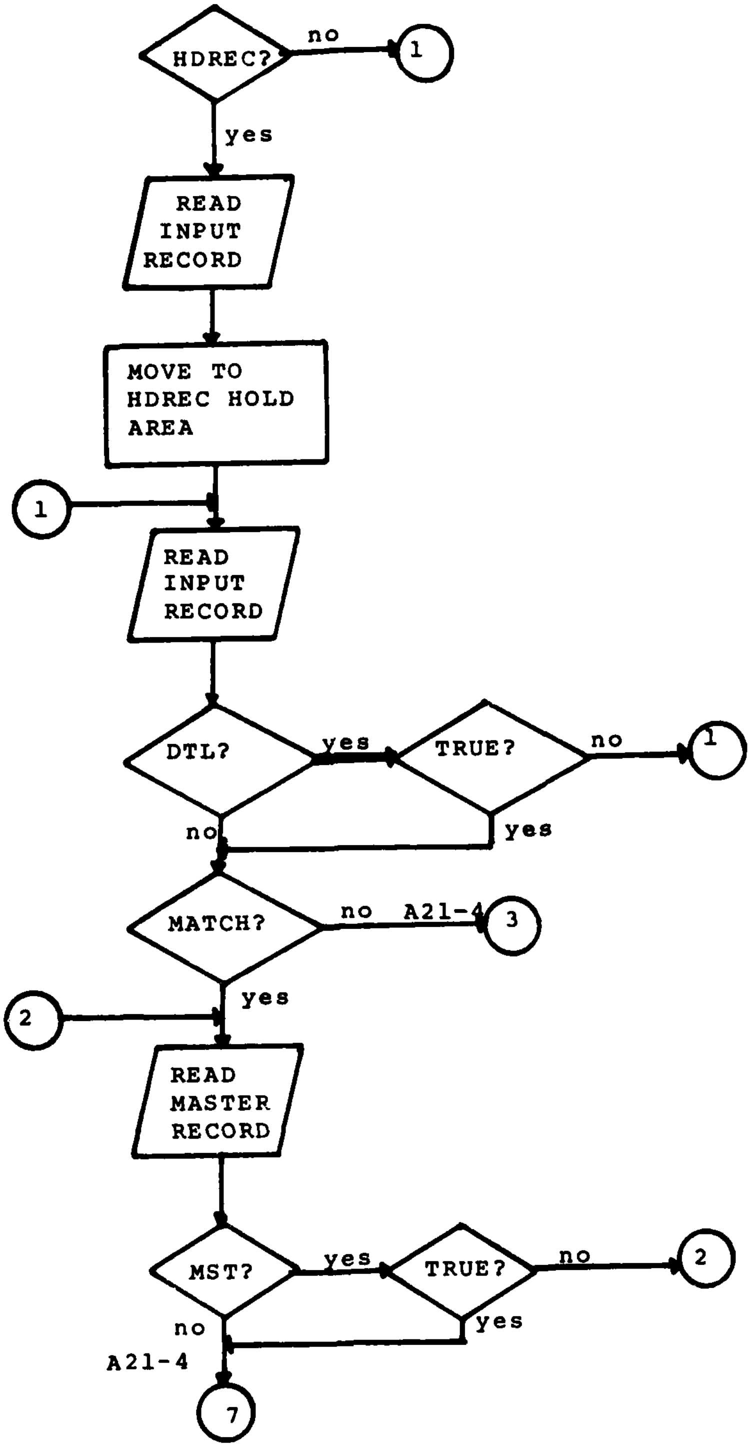


EXHIBIT SIS030-2

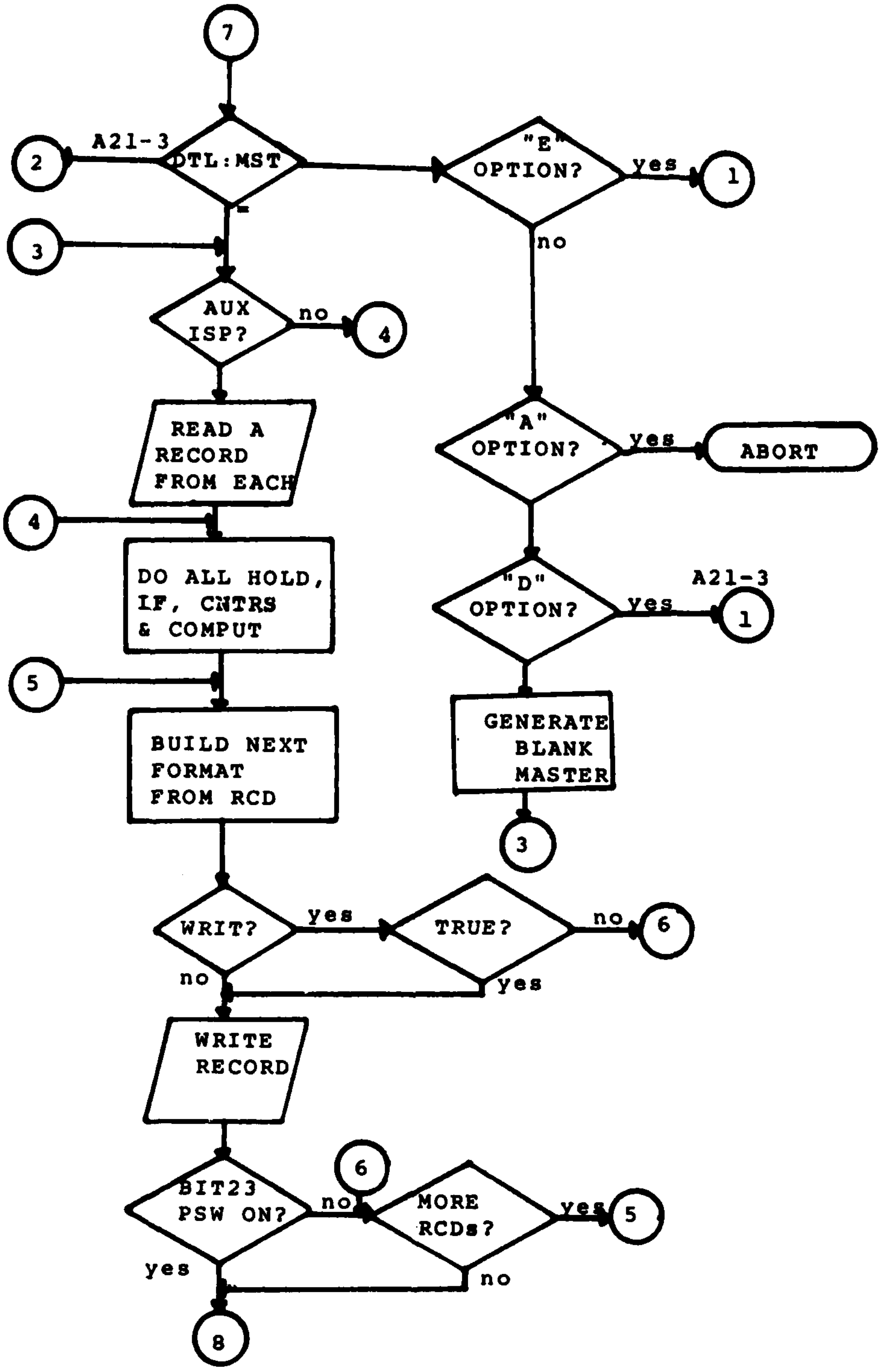


EXHIBIT SIS030-2 (Cont)

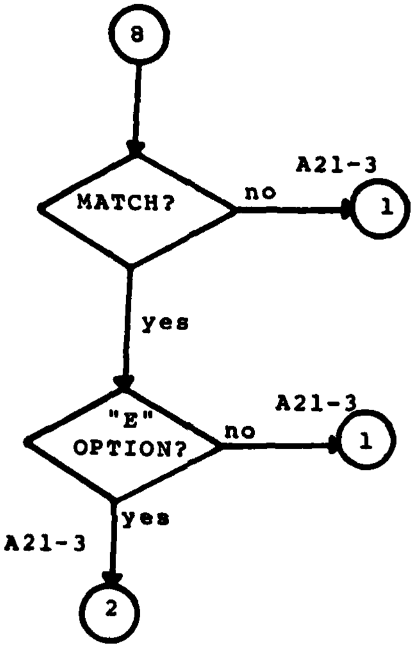


EXHIBIT SIS030-2 (Cont)

EXAMPLE SEQUENTIAL HIERARCHICAL FILE

Record Type "D"	Positions	Length
Department Number	1 - 2	2
Department Name	3 -22	20
Supervisor Name	23-42	20
(unused)	43-79	
Record Code "D"	80	1

Record Type "E"		
Department Number	1 - 2	2
Social Security Number	3 -11	9
Hourly Rate (BCD, Assumed decimal)	12-15	4
Type Employee Code (See Note)	16	1
Sex	17	1
Marital Status	18	1
Employee Name	19-38	20
Employee Address	39-58	20
City/State/Zip code	59-78	20
(unused)	79	
Record Code "E"	80	1

NOTE:
Type employee codes

S	Salaried
P	Production Line
C	Commissioned

EXHIBIT SIS030-3

EXAMPLE INDEXED SEQUENTIAL HIERARCHICAL FILE

Department Heading Record	Positions	Length
Department Number	1 - 2	2
Filler of zeroes	3 - 11	9
Department Name	12-31	20
Supervisor Name	32-51	20
(unused)	52-80	

Employee Detail Record		
Department Number	1 - 2	2
Social Security Number	3 -11	9
Hourly Rate (BCD, assumed decimal)	12-15	4
Type Employee (S,P,C)	16	1
Sex	17	1
Marital Status	18	1
Employee Name	19-38	20
Employee Address	39-58	20
City/State/Zip code	59-78	20
(unused)	79-80	

EXHIBIT SIS030-4

REFORMATTING SEQUENTIAL FILE FOR LOADING AS ISP FILE

```
WRIT1 D80 = "D" (See Note 1)
WRIT2 D80 - "D" (See Note 2)
RCD1WSL14,1/2,"000000000",3/40  (See Note 3)
RCD2WS114,1/78  (See Note 4)
```

NOTE 1: This sets up to write format code 1 if position 80 contains the lette. "D".

NOTE 2: This sets up to write format code 2 if position 80 does not contain the letter "D". It could also be coded: D80 = "E".

NOTE 3: This picks up positions 1-2, department number, from the department heading record, places nine zeroes in character positions 3-11 and copies positions 3-42 of the department heading record into positions 12-51 of the output record. The remainder of the 14 words will be filled with the appropriate fill character, as determined by the setting of bit 22 of the program switch word.

NOTE 4: This copies positions 1-78 of the employee record into positions 1-78 of the output record. Positions 79-80 will be filled with the appropriate fill character as above.

NOTE 5: Reference Exhibits SIS030-3 and SIS030-4 for file formats.

EXHIBIT SIS040-1

```
                                              NOTE
DTL   3 - "ØØØØØØ" AND 9 - "ØØØ"               1
ISPDTLSIZE=14,OFFSET=Ø,KEY=1/11                2
CNTRS C1 = 12$4, C2 = 12$4, C3 = 12$4          3
COMPUTC1 = C1 * 8Ø                             4
COMPUTC2 = C2 * 12Ø                            5
COMPUTC3 = C1 + C3 * 5                         6
WRITA D16 = "S"                                7
WRITB D16 = "P"                                7
WRITC D16 = "C"                                7
RCDAWSL14,1/16,19/20,C1D/8                     8
RCDBWSL14,1/16,19/20,C2D/8                     8
RCDCWSL14,1/16,19/20,C3D/8                     8
```

NOTES

1. In this example we are using the ISP payroll file defined in exhibit SIS030-4. The "DTL" directive is used to bypass all "department heading" records by specifying that only records that do not have zeroes in positions 3-11 are to be processed.

2. The "ISPDTL" directive defines the detail file as being in ISP format.

3. The "CNTRS" directive sets the BCD value in the hourly rate field into counters one through three.

4. This "COMPUT" directive multiplies the hourly rate in C1 by 80 and will be used for "salaried" employees later.

5. This "COMPUT" directive multiplies the hourly rate in C2 by 120 and will be used for "production" employees later.

6. This "COMPUT" directive multiplies the hourly rate in C3 by 5 and adds the value in C1, which has been multiplied by 80 in the first "COMPUT" directive. This result will be used for "commissioned" employees later.

7. The three "WRIT" directives determine the type of employee whose record is being processed.

8. The three "RCD" directives correspond to the "WRIT" directives. A record will be written on file code WS and the contents of counter 1, 2 or 3 will be used in the output record, depending on which type employee is being processed. All records will have Department Number, SSAN, Hourly Rate, Type Employee Code and Employee Name, followed by the appropriate computed value.

EXHIBIT SIS040-2

PHASE II OVERVIEW

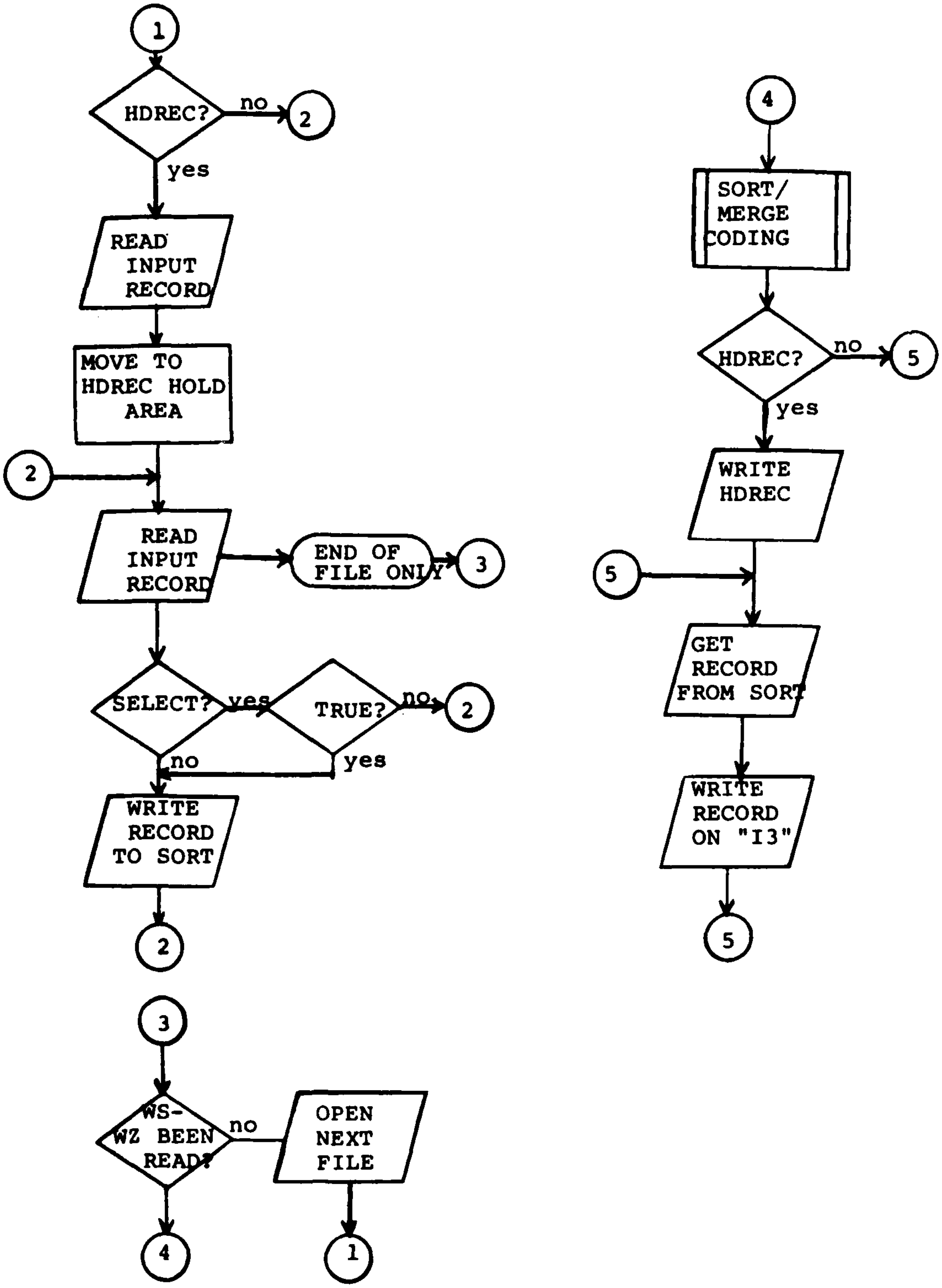


EXHIBIT SIS050-1

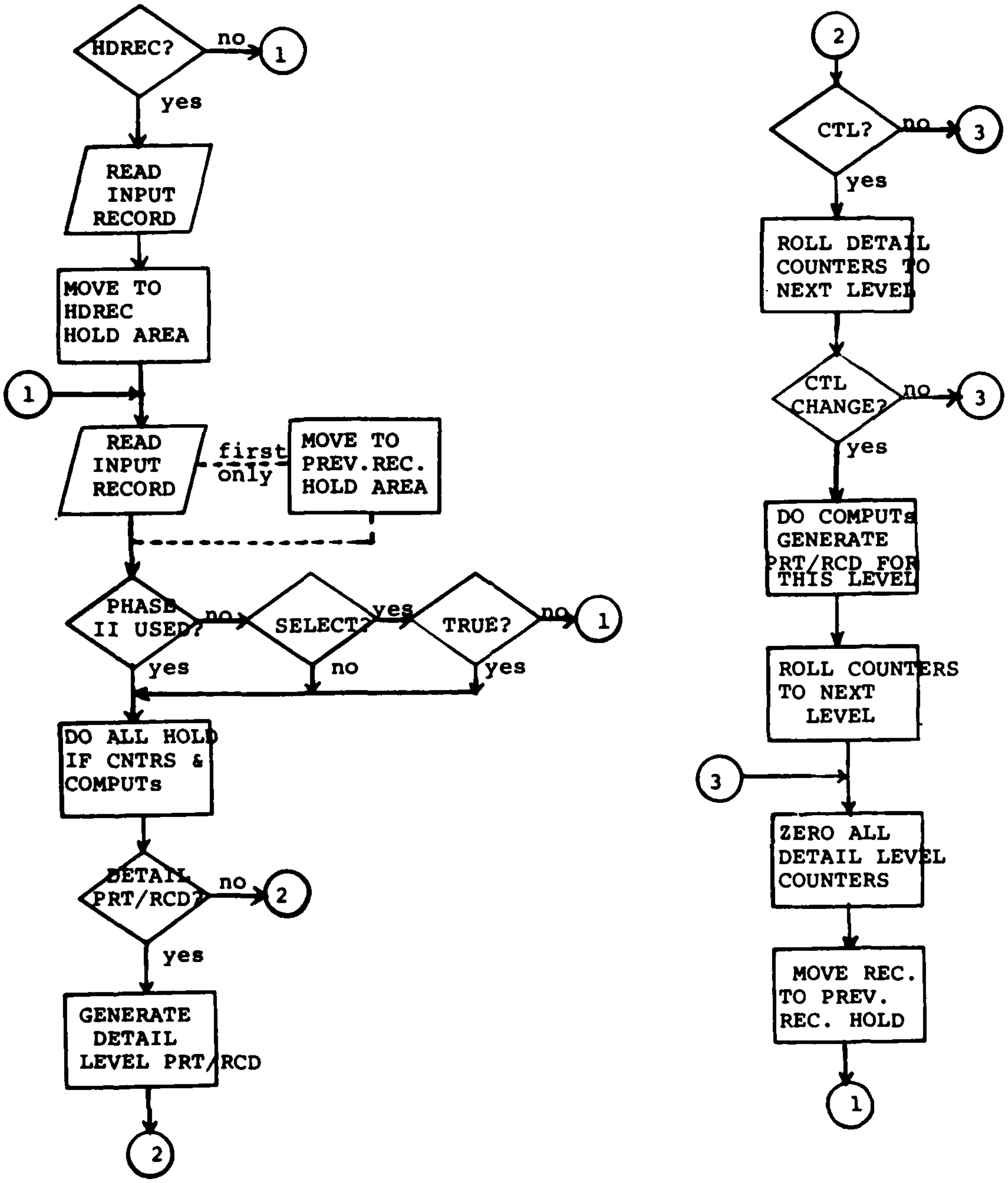

HDREC?
no
1
yes
READ INPUT RECORD
MOVE TO HDREC HOLD AREA
1
READ INPUT RECORD
first only
MOVE TO PREV.REC. HOLD AREA
PHASE II USED?
no
SELECT?
yes
TRUE?
no
1
yes
no
yes
DO ALL HOLD IF CNTRS & COMPUTs
DETAIL PRT/RCD?
no
2
yes
GENERATE DETAIL LEVEL PRT/RCD
2
2
CTL?
no
3
yes
ROLL DETAIL COUNTERS TO NEXT LEVEL
CTL CHANGE?
no
3
yes
DO COMPUTs GENERATE PRT/RCD FOR THIS LEVEL
ROLL COUNTERS TO NEXT LEVEL
3
ZERO ALL DETAIL LEVEL COUNTERS
MOVE REC. TO PREV. REC. HOLD
1

EXHIBIT SIS050-2

MANAGEMENT DATA QUERY SYSTEM (MDQS) BASIC USER (H6000-CDT) COURSE EXHIBITS

MDQ010 - INTRODUCTION TO MDQ - BASIC CONCEPTS AND TERMINOLOGY.
- BASIC DATA BASE TERMS AND CONCEPTS
- MDQ AND ITS USES
- DATA STRUCTURES/H6000 FILE ORGANIZATIONS
- MDQ CONTROL FILES AND RELATED FEATURES
  - DATA FILE
  - DATA DEFINITION FILE (DDF/DDL)
  - DIRECTORY
  - APPLICATION DEFINITION FILE (ADF/ADL)
  - PROCEDURE LANGUAGE (POL)
  - QUERY PROCEDURE LANGUAGE (QPL)
  - USER PROFILE SYSTEM (UPS)
  - MDQ CONVERSATIONAL LANGUAGE (CMDQ)
  - MDQ INTERFACE SOFTWARE PACKAGE (MISP)

MDQ020 - MDQ COMMAND LANGUAGE.
- TSS COMMANDS
- RUN
- CHECK
- TUTOR
- DJST
- REDIRECT
- RUNNING A PARAMETERIZED PROCEDURE
- ADFQ

MDQ030 - IDENTIFIERS, REPORT ELEMENT DESCRIPTORS, AND CONDITIONALS.
- IDENTIFIERS
- VARIABLES
- REPORT ELEMENT DESCRIPTORS
- CONDITIONALS
  - RELATIONAL TEST
  - PRESENCE/ABSENCE TEST
  - SPECIAL RELATION TEST
  - COMPOUND CONDITIONAL

MDQ040 - QUERY PROCEDURE LANGUAGE (QPL)
- QPL STATEMENTS
  - QUERY
  - SORT
  - PRINT
  - LET
  - COPY
  - REMARK
  - LIBRARY
  - OPTIONS ARE
  - END
- CODE AND RUN QPL PROCEDURES

EXHIBIT MDQ010-1

MDQ050 - ELEMENTARY PROCEDURE LANGUAGE (EPL).

- EPL STATEMENTS
  - INVOKE
  - RETRIEVE
  - SORT
  - PRINT
  - END
  - WRAP-UP
- CODE AND RUN EPL PROCEDURES
- SEGMENTS
  - EVENT SEGMENTS
  - MAIN SEGMENT
  - WRAP-UP SEGMENT

MDQ060 - MDQ REPORT PACKAGE.

- REPORT PACKAGE
  - REPORT STATEMENT
  - LINE STATEMENT
  - SPACE STATEMENT
- CODE AND RUN A QPL PROCEDURE WITH A REPORT PACKAGE

MDQ070 - MDQ STATISTICAL FUNCTIONS AND CONTROL SEGMENTS.

- CONTROL HEADINGS AND FOOTINGS
- STATISTICAL FUNCTIONS
- COUNT AND SUM FUNCTIONS IN A LET STATEMENT
- CONTROL SEGMENTS
  - CONTROL STATEMENT
  - ENABLE STATEMENT
  - ENABLING CLAUSE
- CODE AND RUN AND EPL PROCEDURE WITH A REPORT PACKAGE, SUM FUNCTION, AND CONTROL AND WRAP-UP SEGMENTS

MDQ080 - MDQ CONVERSATIONAL LANGUAGE (CMDQ)

- INTRODUCTION TO CMDQ
- CMDQ DIALOGUE (BROWSE FUNCTION)
- USE CMDQ TO RUN MDQS PROCEDURES
- PREVIEW OF THE MDQS ADVANCED USER COURSE

EXHIBIT MDQ010-1 (Cont)

SINGULAR DATA STRUCTURE

Part 1

Part 2

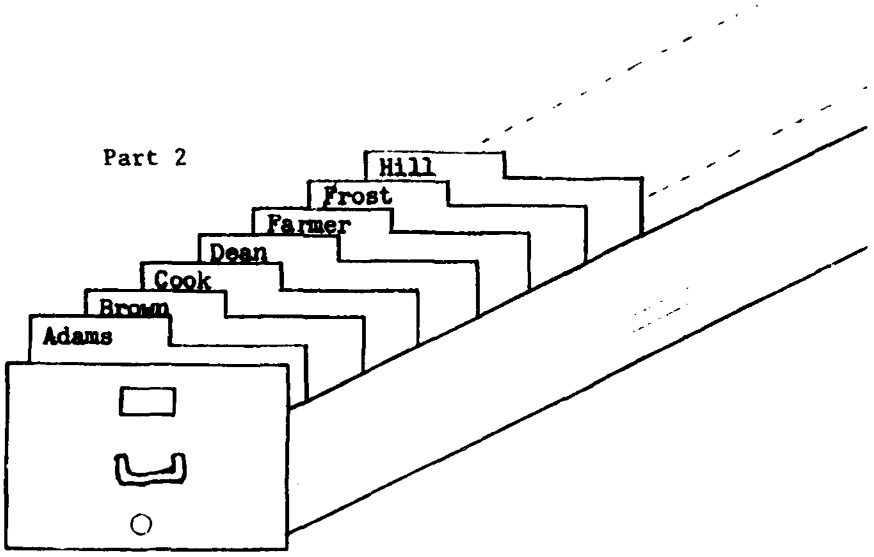
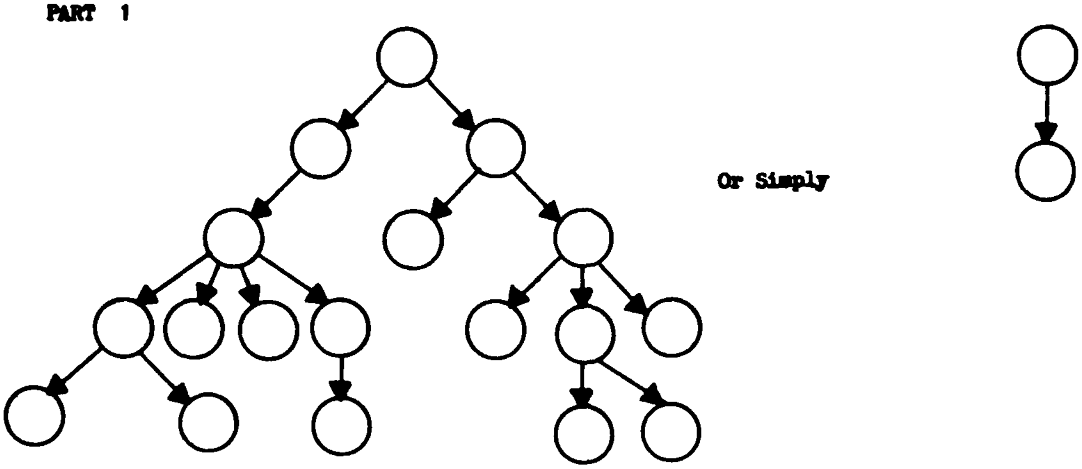


EXHIBIT MDQ010-2

HIERARCHICAL DATA STRUCTURE

PART 1



PART 2

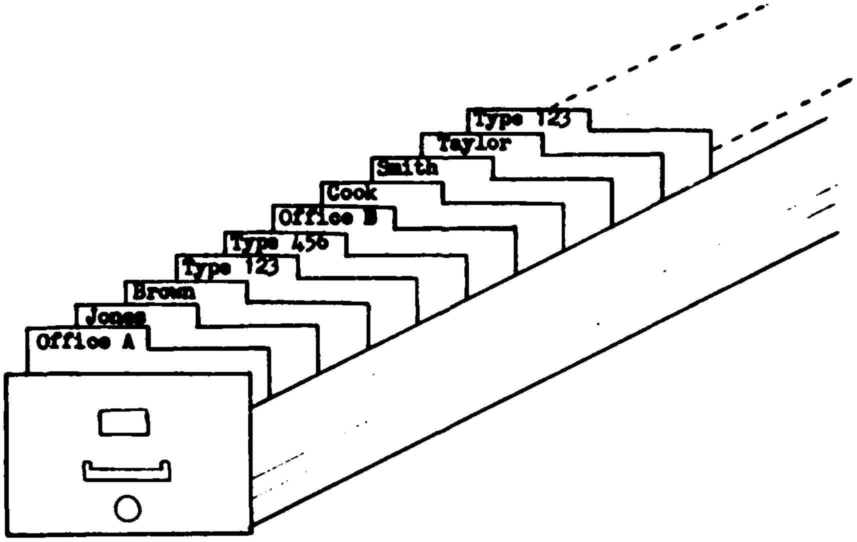
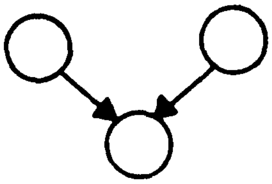


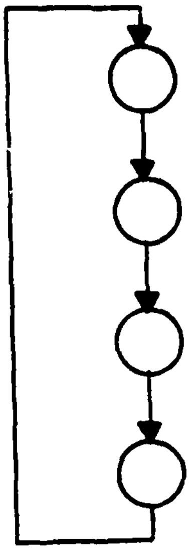
EXHIBIT MDQ010-3

ILLEGAL HIERARCHICAL DATA STRUCTURES

Part 1



Part 2

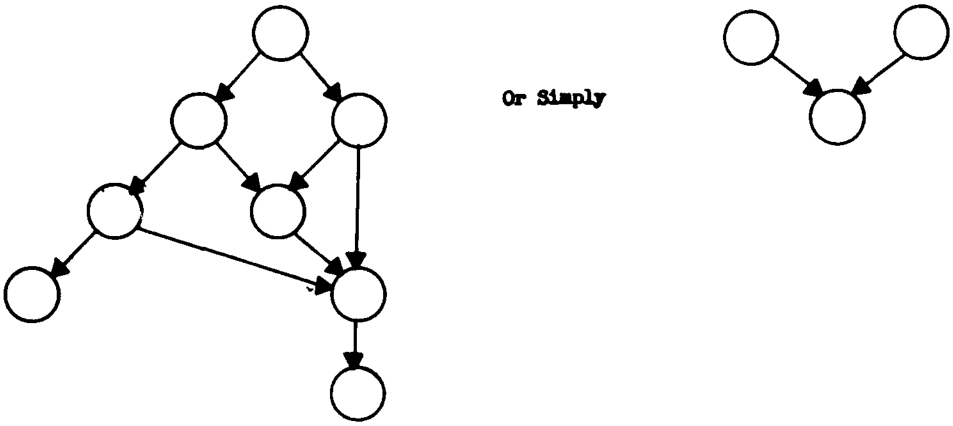


Circular

EXHIBIT MDQ010-4

NETWORK DATA STRUCTURE

PART 1



PART 2

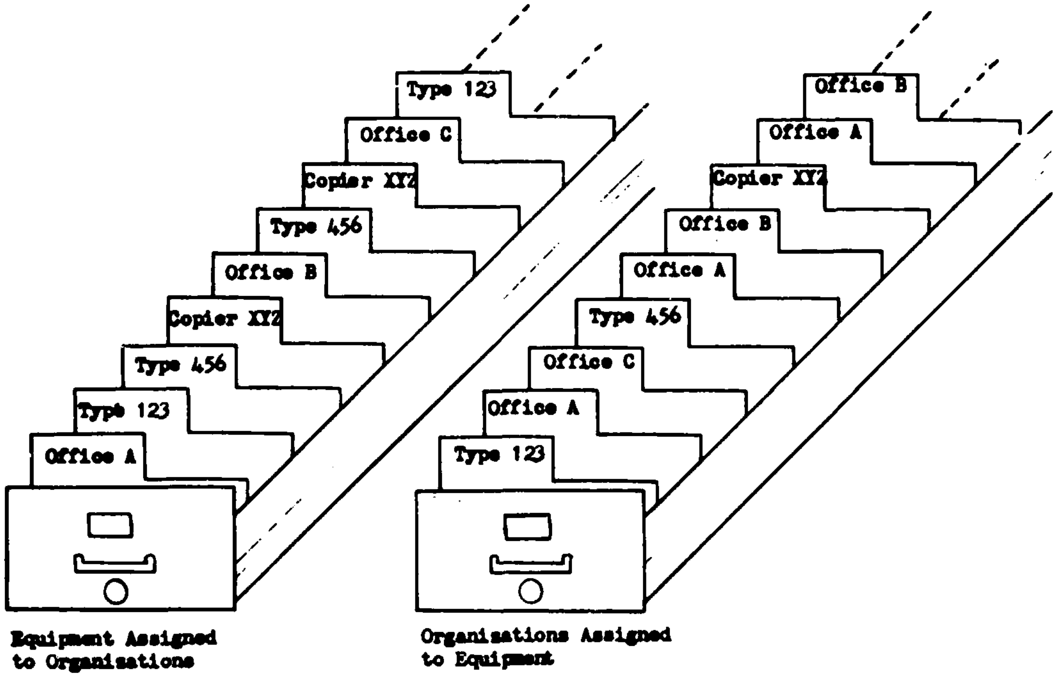
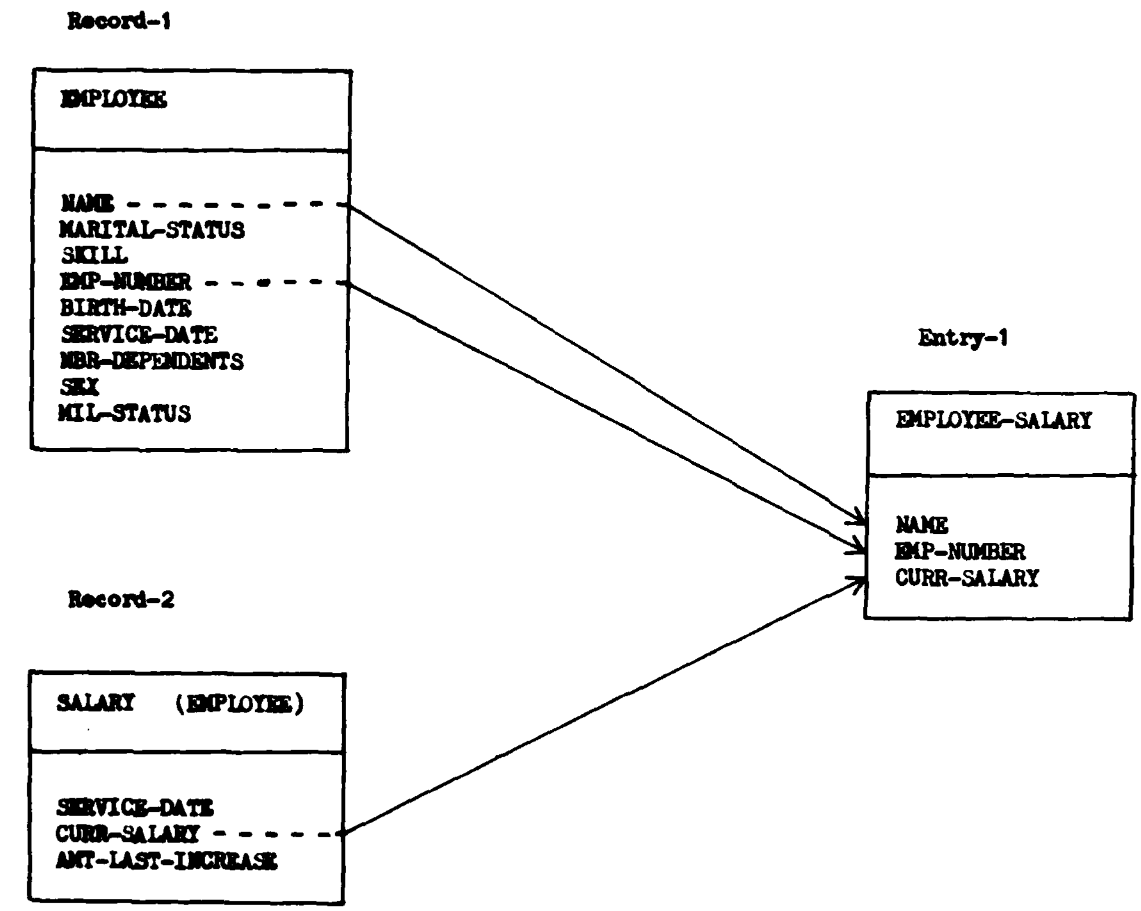


EXHIBIT MDQ010-5

ENTRIES AND DATA ITEMS



Description in DDF

Description in ADF

EXHIBIT MDQ010-6

HIERARCHICAL DATA STRUCTURE

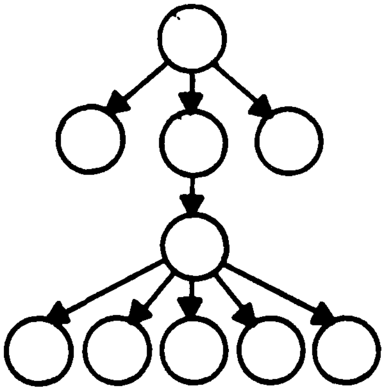


EXHIBIT MDQ010-7

MANUAL SYNTAX

PART 1.

person-name RAN DOWN THE BUSY STREET.

PART 2

person-name $\left\{\begin{array}{l}\underline{\text{RAN}}\\ \underline{\text{WALKED}}\end{array}\right\}$ DOWN [THE BUSY] STREET.

PART 3.

JOHN RAN DOWN STREET.

JOHN RAN DOWN THE STREET.

JOHN RAN DOWN THE BUSY STREET.

JOHN WALKED DOWN STREET.

JOHN WALKED DOWN THE STREET.

JOHN WALKED DOWN THE BUSY STREET.

EXHIBIT MDQ010-8

REPORT ELEMENT DESCRIPTOR FORMATS

FORMAT-1

```
    LITERAL  [TITLE CLAUSE ]   [COLUMN CLAUSE  ]
```

FORMAT-2

```
    IDENTIFIER    [ TITLE CLAUSE ]  [COLUMN CLAUSE ]
         [ STATISTICAL FUNCTIONS ]  [ GROUP INDICATE/SUPPRESS CLAUSE ]
         [ JUSTIFIED CLAUSE ]   [ PICTURE CLAUSE ]
```

FORMAT-3

```
    IDENTIFIER    [ TITLE CLAUSE ]   [ COLUMN CLAUSE ]
         [ TEXT CLAUSE ]
```

EXHIBIT MDQ030-1

COLUMN CLAUSE FORMAT

$$\left\{\begin{array}{l}\underline{\text{COL}}\\ \underline{\text{COLUMN}}\end{array}\right\} \quad [+] \quad \text{NNN}$$

EXHIBIT MDQ030-2

DATA BASE

DAYTON, OHIO

DAYTON, CONNECTICUT

DAYTONA BEACH, FLORIDA

DAYTONA 500

EXHIBIT MDQ030-3

The following are examples of entering the MDQ subsystem build mode; a QPL procedure that retrieves, sorts, and prints; the procedure execution sequence; and the QPL procedure generated report.

(All underlined words are entered by the QPL procedure writer.)

```
* MDQ

* 010 QUERY UMC/SUB-CAT/OADF WHERE RANK = "02" AND DOB LT 500000

  020 SORT ON DOB DESCENDING, NAME DESCENDING

  030 PRINT ON REMOTE R1 DOB, NAME, AFSC, SEX

  040 END

RUN

RUN-ID? LAST-NAME       (up to 12 characters)

SNUMB 1234T


DOB       NAME              AFSC    SEX

490417    JONES JOHN P      5141     M

490104    SMITH ALTON Q     5144     M

481128    BROWN BRENDA A    5144     F

471212    MASON WALTER S    5141     M
```

EXHIBIT MDQ040-1

QUERY STATEMENT FORMAT

```
QUERY   ┌catalog/file-string┐┌┌ WHERE ┐                        ┐
        └*                  ┘│└ WITH  ┘    compound-conditional│
                             └                                 ┘
```

CODED EXAMPLES OF THE QUERY STATEMENT

```
010 QUERY UMC/SUB-CAT/OADF WITH SAAN = 426648033

010 QUERY UMC/SUB-CAT/OADF WHERE DOB GT 570601

010 QUERY UMC/SUB-CAT/OADF WHERE SEX EQ "F"
020 AND AFSC = "T5141" OR AFSC = "5141"
030 AND MS = "M"

010 QUERY UMC/SUB-CAT/OADF WHERE NAME CONTAINS WORD "JONES"
```

EXHIBIT MDQ040-2

LET STATEMENT FORMAT

$$\underline{\text{LET}}\ \text{simple-identifier}\ \left[\text{mask}\right]\ = \left\{\begin{array}{l}\text{identifier}\\ \text{literal}\\ \text{arithmetic-expression}\\ \text{numeric constant}\end{array}\right\}$$

CODED EXAMPLES OF THE LET STATEMENT

```
020 LET RANK [1 - 2] = "04"

020 LET YTD-PAY = 0

020 LET NAME [12 - 20] = "    "

020 LET NO-DEP = NO-DEP + 1
```

EXHIBIT MDQ040-3

PRINT STATEMENT FORMAT

```
PRINT

                        [{DUP      }    n  ]
       {PRINTER         [{DUPLICATE}       ]                                      }
                        [{DUP      }    n  [,id {DUP      }  n ]  ... ]
       {REMOTE    id    [{DUPLICATE}       [    {DUPLICATE}    ]      ]           }
  [                                                                               ]
  [ ON {FILE   catalog/file-string        [,  catalog/file-string]  ...           }]
  [                                                                               ]
             [FOR       {TTY                     }]
             [          {CRT                     }]
             [          {PRINTER      [IN BCD]   }]
       {DAC                                                                       }
        report-elem-descr          [,report-elem-descr]   ...
```

CODED EXAMPLES OF THE PRINT STATEMENT

```
050  PRINT ON REMOTE R1 NAME, RANK, SSAN

040  PRINT ON PRINTER DUP 3 RANK, NAME, AFSC

040  PRINT ON DAC SSAN TITLE "SOC-SEC", NAME COL 20,
050  RANK COL + 6, AFSC

040  PRINT ON FILE UMC/PERS-DATA/PROC-1 FOR TTY NAME, RANK,
050  SSAN, AFSC
```

EXHIBIT MDQ040-4

SORT STATEMENT FORMAT

```
SORT

 ON   sort-key-1 [{DESCENDING}]
                 [{DSC       }]

  [ ,sort-key-n [{DESCENDING}]   ... ]
  [             [{DSC       }]       ]
```

CODED EXAMPLES OF THE SORT STATEMENT

```
040   SORT ON RANK

030   SORT ON NAME, AFSC DSC

030   SORT ON NAME DESCENDING, DOB DESCENDING, NOD
```

EXHIBIT MDQ040-5

ILLUSTRATION PROBLEM FOR MDQ050

You have been assigned to write a procedure that will access the "COMPANY-STRUCTURE" data base to extract a list of all sites and the site managers. The list is to be printed on the company computer facility's on-line printer.

EXHIBIT MDQ050-1

EPL PRINT STATEMENT FORMAT

```
           ┌ PRINTER  [{DUPLICATE}     n]
           │          [{DUP      }      ]
           │
           │ REMOTE id [{DUPLICATE}    n [,id {DUPLICATE}       n] . . . ]
           │           [{DUP      }      [    {DUP      }        ]       ]
PRINT ON  <│
           │ FILE [{REMOVABLE}]     catalog/file-string [,[{REMOVABLE}]
           │      [{RMVBL    }]                         [ [{RMBVL    }]
           │
           │      catalog/file-string]     . . . [FOR  {TTY                  }]
           │                                           {CRT                  }
           │                                           {PRINTER     [IN BCD] }
           │                                           {terminal-name        }
           │
           └ TAPE "tape-label"

           {report-elem descr}      [,report-elem-descr] . . .
           {entry-name       }
```

EXHIBIT MDQ050-2

NUMERIC OUTPUT EDITING EXAMPLES

EDITING FUNCTION	INPUT PICTURE	CONTENT	OUTPUT PICTURE	PRINTED VALUE
SUPPRESS LEADING ZEROES	9(6)	007231 000000	"Z(5)9"	7231 0
INSERT	9(6)	137026 000152	"ZZZ,ZZ9"	137,026 152
INSERT . AND , SUPPRESS ZEROES	9(6)V99	01375678 00000000	"ZZZ,ZZ9.99"	13,756.78 0.00
INSERT $	9(6)V99	01375678	"$ZZZ,ZZ9.99"	$ 13,756.78
FLOAT $	9(6)V99	00125435 12345678 00000000	"$$$$,$$9.99"	$1,254.35 $123,456.78 $0.00
PROTECTING *	9(6)V99	00125435 00000000	"***,**9.99"	**1,254.35 ******0.00
+ OR -	S9(6)V99 ITEM CONTAINS A SIGNED VALUE (S = SIGN)	$\overset{-}{0}$0125435 (NEGATIVE)# $\overset{+}{0}$0654321 (POSITIVE)#	$$$$$,$$9.99-"	$1,254.35- $6,543.21
		$\overset{-}{0}$0125435 $\overset{+}{0}$0654321	$$$$$,$$9.99+"	$1,254.35- $6,543.21+

NOTE: # - For illustrative purposes, the + or - signs are shown over the high-order digit.

EXHIBIT MDQ050-3

EPL SORT STATEMENT FORMAT

$$\underline{\text{SORT}} \quad \begin{Bmatrix}\text{entry-name}\\ \text{identifier}\end{Bmatrix} \quad \left[\text{,identifier}\right] \quad \ldots$$

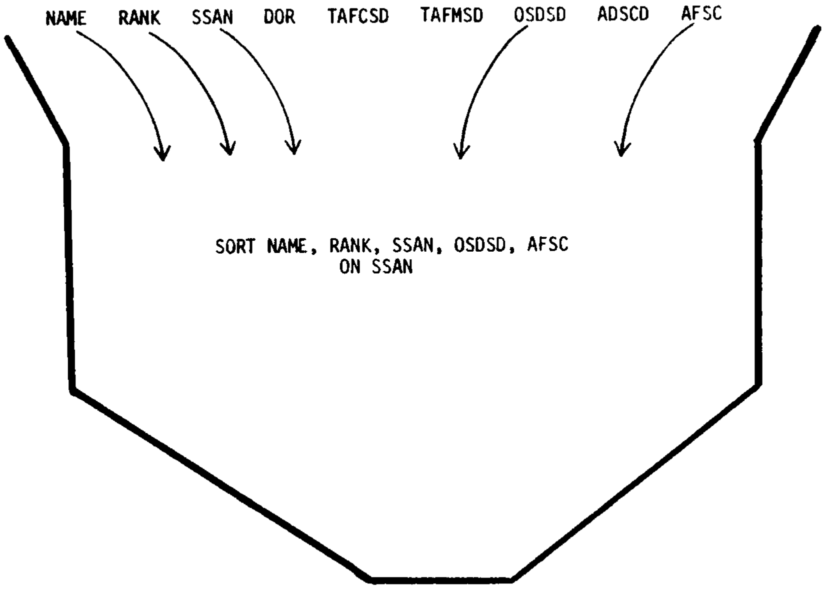
$$\underline{\text{ON}}\ \text{sort-key-1} \quad \left[\text{IN} \quad \begin{Bmatrix}\underline{\text{ASC}}\\ \underline{\text{ASCENDING}}\\ \underline{\text{DSC}}\\ \underline{\text{DESCENDING}}\end{Bmatrix} \quad \left[\begin{Bmatrix}\underline{\text{ORDER}}\\ \underline{\text{SEQUENCE}}\end{Bmatrix}\right]\right]$$

$$\left[\text{,sort-key-2} \quad \left[\text{IN} \quad \begin{Bmatrix}\underline{\text{ASC}}\\ \underline{\text{ASCENDING}}\\ \underline{\text{DSC}}\\ \underline{\text{DESCENDING}}\end{Bmatrix} \quad \left[\begin{Bmatrix}\underline{\text{ORDER}}\\ \underline{\text{SEQUENCE}}\end{Bmatrix}\right]\right] \ldots\right]$$

EXHIBIT MDQ050-4

DIAGRAM OF THE SORT SIEVE

ALL ITEMS IN AN ENTRY ARE AVAILABLE



SORT NAME, RANK, SSAN, OSDSD, AFSC
ON SSAN

ONLY AVAILABLE ITEMS

NAME,RANK,SSAN,OSDSD,AFSC
(SORTED IN ASCENDING SEQUENCE
ON SSAN)

EXHIBIT MDQ050-5

PROBLEM FLOW DIAGRAMS

RETRIEVE
COLLECT FOR SORT
SORT
PRINT
RETRIEVE
COLLECT FOR SORT
SORT
PRINT

RETRIEVE
COLLECT FOR SORT
SORT
PRINT

RETRIEVE
COLLECT FOR SORT
SORT
RETRIEVE
COLLECT FOR SORT
SORT
.
.
.
.
RETRIEVE
COLLECT FOR SORT
SORT
PRINT

RETRIEVE
COLLECT FOR SORT
RETRIEVE
COLLECT FOR SORT
.
.
.
RETRIEVE
COLLECT FOR SORT
SORT
RETURN FROM COLLECTION FILE
PRINT
RETURN FROM COLLECTION FILE
PRINT
.
.
.
RETURN FROM COLLECTION FILE
PRINT

EXHIBIT MDQ050-6

NESTED WRAP-UP STATEMENTS

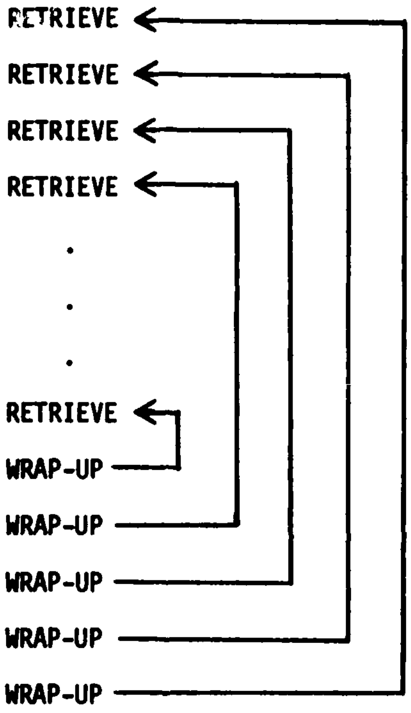


EXHIBIT MDQ050-7

QPL REPORT STATEMENT

```
REPORT report-name
[         { PRINTER [ {DUPLICATE}       n ]                                  }]
[         {           {DUP      }                                            }]
[         {                                                                  }]
[         { REMOTE id {DUPLICATE}  n   [,id {DUPLICATE}  n] . . .            }]
[  ON     {           {DUP      }           {DUP      }                      }]
[         { FILE catalog/file-string    [,catalog/file-string] . . .         }]
[         {                                                                  }]
[         {    FOR          {TTY                    }                        }]
[         {                 {CRT                    }                        }]
[         {                 {PRINTER   [IN BCD]     }                        }]
[SCC IS literal]

[ {COLUMN}  SPACING IS nn ]
[ {COL   }                ]

[PAGE LENGTH nnn LINES]

[PAGE WIDTH nnn CHARACTERS]

[NUMBER OF PAGES nnnnn]

[{[STARTING PAGE NUMBER nnnnn][TOTAL PAGES {nnnnn  }]}]
[{                                         {literal}]}]
[{                                                   }]
[{[NO PAGE NUMBER]                                   }]
[COVER PAGE IS label      [,label] . . .]
[PAGE HEADING IS label      [,label] . . .]
[PAGE FOOTING IS label      [,label] . . .]
[CONTROL HEADING FOR label-2 IS label-2    [,label-n]. . .
[
         BREAK ON ||identifier-1  [,identifier-2] . . . CHANGE ||]
                  ||FIRST                                      ||]
[CONTROL FOOTING FOR label-3 IS label-4  [,label-n]. . .
[
         BREAK ON ||identifier-3  [,identifier-4]. . . CHANGE ||]
                  ||LAST                                      ||]
```

EXHIBIT MDQ060-1

EPL REPORT STATEMENT

```
REPORT report-name
           PRINTER  [{DUPLICATE}     n]
                    [{DUP      }      ]

           REMOTE id{DUPLICATE}     n   [,id {DUPLICATE}    n] . . .
                    {DUP      }         [    {DUP      }     ]

 ON    {FILE[{REMOVABLE}]catalog/file-string [,[{REMOVABLE}]catalog/file-string  ...]}
       {    [{RMVBL    }]                    [ [{RMVBL    }]                        ]}

              [FOR      {TTY                      }]
              [         {CRT                      }]
              [         {PRINTER   [IN BCD]       }]
              [         {terminal-name            }]
           DAC
           TAPE "tape-label"

  [SCC IS literal]

 [{COLUMN}  SPACING IS nn]
 [{COL   }               ]

  [PAGE LENGTH nnn LINES]

  [PAGE WIDTH nnn CHARACTERS]

  [NUMBER OF PAGES nnnnn]

[{[STARTING PAGE NUMBER nnnnn][TOTAL PAGES {nnnnn  }]}]
[{                                         {literal}  }]
[{[NO PAGE NUMBER]                                    }]

 [COVER PAGE IS label        [,label] . . .]

 [PAGE HEADING IS label      [,label] . . .]

 [PAGE FOOTING IS label      [,label] . . .]

 [CONTROL HEADING FOR label-2 IS label-2    [,label-n]. . .

           BREAK ON ||identifier-1  [,identifier-2] . . . CHANGE||]
                    ||FIRST                                     ||

 [CONTROL FOOTING FOR label-3 IS label-4  [,label-r]. . .

           BREAK ON ||identifier-3  [,identifier-4]. . . CHANGE ||]
                    ||LAST                                      ||
```

EXHIBIT MDQ060-2

```
010 REPORT OFFICER-LIST
020         ON PRINTER
030         COLUMN SPACING IS 5
040         PAGE LENGTH 25 LINES
050         PAGE WIDTH 100 CHARACTERS
060         NUMBER OF PAGES 25
070         STARTING PAGE NUMBER 2
080         TOTAL PAGES
090         COVER PAGE IS CVR
100         PAGE HEADING IS PG-HDG
110         PAGE FOOTING IS PG-FTG
120 CVR.    SPACE 20
130         LINE "OFFICER LISTING" COL 35
140         SPACE 5
150         LINE "AS OF" COL 35, $DATE COL 41
160 PG-HDG. LINE "OFFICER LIST AS OF" COL 30, $DATE
170         SPACE 2
180 PG-FTG. SPACE 2
190         LINE SPACE-NUMBER COL 50, "OF", $TOTAL-PAGES, "PAGES"
200 L1.LINE RANK GROUP INDICATE 1, NAME, SSAN, DOR
210 QUERY MCAI/WWDMS/OADF WHERE RANK CONTAINS "O"
220 SORT ON RANK DSC, DOR, NAME
230 PRINT L1
240 END
```

EXHIBIT MDQ060-3

LINE STATEMENT

Part 1

$$\underline{\text{LINE}}\ \text{report-elem-descr}\quad [\text{,report-elem-descr}]\ldots$$

QPL LINE STATEMENT

Part 2

$$\underline{\text{LINE}}\ \left\{\begin{array}{l}\text{report-elem-descr}\\ \text{entry-name}\end{array}\right\}\quad [\text{,report-elem-descr}]\ldots$$

EPL LINE STATEMENT

EXHIBIT MDQ060-4

01	F	SMITH SUSAN T
	M	COLLINS PAUL
		VON BUREN CHARLES E
		GEORGE ARNOLD T
		BARNES SONNY
		HANSON ROBERT
02	F	COLLINS MINNIE
	M	HARRISON DONALD T
		SMITH THOMAS B
	F	GEORGE JANICE A
		BLACK TOOTSIE
		WHITE SANDY
	M	BARROWS BRENT W
	F	KNOX WENDY J
	M	BROWN STEPHEN
		DELAND JASON E
03	M	ANDERSON DONALD
	F	ANDERSON BARBARA
	M	COLLINS MICKEY
		LELAND PAUL T
		BLACK HERB
	F	JOHNSON LINDA S
	M	SCABBARD WILLIAM
	F	LEONARD KAREN D
	M	JOHNSON JOHN S
		TATE WILLIAM Z
04	F	ALLEN PATTY
	M	RODGERS RAYMOND J
		MATSON MARCUS B
		BATES PETER
		GREENTREE GARY
05	M	BAKER SCROOGE
	F	JACKSON BETTY C
	M	ALBERT DWIGHT R
06	M	CARR JULIUS
		DAVIS JOHN Q
		GAUSS ADAM E
	F	SMITH LUCY R
07	F	CARR CORA
	M	FARMER AMOS T

EXHIBIT MDQ060-5

01	F	SMITH SUSAN T
		COLLINS PAUL
		VON BUREN CHARLES E
		GEORGE ARNOLD T
		BARNES SONNY
		HANSON ROBERT
02	F	COLLINS MINNIE
		HARRISON DONALD T
		SMITH THOMAS B
		GEORGE JANICE A
		BLACK TOOTSIE
		WHITE SANDY
		BARROWS BRENT W
		KNOX WENDY J
		BROWN STEPHEN
		DELAND JASON E
03	M	ANDERSON DONALD
		ANDERSON BARBARA
		COLLINS MICKEY
		LELAND PAUL T
		BLACK HERB
		JOHNSON LINDA S
		SCABBARD WILLIAM
		LEONARD KAREN D
		JOHNSON JOHN S
		TATE WILLIAM Z
04	F	ALLEN PATTY
		RODGERS RAYMOND J
		MATSON MARCUS B
		BATES PETER
		GREENTREE GARY
05	M	BAKER SCROOGE
		JACKSON BETTY C
		ALBERT DWIGHT R
06	M	CARR JULIUS
		DAVIS JOHN Q
		GAUSS ADAM E
		SMITH LUCY R
07	F	CARR CORA

EXHIBIT MDQ060-6

```
01          F  SMITH SUSAN T
01          M  COLLINS PAUL
01             VON BUREN CHARLES E
01             GEORGE ARNOLD T
01             BARNES SONNY
01             HANSON ROBERT
02          F  COLLINS MINNIE
02          M  HARRISON DONALD T
02             SMITH THOMAS B
02          F  GEORGE JANICE A
02             BLACK TOOTSIE
02             WHITE SANDY
02          M  BARROWS BRENT W
02          F  KNOX WENDY J
02          M  BROWN STEPHEN
02             DELAND JASON E
03          M  ANDERSON DONALD
03          F  ANDERSON BARBARA
03          M  COLLINS MICKEY
03             LELAND PAUL T
03             BLACK HERB
03          F  JOHNSON LINDA S
03          M  SCABBARD WILLIAM
03          F  LEONARD KAREN D
03          M  JOHNSON JOHN S
03             TATE WILLIAM Z
04          F  ALLEN PATTY
04          M  RODGERS RAYMOND J
04             MATSON MARCUS B
04             BATES PETER
04             GREENTREE GARY
05          M  BAKER SCROOGE
05          F  JACKSON BETTY C
05          M  ALBERT DWIGHT R
06          M  CARR JULIUS
06             DAVIS JOHN Q
06             GAUSS ADAM E
06          F  SMITH LUCY R
07          F  CARR CORA
07          M  FARMER AMOS T
```

EXHIBIT MDQ060-7

SPACE STATEMENT

```
         ( nn  )
SPACE   <      >
         ( TOP )
```

EXHIBIT MDQ060-8

CONTROL HEADING CLAUSE

```
┌
│ CONTROL HEADING FOR label-1  IS  label-2  [,label-n]...
│                                       ││ identifier-1   [,identifier-2]...CHANGE ││┐
│                                       ││                                          │││
└                       BREAK ON        ││                                          │││
                                        ││                                          │││
                                        ││ FIRST                                    │││
                                                                                      ┘
```

EXHIBIT MDQ070-1

```
010 REPORT PRT ON REMOTE R2
020     PAGE HEADING IS PH
030     PAGE FOOTING IS PF
040     CONTROL HEADING FOR PL  IS CH BREAK ON RANK CHANGE, FIRST
050     CONTROL FOOTING FOR PL  IS CF BREAK ON RANK CHANGE, LAST
060 PH. LINE "THIS IS THE NORMAL HEADING"
070     SPACE 3
080 PF. SPACE 3
090     LINE "THIS IS THE NORMAL FOOTING"
100 CH. LINF "THIS IS THE NORMAL FOOTING"
110 CF. LINE "THIS IS THE NORMAL FOOTING"
120 PL. LINE RANK, NAME, SEX, MS, SSAN
130 QUERY MCAI/WWDMS/OADF WHERE RANK = "05" OR "06" OR "07"
140 SORT ON RANK, NAME
150 PRINT PL
160 END
```

EXHIBIT MDQ070-2

```
THIS IS THE NORMAL HEADING


THIS IS THE CONTROL HEADING
05  ALBERT DWIGHT R      M  M  326483721
05  BAKER SCROOGE        M  W  207102244
05  JACKSON BETTY C      F  M  209127476
THIS IS THE CONTROL FOOTING
THIS IS THE CONTROL HEADING
06  CARR JULIUS          M  M  200041111
06  DAVIS JOHN Q         M  M  206841522
06  GAUSS ADAM E         M  M  245657214
06  SMITH LUCY R         F  S  587223344
THIS IS THE CONTROL FOOTING
THIS IS THE CONTROL HEADING
07  CARR CORA            F  M  200046677
07  FARMER AMOS T        M  M  242321234
THIS IS THE CONTROL FOOTING

THIS IS THE NORMAL FOOTING
```

EXHIBIT MDQ070-3

CONTROL FOOTING CLAUSE

```
[
  CONTROL FOOTING FOR label-3  IS  label-4  [,label-n]...
                            ||identifier-3   [,identifier-4]...CHANGE||]
              BREAK ON      ||                                       ||
]                           ||LAST                                   ||]
```

EXHIBIT MDQ070-4

STATISTICAL FUNCTIONS CLAUSE

```
⎡⎧ SUM     ⎫                                 ⎤
⎢⎪ COUNT   ⎪                                 ⎥
⎢⎪ STD DEV ⎪                                 ⎥
⎢⎨ AVERAGE ⎬    identifier     [NO RESET]    ⎥
⎢⎪ MINIMUM ⎪                                 ⎥
⎣⎩ MAXIMUM ⎭                                 ⎦
```

EXHIBIT MDQ070-5

```
E5   691228   PETERS PATRICA F      452415422   M   F   1
E5   700615   FRANKLIN ALEXANDER    206982256   D   M   2
E5   711231   COOKE CHARLES E       205157882   M   M   3
E5   720328   SMITH LARRY T         58724555    S   M   0
E5   731022   ADAMS JOHN E          15436095    S   M   0
E5   740812   WATSON HARRY          21738001    S   M   0
E5   740812   WATSON GARY           21738002    S   M   0
E5   740812   WATSON SAM            21738003    S   M   0
E5   740812   WATSON DANNY          21738004    S   M   0
E5   740812   WATSON DONALD         21738005    S   M   0
E5   740812   WATSON STEVEN         21738006    S   M   0
E5   740812   WATSON BOB            21738007    S   M   0
E5   740814   WALTERS PHILIP S      422721549   M   M   2
E5   741201   LANE DONALD           205727654   M   M   1
E5   760814   BROWN SANDY           264550001   S   F   0
E5   761125   ANDERSON ALEXANDER    11892888    S   M   0

TOTAL IN THIS RANK IS      16
```

EXHIBIT MDQ070-6

COUNT AND SUM FUNCTIONS

Part 1

```
COUNT identifier-1

   ⎡RESET ON identifier-2  CHANGE
   ⎣
         [ENABLING control-identifier]⎤
                                      ⎦
```

Part 2

```
        ⎧identifier-1              ⎫
   SUM  ⎨                          ⎬
        ⎩identifier-expression     ⎭

   ⎡RESET ON identifier-2  CHANGE
   ⎣
         [ENABLING control-identifier]⎤
                                      ⎦
```

EXHIBIT MDQ070-7

CONTROL and ENABLE STATEMENTS

```
CONTROL  control-identifier

ENABLE control-identifier ⎡ON identifier [,identifier]...CHANGE⎤
                          ⎣                                     ⎦
```

EXHIBIT MDQMDQ070-8

```
010 INVOKE MCAI/WWDMS/OADF
020 REPORT RPT ON REMOTE R1
030 PL. LINE NAME TITLE, RANK TITLE, AFSC TITLE
040 ST. LINE "WE HAVE", $CNT-AFSC, "PEOPLE IN THE ABOVE AFSC"
050 RETRIEVE PERSON WHERE AFSC PRESENT
060 SORT PERSON ON AFSC, NAME
070 LET $CNT-AFSC = COUNT NAME RESET ON AFSC CHANGE
080      ENABLING PRT-SUB-TOT
090 PRINT PL
100 CONTROL PRT-SUB-TOT
110 PRINT ST
120 END
```

EXHIBIT MDQ070-9

```
NAME                                  RANK     AFSC
BROWN CHARLES F                       E1       51010
BROWN WILLIAM B                       E2       51010
FORD BERTHA                           E1       51010
KENNEDY MARSHA E                      E1       51010
MARTIN CARL W                         E2       51010
WILLIAMS PETER E                      E1       51010
WILLIAMS WANDA A                      E2       51010
WE HAVE       7.000000 PEOPLE IN THE ABOVE AFSC
DAVID RICHARD Q                       E3       51130
SMITH TIMOTHY A                       E3       51130
WE HAVE       2.000000 PEOPLE IN THE ABOVE AFSC
ANDERSON COOKIE                       E3       51131
BLACK OTTO V                          E3       51131
CLARK GARY                            E3       51131
DANIELS DONALD M                      E3       51131
FRANKS BARBARA                        E1       51131
LANE DAISY                            E4       51131
RODGERS RAYMOND F                     E3       51131
WE HAVE       7.000000 PEOPLE IN THE ABOVE AFSC
COOKE CHARLES E                       E5       51132
WILSON ELIZABETH T                    E4       51132
WE HAVE       2.000000 PEOPLE IN THE ABOVE AFSC
ADAMS JOHN E                          E5       51150
FRANKLIN ALEXANDER                    E5       51150
SMITH LARRY T                         E5       51150
WALTERS PHILIP S                      E5       51150
WE HAVE       4.000000 PEOPLE IN THE ABOVE AFSC
ANDERSON ALEXANDER                    E5       51151
PETERS PATRICA F                      E5       51151
TYLER QUINCY E                        E4       51151
WATSON BOB                            E5       51151
WATSON DANNY                          E5       51151
WATSON DONALD                         E5       51151
WATSON GARY                           E5       51151
WATSON HARRY                          E5       51151
WATSON SAM                            E5       51151
WATSON STEVEN                         E5       51151
WE HAVE      10.000000 PEOPLE IN THE ABOVE AFSC
BAKER SCROOGE                         05       5116
CARR CORA                             07       5116
CARR JULIUS                           06       5116
DAVIS JOHN Q                          06       5116
FARMER AMOS T                         07       5116
GAUSS ADAM E                          06       5116
SMITH LUCY R                          06       5116
WE HAVE       7.000000 PEOPLE IN THE ABOVE AFSC
BROWN SANDY                           E5       51171
BUCHANAN WILSON A                     E6       51171
GREEN GEORGE                          E6       51171
LANE DONALD                           E5       51171
LUTHER JAMES A                        E7       51171
WE HAVE       5.000000 PEOPLE IN THE ABOVE AFSC
GARDNER TOM Q                         E6       51172
```

EXHIBIT MDQ070-10

EXAMPLE ADF

```
ADF OBJECT FILE IS UMC/SUBCAT/OADF
DATA BASE IS WBU-DBR
     ENTRY IS PERSON
          ITEMS ARE
               ID-1
               KEY-1
               SSAN
               RANK
               NAME
               AFSC
               DOB
               SEX
               MS
               NOD
               DOR
               DEROS
               TAFCSD
               TAFMSD
               ADSCD
     ENTRY IS ASSIGNMENT
          ITEMS ARE
               ID-1
               KEY-1
               SSAN
               RANK
               NAME
               BASE
               COMMAND
               ARRIVAL-DATE
               DEPARTURE-DATE
```

EXHIBIT MDQ080-1

INDEXED SEQUENTIAL PROCESSOR (ISP) (H6000-CDT) COURSE EXHIBITS

ISP010 - ISP Concepts, Capabilities, and Formats

ISP is concerned with two access methods:

linked list and index organization.

There are six different formats with ISP:

Data File, Data Page, Data Record Formats,
Index File, Index Page, Index Record Formats.

ISP020 - ISP Data Base Load

There are four ISP subroutines concerned with loading an ISP data base:

NOPEN - opens a new ISP data base.

FILINT - writes a new ISP data record and possible index record to the ISP data base.

ICLOSE - closes a specific ISP data base.

WRAPUP - closes all ISP data bases.

ISP030 - ISP Data Base Retrieval

There are two ISP data base retrieval methods:

sequential and random.

IOPEN - opens an existing ISP data base.

GETSEQ - sequentially retrieves ISP records.

GETRAN - randomly retrieves ISP records.

ISP040 - ISP Data Base Modification

There are three ISP subroutines to modify an existing ISP data base:

ADREC - adds new ISP records to an ISP data base.

DELREC - deletes existing ISP records from an ISP data base.

CHGREC - changes field within an ISP record or an ISP data base.

ISP050 - ISP Utilities

EXHIBIT ISP010-1

ISP050 - ISP Utilities

There are three miscellaneous ISP subroutines:

XREWND - repositions the current record pointer within an ISP data base.

XSTAT - obtains ISP data base statistics.

XFLUSH - writes ISP record buffer contents to an ISP data base.

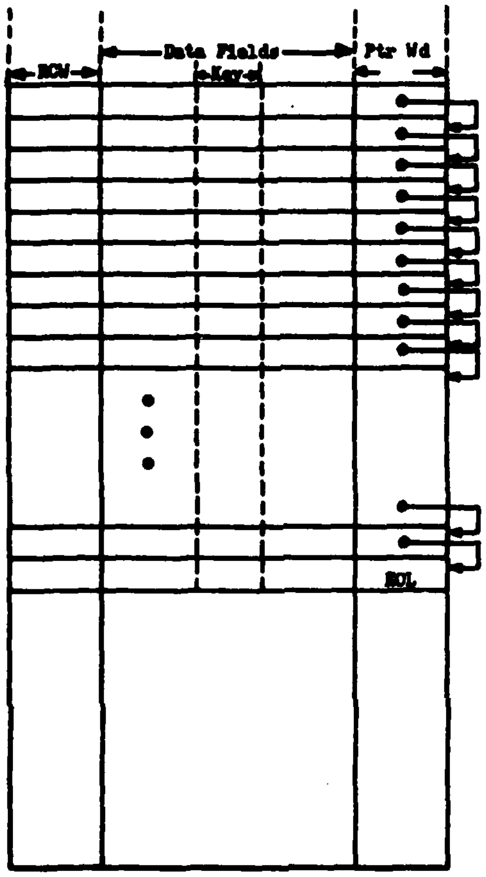
There are two ISP utilities:

XUTIL - reorganizes an existing ISP data base removing logically deleted records and replacing overflow records into physical sequence.

XJRNAL - reconstructs a destroyed ISP data base from a journal tape.

EXHIBIT ISP010-1 (Cont)

LINKED SEQUENTIAL ORGANIZATION



Initial Load

RCW - Record Control Word (Contains Logical Delete Switch)

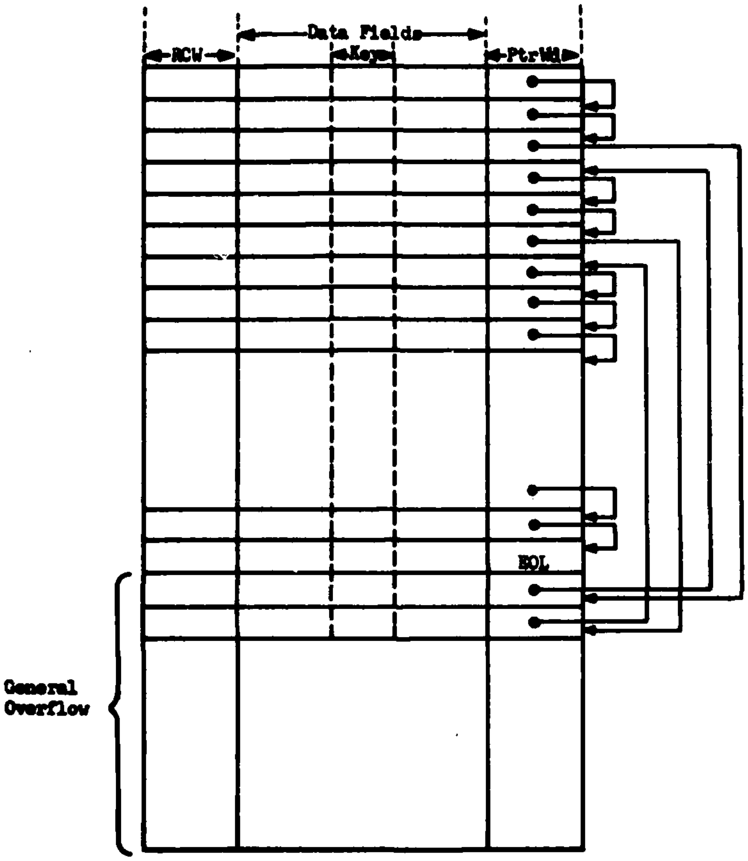
Ptr Wd - Pointer Word (Address of Next Record in Logical Sequence)

Key - Data Field on which Logical Sequence is based.

EOL - end of list

EXHIBIT ISP010-2

LINKED SEQUENTIAL ORGANIZATION



After Update

EXHIBIT ISP010-3

COARSE INDEX

	FINE PAGE #
500	1
1000	2
1500	3
2000	4

FINE INDEX
PAGE 4

KEY	DATA PAGE
1600	5
1700	6
1800	7
1900	8
2000	9

In the FINE INDEX, 2000/9 is the low value for page 10 and 1900/8 is the low value for page 9.

EXHIBIT ISP010-4

INDEXED ORGANIZATION

INDEX FILE
(Coarse and Fine)

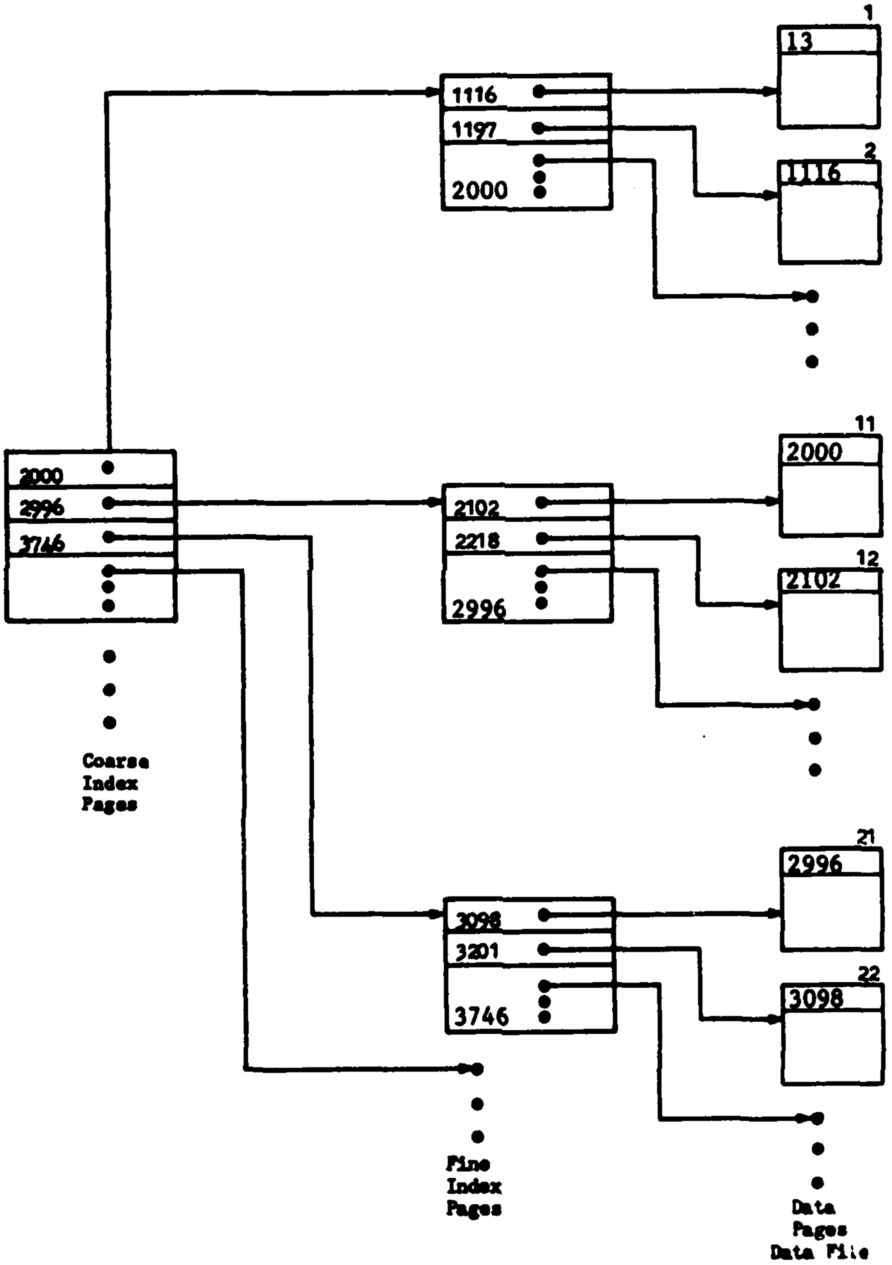


EXHIBIT ISP010-5

DATA FILE FORMAT

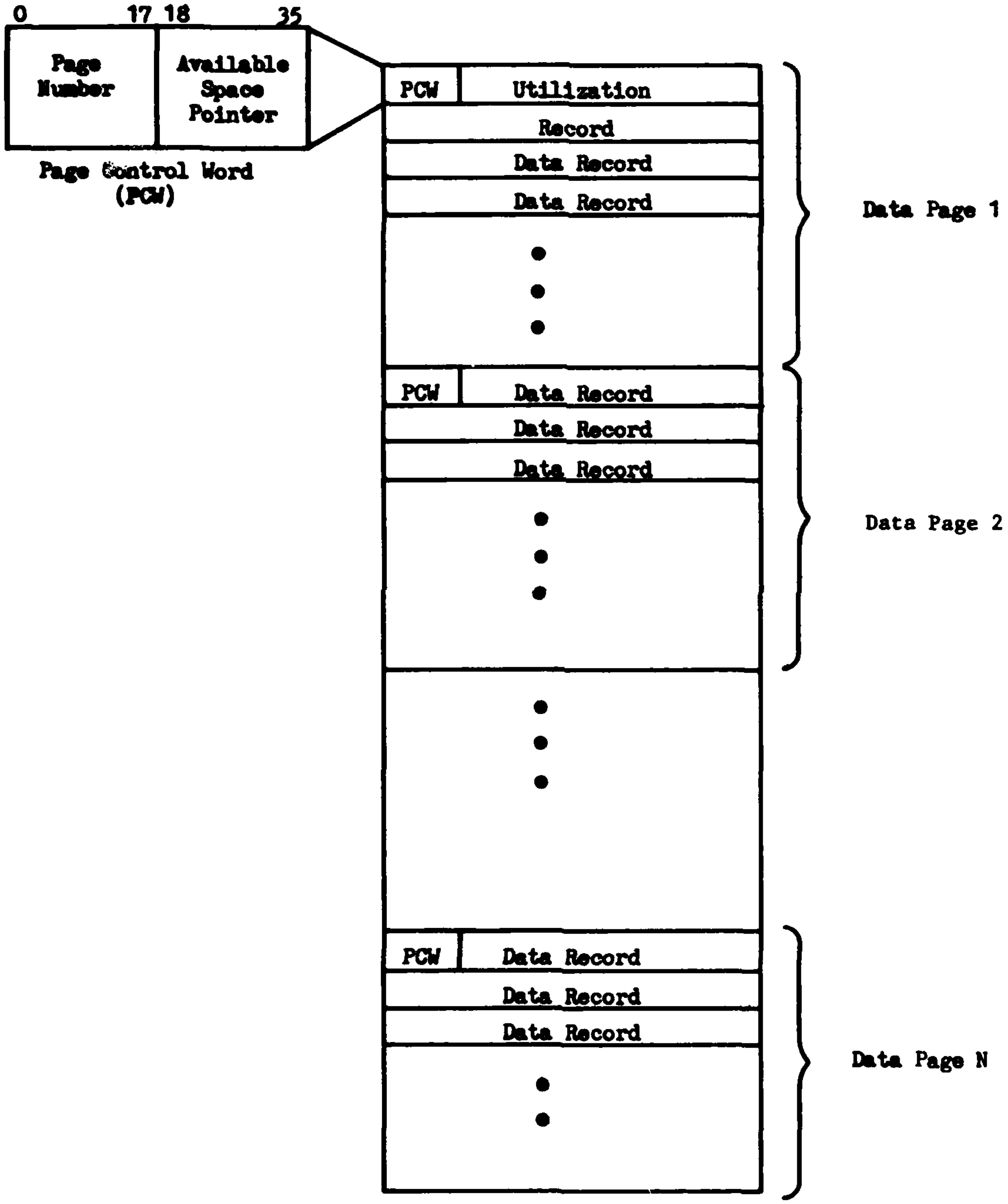
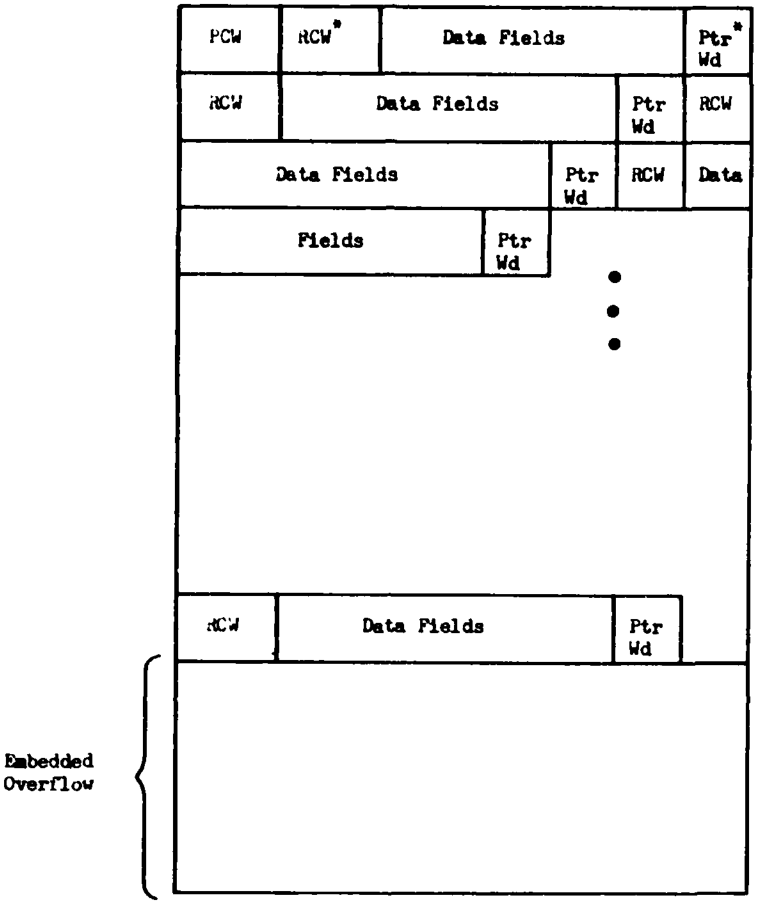


EXHIBIT ISP010-6

DATA PAGE FORMAT



*See Exhibit ISP010-8 for the format of the RCW (Record Control Word) and Ptr Wd (Pointer Word).

EXHIBIT ISP010-7

DATA RECORD FORMAT

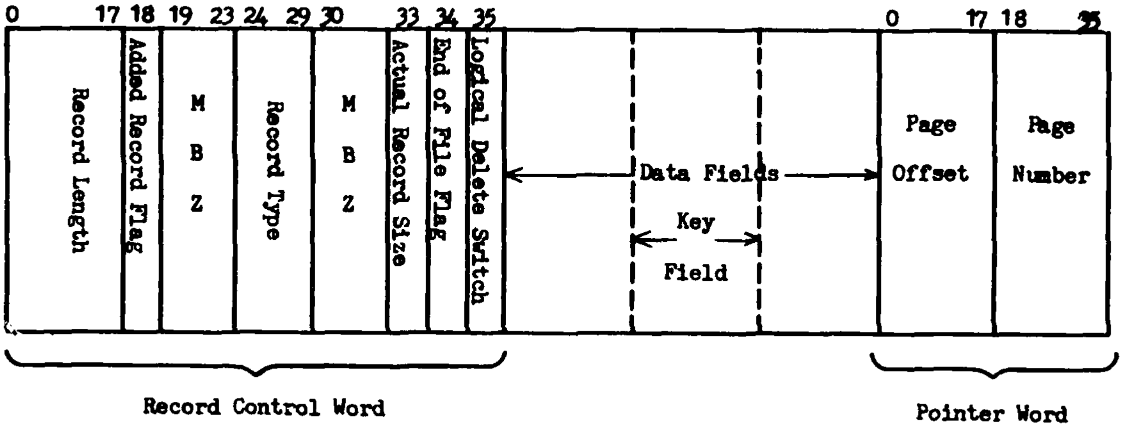


EXHIBIT ISP010-8

INDEX FILE FORMAT

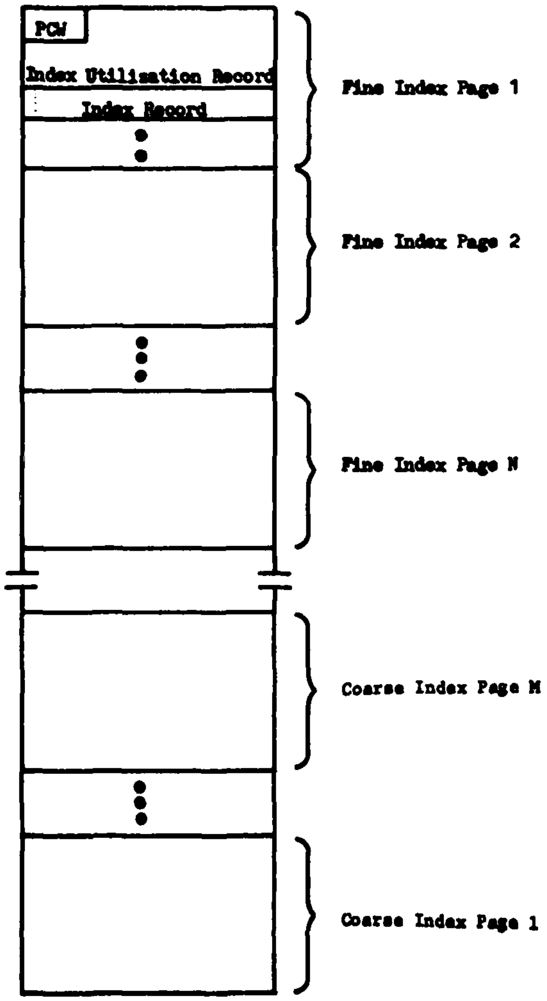


EXHIBIT ISP010-9

INDEX PAGE FORMAT

Page Control Word		
Key Field	M B Z	Index Pointer
Key Field	M B Z	Index Pointer
● ● ●		

EXHIBIT ISP010-10

INDEX RECORD FORMAT

	0 17	18 35
Key Field	M B Z	Fine Index Page Pointer

Coarse Index Record

	0 17	18 35
Key Field	M B Z	Data Page Pointer

Fine Index Record

EXHIBIT ISP010-11

```
1      8         16
$      IDENT
$      USERID
$      FILSYS
       FCREAT  DATA-FILE CAT/FILE STRING,BLOCKS/10,10/,MODE/RAND/,FCLASS/UZZ/
       FCREAT  INDEX-FILE CAT/FILE STRING,BLOCKS/2,2/,MODE/RAND/,FCLASS/UZZ/

$      COBOL   CBL68
       IDENTIFICATION DIVISION.
       PROGRAM-ID.  XHBT20
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. 6000-EIS.
       OBJECT-COMPUTER. 6000-EIS.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
            SELECT IN-FIL ASSIGN TO IN.

    **    ISP DATA AND INDEX FILES DO NOT REQUIRE SELECT STATEMENTS.

       I-O-CONTROL.
            APPLY STANDARD ON IN-FIL.

    **    SYSTEM STANDARD FORMAT IS NOT APPLIED TO ISP DATA AND INDEX FILES.

       DATA DIVISION.
       FILE SECTION.
       FD IN-FIL      LABEL RECORDS ARE STANDARD.
       01 INVENTORY.
           03    PART-NUMBER          PIC X(5).
           03    PART-NAME            PIC X(20).
           03    PRICE                PIC 9(5)V99.
           03    QTY-ON-HAND          PIC 9(5).
           03    BIN-LOC              PIC X(8).
       WORKING-STORAGE SECTION.

    **    PARAMETERS USED BY THE ISP SUBROUTINES ARE NORMALLY
    **    DESCRIBED IN THE WORKING-STORAGE SECTION AS 77 LEVEL OR
    **    01 LEVEL ITEMS.

       77  IFC                        PIC X(2)             VALUE "I1".
       77  DFC                        PIC X(2)             VALUE "D1".
       77  REC-SZ                     PIC 9(6)   COMP-1    VALUE 8.
       77  KEY-OFF                    PIC 9(6)   COMP-1    VALUE 0.
       77  KEY-SZ                     PIC 9(6)   COMP-1    VALUE 5.
       77  PCT-FILL                   PIC 9(6)   COMP-1    VALUE 75.
       77  D-PG-SZ                    PIC 9(6)   COMP-1    VALUE 384.
       77  COL-COM                    PIC X(2)             VALUE "CC".
       77  ALT-REC-SZ                 PIC 9(6)   COMP-1.
       77  ERR                        PIC X(3)   SYNC RIGHT.
           88    OKAY                                      VALUE "000".
       77  SEQ-CK                     PIC X(12).
       77  EOF-FLAG                   PIC 9(1).
```

EXHIBIT ISP020-1

```
     PROCEDURE DIVISION.
     000-DRIVER SECTION.
     010-DRIVER.
         PERFORM 110-OPEN.
         PERFORM 210-READ.
         MOVE 0 TO EOF-FLAG.
         PERFORM 310-CHECK UNTIL EOF-FLAG = 1.
         PERFORM 410-WRAP-UP.
         STOP RUN.
     100-OPEN SECTION.
     110-OPEN.
         OPEN INPUT IN-FIL.
         CALL NOPEN USING IFC, DFC, INVENTORY, REC-SZ, KEY-OFF,
              KEY-SZ, PCT-FILL, D-PG-SZ, COL-COM.

   **   NOTICE THAT THE RECORD DELIVERY AREA (INVENTORY) IS AN
   **   01 LEVEL ITEM IN THE FILE SECTION.  IT COULD ALSO BE IN
   **   THE WORKING-STORAGE SECTION.

     200-READ SECTION.
     210-READ.
         READ IN-FIL AT END MOVE 1 TO EOF-FLAG.
     300-CHECK SECTION.
     310-CHECK.
         CALL FILINT USING DFC, ALT-REC-SZ,ERR,SEQ-CK.
         IF NOT OKAY DISPLAY "LOAD ERROR".
         PERFORM 210-READ.
     400-WRAP-UP SECTION.
     410-WRAP-UP.
         DISPLAY "LOAD COMPLETE".
         CLOSE IN-FIL.
         CALL ICLOSE USING DFC.

   **     NOTICE THAT THE ONLY ISP SUBROUTINES WHICH CAN FOLLOW A
   **     NOPEN ARE THE FILINT, ICLOSE, OR WRAPUP SUBROUTINES.

$       EXECUTE
$       PRMFL           D1,R/W,R,DATA FILE CATA/FILE STRING
$       PRMFL           I1,R/W,R,INDEX FILE CAT/FILE STRING
$       PRMFL           IN,R,S,CDTSCDTS/ISP/DATA020
$       ENDJOB
***EOF
```

EXHIBIT ISP020-1 (Cont)

NOPEN SUBROUTINE

Format 1

CALL NOPEN USING record-area, data-file-code-1, desc-file-code [, error-code].

Format 2

CALL NOPEN USING index-file-code, data-file-code-1, record area, record-size, key-offset, key-size [, pct-fill [, data-page-size [, coll-comm]]].

EXHIBIT ISP020-2

FILINT SUBROUTINE

CALL FILINT USING data-file-code-1

[, alt-record-size [, error-code, seq-check]] .

EXHIBIT ISP020-3

ICLOSE SUBROUTINE

CALL ICLOSE USING data-file-code-1.

EXHIBIT ISP020-4

WRAPUP SUBROUTINE

CALL WRAPUP.

EXHIBIT ISP020-5

Error Code	Subroutine	Meaning
000	ALL	No error encountered.
X02	CHGREC DELREC GETRAN XREWND	A record having a specified key value does not exist in the file.
X03	FILINT	During file initialization, a record was encountered with a key value less than a key value already stored in the file.
X04	GETSEQ XREWND	During sequential processing, the ISP end-of-file record was encountered.
X07	ADDREC	The key value of a new record that is to be added to the file already exists in the file.
X09	NOPEN IOPEN	Either the index or the primary data file is not present.
X10	NOPEN IOPEN	Either the index or the primary data file is not a random file.
X11	IOPEN	The index and the primary data file do not match; that is, they were not initialized at the same time.
X12	NOPEN IOPEN	Either the index or the primary data file is already open.

EXHIBIT ISP020-6

DATA BASE LOAD PROBLEM

1. Create a random file for the ISP data file and another for the ISP index file, using either FILSYS or ACCESS.

The following steps demonstrate a procedure to approximate the data file size (assuming 100% fill):

a. Determine the page size in words, rounding up to the nearest 64 word increment (minimum size is 320, maximum is 4032).

Page size = 320 words

b. Compute the data record size of the largest record.

Record size = # words in largest record + two ISP control words.
Record size = 14 words + 2 words = 16 words

c. Compute the number of data records per page, ignoring the remainder.

Number of data records per page = (page size - 2)/(record size)
Number of data records per page = (320 - 2)/16 = 19 records/page

d. Compute the number of pages needed for the data file, rounding up.

Number of pages = (# records in file)/(# records/page)
Number of pages = (100 records)/(19 records/page) = 5.26
Number of pages = 6 pages

e. Compute the number of LLinks needed, rounding up.

Number of LLinks = (# of pages)*(# words/page)/320
Number of LLinks = (6*320)/320 = 6
Number of LLinks = 6 LLinks

HINT: The data file should be 7 LLinks for student problem to allow for overflow records.

The following steps demonstrate a procedure to approximate the index file size:

a. Determine the number of data pages using the above steps.

Number of data pages = 6 pages

b. Calculate the number of fine index pages, again rounding up.

Number of fine index pages = ((# data pages)*(index record size in words))/(index page size)
Number of fine index pages = (6*3)/320 = .05
Number of fine index pages = 1 page

EXHIBIT ISP020-7

c. Calculate the number of coarse index pages.

Number of coarse index pages = ((# fine index pages)*
(index record size in words))/index page size.
Number of coarse index pages = (1*3)/320 = 0
Number of coarse index pages = 0 pages

(NOTE: Coarse index pages are only created when more than one fine index page exists.)

d. Compute the number of LLinks needed, rounding up.

Number of LLinks = ((# fine + coarse index pages)*(page size))/320
Number of LLinks = (1+0)*(320)/320 = 1
Number of LLinks = 1 LLink

HINT: The index file should be 1 LLink for the student problem.

2. Given a sequential file sorted on a key field (SSAN) in ascending order, use the appropriate subroutines to initialize and load the index and data files. The input will be obtained from the file located at the catalog/file string:

CDTSCDTS/ISP/DATA020

The format of the input records is:

FIELD		STARTING COLUMN	ENDING COLUMN
SSAN		2	10
RANK		13	14
NAME		15	34
AFSC	(Specialty Code)	35	40
DOB	(Date of Birth)	41	46
SEX		47	47
MS	(Marital Status)	48	48
NOD	(Number of Dependents)	49	50
DOR	(Date of Rank)	51	56
DEROS	Date Estimated Return from O/S	57	62
TAFCSD	Total Active Federal Commissioned Service Date)	63	68
TAFMSC	(Total Active Federal Military Service Date)	69	74
ADSCD	(Active Duty Service Commitment Date)	75	80

NOTE: Use 100% fill and a data and index page size of 320 words.

Use the following JCL for your job:

```
$       IDENT     . . .
$       USERID    . . .
$       COBOL     CBL68
          .
          .   YOUR ISP LOAD PROGRAM
          .
$       EXECUTE
$       PRMFL     IN,R,S,CDTSCDTS/ISP/DATA020
$       PRMFL     D1,W,R,data-file cat/file string
$       PRMFL     I1,W,R,index-file cat/file string
$       ENDJOB
***EOF
```

NOTE: Be sure to check and report any errors which are encountered during the data base loading.

WARNING: This problem does not require sorting of the input data. But, if you do encounter a problem requiring both sorting and ISP data base loading, ISP core buffers must be placed in labeled common and a special $ LIMITS card must be used. This is to keep ISP and COBOL SORT/MERGE from using the same core area for their buffers. This is discussed in your ISP manual.

EXHIBIT ISP020-7 (Cont)

```
1      8
$      IDENT
$      USERID
$      COBOL   CBL68
       IDENTIFICATION DIVISION.
       PROGRAM-ID.  XHBT31.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. 6000-EIS.
       OBJECT-COMPUTER.  6000-EIS.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
          SELECT OT-FIL ASSIGN TO OT-LISTING.
       I-O-CONTROL.
          APPLY STANDARD ON OT-FIL.
       DATA DIVISION.
       FILE SECTION.
       FD  OT-FIL LABEL RECORDS ARE STANDARD.
       01 INVENTORY.
          03   PART-NUMBER-OUT     PIC    X(5).
          03   FILLER              PIC    X(2).
          03   PART-NAME-OUT       PIC    X(20).
          03   FILLER              PIC    X(2).
          03   PRICE-OUT           PIC    Z(5)9.99.
          03   FILLER              PIC    X(2).
          03   QTY-ON-HAND-OUT     PIC    Z(5).
          03   FILLER              PIC    X(2).
          03   BIN-LOC-OUT         PIC    X(8).
       WORKING-STORAGE SECTION.
       77  IFC                     PIC    X(2)                VALUE "I1".
       77  DFC                     PIC    X(2)                VALUE "D1".
       77  REC-SZ                  PIC    9(6)     COMP-1     VALUE 8.
       77  KEY-OFF                 PIC    9(6)     COMP-1     VALUE 0.
       77  KEY-SZ                  PIC    9(6)     COMP-1     VALUE 5.
       77  ERR                     PIC    X(3)                SYNC RIGHT.
           88     EOF                                         VALUE "X04".
       77  EOF-FLAG                PIC    9(1).

    **    NOTICE THAT AN "X04" IN ERR MEANS AN END-OF-FILE
    **    RECORD WAS ENCOUNTERED BY THE GETSEQ SUBROUTINE.

       01    REC-AREA.
             03  PART-NUMBER        PIC    X(5).
             03  PART-NAME          PIC    X(20).
             03  PRICE              PIC    9(5)V99.
             03  QTY-ON-HAND        PIC    9(5).
             03  BIN-LOC            PIC    X(8).
```

EXHIBIT ISP030-1

```
       PROCEDURE DIVISION.
       000-DRIVER SECTION.
       010-DRIVER.
           PERFORM 110-OPEN.
           PERFORM 210-RETRIEVE.
           MOVE 0 TO EOF-FLAG.
           PERFORM 310-WRITE UNTIL EOF-FLAG = 1.
           PERFORM 410-WRAPUP.
           STOP RUN.
       100-FILE-INITIALIZATION SECTION.
       110-OPEN.
           OPEN OUTPUT OT-FIL.
           CALL IOPEN USING IFC,DFC,REC-AREA,REC-SZ,KEYOFF,KEY-SZ.

    **    NOTICE THAT THE RECORD DELIVERY AREA (REC-AREA) IS AN 01 LEVEL
    **    ITEM IN THE WORKING-STORAGE SECTION.  IT COULD ALSO BE IN THE
    **    FILE SECTION.

       200-RETRIEVAL SECTION.
       210-RETRIEVE.
           MOVE SPACES TO INVENTORY.
         CALL GETSEQ USING DFC,ERR.
           IF EOF MOVE 1 TO EOF-FLAG.
       300-WRITE SECTION.
       310-WRITE.
           PERFORM 350-MOVE.
           WRITE INVENTORY AFTER 2.
           PERFORM 210-RETRIEVE.
       350-MOVE.
           MOVE PART-NUMBER TO PART-NUMBER-OUT.
           MOVE PART-NAME TO PART-NAME-OUT.
           MOVE PRICE TO PRICE-OUT.
           MOVE QTY-ON-HAND TO QTY-ON-HAND-OUT.
           MOVE BIN-LOC TO BIN-LOC-OUT.
       400-WRAPUP SECTION.
       410-WRAPUP.
           DISPLAY "RETRIEVAL COMPLETE".
           CLOSE OT-FIL.
           CALL ICLOSE USING DFC.
$      EXECUTE
$      PRMFL      D1,R/W,R,DATA FILE CAT/FILE STRING
$      PRMFL      I1,R/W,R,INDEX FILE CAT/FILE STRING
$      SYSOUT     OT
$      ENDJOB
***EOF
```

EXHIBIT ISP030-1 (Cont)

IOPEN SUBROUTINE

Format 1

CALL IOPEN USING record-area, data-file-code-1, desc-file-code [, error-code].

Format 2

CALL IOPEN USING index file-code, data-file-code-1, record-area, record-size, key-offset, key-size [, pct-fill [, data-page-size [, coll-comm]]].

EXHIBIT ISP030-2

GETSEQ SUBROUTINE

CALL GETSEQ USING data-file-code-1, error-code [, alt-record-area [, alt-record-size]].

EXHIBIT ISP030-3

DATA BASE SEQUENTIAL RETRIEVAL PROBLEM

Using the data base built in lesson ISP020 (described in exhibit ISP020-7), write a program, including the appropriate ISP subroutines, to sequentially retrieve all the data records from the data base and print them in a readable fashion.

Use the following JCL for your job:

```
$      IDENT    . . .
$      USERID   . . .
$      COBOL    CBL68
         .
         .  Your ISP sequential retrieval program
         .
$      EXECUTE
$      PRMFL    D1,R,R,DATA-FILE CAT/FILE STRING
$      PRMFL    I1,R,R,INDEX-FILE CAT/FILE STRING
$      SYSOUT   OT
$      ENDJOB
***EOF
```

NOTE: Be sure to check and report any errors which are encountered during the data base retrieval.

EXHIBIT ISP030-4

```
$      IDENT
$      USERID
$      COBOL   CBL68
       IDENTIFICATION DIVISION.
       PROGRAM-ID.  EXBT32.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER.  6000-EIS.
       OBJECT-COMPUTER.  6000-EIS.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
          SELECT OT-FIL ASSIGN TO OT-LISTING.
          SELECT INFO-FILE ASSIGN TO IN.
       I-O-CONTROL.
          APPLY STANDARD ON OT-FIL,INFO-FILE.
       DATA DIVISION.
       FILE SECTION.
       FD OT-FIL   LABEL RECORDS ARE STANDARD.
       01 INVENTORY
          03      PART-NUMBER-OUT           PIC   X(5).
          03      FILLER                    PIC   X(2).
          03      PART-NAME-OUT             PIC   X(20).
          03      FILLER                    PIC   X(2).
          03      PRICE-OUT                 PIC   X(5)9.99.
          03      FILLER                    PIC   X(2).
          03      QTY-ON-HAND-OUT           PIC   Z(5).
          03      FILLER                    PIC   X(2).
          03      BIN-LOC-OUT               PIC   X(8).
       FD INFO-FILE     LABEL RECORDS ARE STANDARD.
       01 IN-PART-NUM-FIL.
          03      PART-NUM-IN               PIC   X(5).
          03      FILLER                    PIC   X(75).
       WORKING-STORAGE SECTION.
       77 IFC                               PIC   X(2)          VALUE "I1".
       77 DFC                               PIC   X(2)          VALUE "D1".
       77 REC-SZ                            PIC   9(6) COMP-1   VALUE 14.
       77 KEY-OFF                           PIC   9(6) COMP-1   VALUE 1.
       77 KEY-SZ                            PIC   9(6) COMP-1   VALUE 9.
       77 ERR                               PIC   X(3)
          88      NO-REC                                        VALUE "X02".
       77 EOF-FLAG                          PIC   9(1).

       01 REC-AREA.
          03     PART-NUMBER                PIC   X(5).
          03     PART-NAME                  PIC   X(20).
          03     PRICE                      PIC   9(5)V99.
          03     QTY-ON-HAND                PIC   9(5).
          03     BIN-LOC                    PIC   X(8).
```

EXHIBIT ISP030-5

```
       PROCEDURE DIVISION.
       000-DRIVER SECTION.
       010-DRIVER.
           PERFORM 110-OPEN.
           PERFORM 210-READ.
           MOVE 0 TO EOF-FLAG.
           PERFORM 310-RETRIEVE UNTIL EOF-FLAG = 1.
           PERFORM 410-WRAPUP.
           STOP RUN.
100-FILE-INIT SECTION.
110-OPEN.
           OPEN INPUT INFO-FILE OUTPUT OT-FIL.
           CALL IOPEN USING IFC,DFC,REC-AREA,REC-SZ,KEY-OFF,KEY-SZ.
       200-READ SECTION.
       210-READ.
           READ INFO-FILE AT END MOVE 1 TO EOF-FLAG.
       300-INFORMATION SECTION.
       310-RETRIEVE.
           MOVE SPACES TO INVENTORY.
           MOVE PART-NUM-IN TO PART-NUMBER.
           CALL GETRAN USING DFC,ERR.
           IF NO-REC
                DISPLAY PART-NUMBER "RECORD NOT ON FILE"
                PERFORM 210-READ
           ELSE
               PERFORM 350-MOVE-WRITE
               PERFORM 210-READ
       350-MOVE-WRITE.
           MOVE PART-NUMBER TO PART-NUMBER-OUT
           MOVE PART-NAME TO PART-NAME-OUT
           MOVE PRICE TO PRICE-OUT
           MOVE QTY-ON-HAND TO QTY-ON-HAND-OUT
           MOVE BIN-LOC TO BIN-LOC-OUT
           WRITE INVENTORY AFTER 2.
       400-WRAPUP SECTION.
       410-WRAPUP.
           DISPLAY "RETRIEVAL COMPLETE".
           CLOSE OT-FIL, INFO-FILE.
           CALL ICLOSE USING DFC.
$      EXECUTE
$      PRMFL        IN,R,S,CDTSCDTS/ISP/DATA030
$      PRMFL        D1,R/W,R,DATA FILE CAT/FILE STRING
$      PRMFL        I1,R/W,R,INDEX FILE CAT/FILE STRING
$      SYSOUT       OT
$      ENDJOB
***EOF
```

EXHIBIT ISP030-5 (Cont)

GETRAN SUBROUTINE

```
CALL GETRAN USING data-file-code-1,        error-code
       [, alt-record-area [, alt-record-size]] .
```

EXHIBIT ISP030-6

DATA BASE RANDOM RETRIEVAL PROBLEM

Write a program to randomly access the ISP data base created in lesson ISP020 (described in Exhibit ISP020-7). Read the social security numbers from the file located at catalog/file string:

CDTSCDTS/ISP/DATA030

For each number you read, access your ISP data base to retrieve the record which contains the SSAN which matches that number and print out the contents of the record.

The social security numbers will appear in the following format on the input file:

FIELD	STARTING COLUMN	ENDING COLUMN
SSAN	1	9
FILLER	10	80

Be advised that some of the social security numbers on the input file may not be on your ISP data base.

Using the following JCL for your job:

```
$     IDENT   . . .
$     USERID  . . .
$     COBOL       CBL68
        .
        .  Your ISP random retrieval program
        .
$     EXECUTE
$     PRMFL   IN,R,S,CDTSCDTS/ISP/DATA030
$     PRMFL   D1,R,R,DATA-FILE CAT/FILE STRING
$     PRMFL   I1,R,R,INDEX-FILE CAT/FILE STRING
$     SYSOUT  OT
$     ENDJOB
***EOF
```

NOTE: Be sure to check and report any errors which are encountered during the data base retrieval.

EXHIBIT ISP030-7

```
$      IDENT
$      USERID
$      COBOL   CBL68
       IDENTIFICATION DIVISION.
       PROGRAM-ID.  XHBT40.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER.  6000-EIS.
       OBJECT-COMPUTER.  6000-EIS.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
          SELECT IN-FIL ASSIGN TO IN.
       I-O-CONTROL.
          APPLY STANDARD ON IN-FIL.
       DATA DIVISION.
       FILE SECTION.
       FD IN-FIL LABEL RECORDS ARE STANDARD.
       01 INVENTORY.
          03    PART-NUMBER       PIC  X(5).
          03    PART-NAME         PIC  X(20).
          03    PRICE             PIC  9(5)V99.
          03    QTY-ON-HAND       PIC  9(5).
          03    BIN-LOC           PIC  X(8).
          03    TRANS-CODE        PIC  X(1).
                88  ADRC                                VALUE "A".
                88  DLRC                                VALUE "D".
                88  CHGRC                               VALUE "C".
       WORKING-STORAGE SECTION.
       77 IFC                     PIC  X(2)             VALUE "I1".
       77 DFC                     PIC  X(2)             VALUE "D1".
       77 REC-SZ                  PIC  9(6)   COMP-1    VALUE 8.
       77 KEY-OFF                 PIC  9(6)   COMP-1    VALUE 0.
       77 KEY-SZ                  PIC  9(6)   COMP-1    VALUE 5.
       77 ALT-REC-SZ              PIC  9(6)   COMP-1.
       77 ERR                     PIC  X(3)   SYNC RIGHT.
          88    OKAY                                    VALUE "000".
          88    NO-REC                                  VALUE "X02".
          88    DUP-REC                                 VALUE "X07".
       77 SEQ-CK                  PIC  X(12).
       77 EOF-FLAG                PIC  9(1).
       01 ALT-REC-AREA
          03    ALT-PART-NUMBER   PIC  X(5).
          03    ALT-PART-NAME     PIC  X(20).
          03    ALR-PRICE         PIC  9(5)V99.
          03    ALT-QTY-ON-HAND   PIC  9(5).
          03    ALT-BIN-LOC       PIC  X(8).
          03    ALT-TRANS-CODE    PIC  X(1).
       PROCEDURE DIVISION.
       000-DRIVER SECTION.
       010-DRIVER.
            PERFORM 110-OPEN.
            PERFORM 210-READ.
            MOVE 0 TO EOF-FLAG.
```

EXHIBIT ISP040-1

```
          PERFORM 310-CHANGE UNTIL EOF-FLAG = 1.
          PERFORM 410-WRAPUP.
          STOP RUN.
    100-FILE-INIT SECTION.
    110-OPEN.
          OPEN INPUT IN-FIL.
          CALL IOPEN USING IFC,DFC,INVENTORY,REC-SZ,KEY-OFF,KEY-SZ.
    200-READ SECTION.
    210-READ.
          READ IN-FIL AT END MOVE 1 TO EOF-FLAG.
    300-CHANGE SECTION.
    310-CHANGE.
          IF ADRC PERFORM 340-ADD-RCD.
          IF DLRC PERFORM 360-DEL-RCD.
          IF CHGRC PERFORM 380-CHG-RCD.
          PERFORM 210-READ.
    340-ADD-RCD.
   ** NOTICE THAT THE RECORD DELIVERY AREA (INVENTORY) WAS PRIMED
   ** WITH THE RECORD TO BE ADDED BY THE COBOL READ VERB, PRIOR
   ** TO CALLING ADDREC.
          CALL ADDREC USING DFC,ERR.
          IF DUP-REC DISPLAY PART-NUMBER, "DUP RECORD".
    370-DEL-RCD.

   ** NOTICE THAT THE RECORD DELIVERY AREA (INVENTORY) WAS PRIMED WITH
   ** THE RIGHT KEY VALUE BY THE COBOL READ VERB BEFORE CALLING
   ** DEL REC.
          CALL DELREC USING DFC,ERR.
          IF NO-REC DISPLAY PART-NUMBER, "NOT IN FILE".
    380-CHG-RCD.
          MOVE PART-NUMBER TO ALT-PART-NUMBER.
   ** FOR CHGREC, THE ALTERNATE RECORD DELIVERY AREA (ALT-REC-AREA)
   ** IS PRIMED BY A CALL TO GETRAN AND BY ANY MOVES TO FIELDS WHICH
   ** ARE TO CHANGE.
          CALL GETRAN USING DFC,ERR,ALT-REC-AREA,ALT-REC-SZ.
          IF NO-REC DISPLAY PART-NUMBER, "NOT IN FILE".
          IF PART-NUMBER EQUAL SPACES MOVE ALT-PART-NUMBER TO PART-NUMBER.
          IF PRICE EQUAL SPACES MOVE ALT-PRICE TO PRICE.
          IF QTY-ON-HAND EQUAL SPACE
            MOVE ALT-QTY-ON-HAND TO QTY-ON-HAND.
          IF BIN-LOC EQUAL SPACES MOVE ALT-BIN-LOC TO BIN-LOC.
          CALL CHGREC USING DFC,ERR.
          IF DUP-REC DISPLAY PART-NUMBER, "DUP RECORD".
    400-WRAPUP SECTION.
    410-WRAPUP.
          DISPLAY "UPDATE COMPLETE".
          CLOSE IN-FILE.
          CALL ICLOSE USING DFC.
$   EXECUTE
$   PRMFL   D1,R/W,R,DATA FILE CAT/FILE STRING
$   PRMFL   I1,R/W,R,INDEX FILE CAT/FILE STRING
$   PRMFL   IN,R,S,CDTSCDTS/ISP/DATA040
$   ENDJOB
***EOF
```

EXHIBIT ISP040-1 (Cont)

ADDREC SUBROUTINE

CALL ADDREC using data-file-code-1, error-code [, alt-record-size].

EXHIBIT ISP040-2

DELREC SUBROUTINE

CALL DELREC USING data-file-code-1, error-code.

EXHIBIT ISP040-3

CHGREC SUBROUTINE

CALL CHGREC USING data-file-code-1, error-code.

EXHIBIT ISP040-4

DATA BASE UPDATE PROBLEM

Write a program which will update the ISP data base created in Lesson ISP020 (described in Exhibit ISP020-7). Include routines to add, delete, and change records on your ISP data base. Read the update transactions from the file located at cat/file string:

CDTSCDTS/ISP/DATA040

These records are identical in format to the original input data (see Exhibit ISP020-7), except that column 1 contains a character code for the type of transaction:

A - Addition
D - Deletion
C - Change

For additions, all fields will contain data. For the deletions, only the transaction type (column 1) and the SSAN field (columns 2-10) will contain data (the key field). For the changes, only the transaction type (column 1), SSAN field, and the field(s) to be changed will contain data. So fields on the transaction records without data should not change on the data base.

When the program, which posts all the transaction to your data base, has been run, rerun the program from Lesson ISP030 (described in Exhibit ISP030-4) to obtain a sequential listing of the updated data base.

Use the following JCL for your job:

```
$     IDENT  . . .
$     USERID . . .
$     COBOL       CBL68
       .
       .   Your ISP update program
       .
$     EXECUTE
$     PRMFL   IN,R,S,CDTSCDTS/ISP/DATA040
$     PRMFL   D1,R/W,R,DATA FILE CAT/FILE STRING
$     PRMFL   I1,R/W,R,INDEX FILE CAT/FILE STRING
$     SYSOUT  OT
$     ENDJOB
***EOF
```

NOTE: Be sure to check and report any errors which are encountered during the data base update.

XREWND SUBROUTINE

CALL XREWND USING data-file-code-1

[, error-code, function-code].

EXHIBIT ISP050-1

XSTAT SUBROUTINE

CALL SXTAT USING data-file-code-1, stat-area.

EXHIBIT ISP050-2

XFLUSH SUBROUTINE

CALL XFLUSH (USING function-code).

EXHIBIT ISP050-3

XUTIL PROGRAM

```
1      8       16
$      PROGRAM XUTIL
$      LIMITS  limit values
$      file    .X,sequential,descriptor cards
$      file    OT,sequential,unload file
$      file    fc,random,ISP index
$      file    fc,random,ISP primary data file
$      file    fc,random,ISP additional data file
       .
       .
       .
```

UNLOAD FUNCTION

```
1      8       16
$      PROGRAM XUTIL
$      LIMITS  limit values
$      file    .X,sequential,descriptor cards
$      file    IN,sequential,load/reload file
$      file    fc,random,ISP index
$      file    fc,random,ISP primary data file
$      file    fc,random,ISP additional data file
       .
       .
       .
```

LOAD/RELOAD FUNCTION

EXHIBIT ISP050-4

XUTIL PROGRAM

```
1      8       16
$      IDENT   ...
$      PROGRAM XUTIL
$      LIMITS  ...
$      DATA    .X
ISP    DATA    FC=D1
ISP    INDEX   FC-I1
ISP    RECORD  RECSZ=8,KEYSZ=5
$      FILE    OT,X1S,5L
$      PRMFL   D1,R/W,R,ISP/DATA
$      PRMFL   I1,R/W,R,ISP/INDEX
$      PROGRAM XUTIL
$      LIMITS  ...
$      DATA    .X
ISP    DATA    FC=D1
ISP    INDEX   FC=I1
ISP    RECORD  RECSZ=8,KEYSZ=5
$      FILE    OT,X1R,5L
$      PRMFL   D1,R/W,R,ISP/DATA
$      PRMFL   I1,R/W,R,ISP/INDEX
$      ENDJOB
```

EXHIBIT ISP050-4 (Cont)

ISP DATA DESCRIPTOR CARD

```
1      8       16
ISP    DATA    FC=data-file-code-n n=1,2,...,8

              [,PAGESZ=data-page-size]

              [,PAGEFIL=pct-fill]

              [,COLCOM]

              [,BLKSZ=block-size]
```

EXHIBIT ISP050-5

ISP INDEX DESCRIPTOR CARD

```
1      8      16
ISP    INDEX  FC=index-file-code

             [,PAGESZ=index-page-size]
```

EXHIBIT ISP050-6

ISP RECORD DESCRIPTOR CARD

```
1      8      16
ISP    RECORD RECSZ=record-size

              ,KEYSZ=key-size

              [,KEYOFF=key-offset]
```

EXHIBIT ISP050-7

```
$       IDENT
$       USERID
$       COBOL   CBL68
        IDENTIFICATION DIVISION.
        PROGRAM-ID.  XHBT50.
      ** THIS IS THE SAME PROGRAM AS IN ISP020-1 EXCEPT HERE WE USE
      ** DESCRIPTOR CARDS.
        ENVIRONMENT DIVISION.
        CONFIGURATION SECTI8ON.
        SOURCE-COMPUTER. 6000-EIS.
        OBJECT-COMPUTER. 6000-EIS.
        INPUT-OUTPUT SECTION.
        FILE-CONTROL.
                SELECT IN-FIL ASSIGN TO IN.
        I-O-CONTROL.
                APPLY STANDARD ON IN-FIL.
        DATA DIVISION.
        FILE SECTION.
        FD      IN-FIL LABEL RECORDS ARE STANDARD.
        01      INVENTORY.
                03    PART-NUMBER              PIC  X(5).
                03    PART-NAME                PIC  X(20).
                03    PRICE                    PIC  9(5)V99.
                03    QTY-ON-HAND              PIC  9(5).
                03    BIN-LOC                  PIC  X(8).
        WORKING-STORAGE SECTION.
        77      ESC-FC                         PIC  X(2)          VALUE "DD".
        77      DFC                            PIC  X(2)          VALUE "D1".
        77      ALT-REC-SZ                     PIC  9(6)  COMP-1.
        77      ERR                            PIC  X(3)  SYNC-RIGHT.
                88      OKAY                                      VALUE "000".
        77      SEQ-CK                         PIC  X(12).
        77      EOF-FLAG                       PIC  9(1).
        PROCEDURE DIVISION.
        000-DRIVER SECTION.,
        010-DRIVER.
                PERFORM 110-OPEN.
                PERFORM 210-READ.
                MOVE 0 TO EOF-FLAG.
                PERFORM 310-LOAD UNTIL EOF-FLAG = 1.
                PERFORM 410-WRAPUP.
                STOP RUN.
        100-OPEN SECTION.
        110-OPEN.
                OPEN INPUT IN-FIL.
                CALL NOPEN USING INVENTORY,DFC,DESC-FAC,ERR.
        200-READ SECTION.
        210-READ.
                READ IN-FIL AT END MOVE 1 TO EOF-FLAG.
        300-LOAD SECTION.
        310-LOAD.
                CALL FILINT USING DFC,ALT-REC-SZ,ERR,SEQ-CK.
                IF NOT OKAY DISPLAY "LOAD ERRORS".
                PERFORM 210-READ.
```

EXHIBIT ISP050-8

```
        400-WRAPUP SECTION.
        410-WRAPUP.
                CLOSE IN-FIL.
                CALL ICLOSE USING DFC.
$       EXECUTE
$       PRMFL        D1,R/W,R,DATA FILE CAT/FILE STRING
$       PRMFL        I1,R/W/R,INDEX FILE CAT/FILE STRING
$       PRMFL        IN,R,S,CDTSCDTS/ISP/DATA020
$       DATA         DD
ISP     DATA         FC=D1,PAGESZ-384,PAGEFIL=75,COLCOM,BLKS-384
ISP     INDEX        FC=I1,PAGESZ=384
ISP     RECORD       REC,SZ=8,KEYSZ=5,KEYOFF=0
$       ENDJOB
***EOF
```

EXHIBIT ISP050-8 (Cont)

DATA BASE REORGANIZATION PROBLEM

Use the facility of the "XUTIL" utility program to reorganize your ISP data base. This will physically move the records from the overflow areas to their proper places within your data base and also physically delete and recapture the space occupied by logically deleted records. Use the same specifications as described in Exhibit ISP020-7, the load problem.

These specifications are:

page size - 320 words

percent fill - 100%

record size - 14 words.

key size - 9 characters.

key offset - 1 character.

For the block size, use the default.

EXHIBIT ISP050-9

ISP JOURNAL DESCRIPTOR CARD

```
1      8       16
ISP    JOURNAL FC=journ-file-code
```

EXHIBIT ISP050-10

XJRNAL PROGRAM

```
1      8       16
$      PROGRAM XJRNAL

$      LIMITS  limit values

$      file    .X,sequential,descriptor cards

$      file    fc,random,ISP primary data file

[$     file    fc,random,ISP additional data file]

       .
       .
       .

       TAPE    T1,...
```

EXHIBIT ISP050-11

ISP BEFORE/AFTER DESCRIPTOR CARDS

```
1      8       16
ISP    BEFORE  FC=data-file-code-n

ISP    AFTER   FC-data-file-code-n
```

EXHIBIT ISP050-12

INTRODUCTION TO DATA PROCESSING (H6000-CDT) COURSE EXHIBITS

GLOSSARY OF DATA PROCESSING TERMS

ADDRESS - An identification, as represented by a name, label, or number, for a register, location in storage or any other data.

BINARY - A number system which uses only the numbers 0 and 1 to represent all numbers.

BIT - A binary digit which has either a 0 or 1 value.

BYTE - A sequence of adjacent binary digits (bits) operated upon as a unit and usually shorter than a computer word.

CHARACTER - A letter, digit, or other symbol that is used as part of the organization, control, or representation of data.

FIELD - A record in an assigned area to be marked with information.

FILE - An organized collection of logical records.

LOGICAL RECORD - A collection of items independent of their physical environment. Portions of the same logical record may be located in different physical records.

MACHINE LANGUAGE - A language that is used directly (without translation) by the machine, usually in binary code, i.e., numeric ones and zeroes used to signify an "ON" or "OFF" condition such as the presence or absence of electrical or magnetic charge.

MAGNETIC CORE STORAGE - A storage device in which binary data is represented by the direction of magnetization in each unit of an array of magnetic material. It may be made of such materials as iron, iron oxide, or ferrite and in such shapes as wires, tapes, toroids, rods, or thin film.

MAGNETIC DISK - A flat circular plate with a magnetic surface on which data can be stored by selective magnetization of portions of the surface.

MAGNETIC INK - An ink that contains particles of a magnetic substance whose presence can be detected by magnetic sensors.

MAGNETIC TAPE - A tape with a magnetic surface on which data can be stored by selective magnetization of portions of the surface.

MULTIPROCESSING SYSTEM - A computer system using two or more interconnecting processing units to execute programs simultaneously.

MULTIPROGRAMMING - A technique for handling numerous routines or programs seemingly simultaneously by overlapping or interweaving their execution, that is, by permitting more than one program to time-share machine components.

OCTAL - A numbering system which uses the numbers 0 through 7 to compose all numbers.

PAPER TAPE - Also called punched tape; a tape on which a pattern of holes or cuts is used to represent data.

PHYSICAL RECORD - A number of logical records that are read at one memory access of a computer to make input/output mode efficient.

PROGRAM - A series of actions proposed in order to achieve a certain result.

PUNCHED CARDS - A card on which can be punched a pattern of holes to represent data.

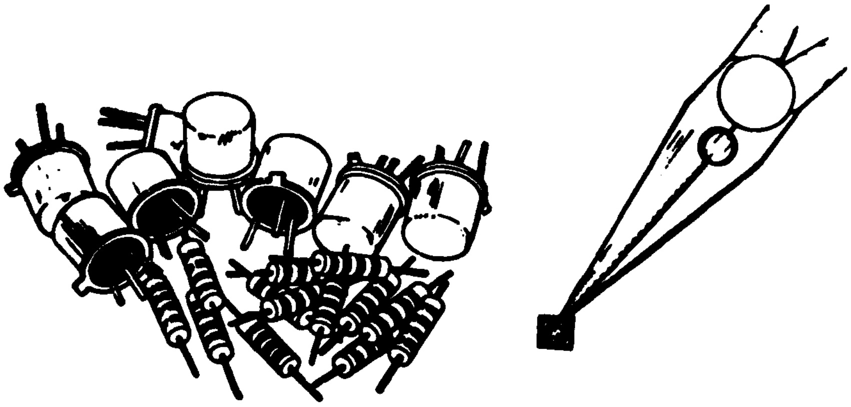
REAL-TIME PROCESSING - Pertaining to the performance of a computation during the actual time that the related physical process transpires in order that results of the computation can be used in guiding the physical process.

SORT - To arrange items into groups according to definite rules.

TIME-SHARING - The use of a device for two or more purposes during the same overall time interval, accomplished by interspersing component actions in time.

WORD - A sequence of bits or characters treated as a unit and capable of being stored in one computer location.

EXHIBIT IDP010-1 (Cont)



The transistors and resistors pictured on the left were replaced by the third generation integrated circuit (pictured on the right).

EXHIBIT IDP020-1

INPUT ELEMENT OF A DIGITAL COMPUTER

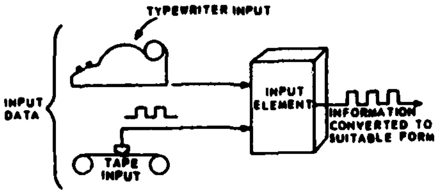


EXHIBIT IDP020-2

OUTPUT ELEMENT OF A DIGITAL COMPUTER

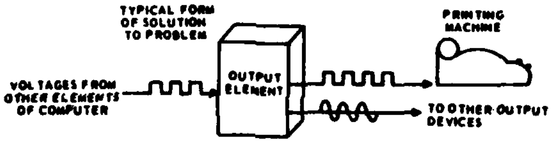

TYPICAL FORM OF SOLUTION TO PROBLEM
PRINTING MACHINE
VOLTAGES FROM OTHER ELEMENTS OF COMPUTER
OUTPUT ELEMENT
TO OTHER OUTPUT DEVICES

EXHIBIT IDP020-3

ARITHMETIC ELEMENT OF A DIGITAL COMPUTER

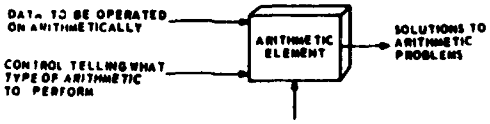
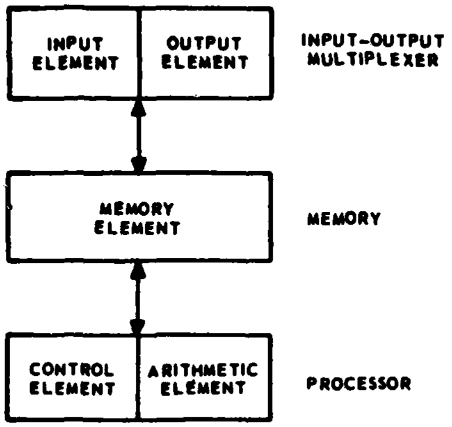

DATA TO BE OPERATED ON ARITHMETICALLY
CONTROL TELLING WHAT TYPE OF ARITHMETIC TO PERFORM
ARITHMETIC ELEMENT
SOLUTIONS TO ARITHMETIC PROBLEMS

EXHIBIT IDP020-4



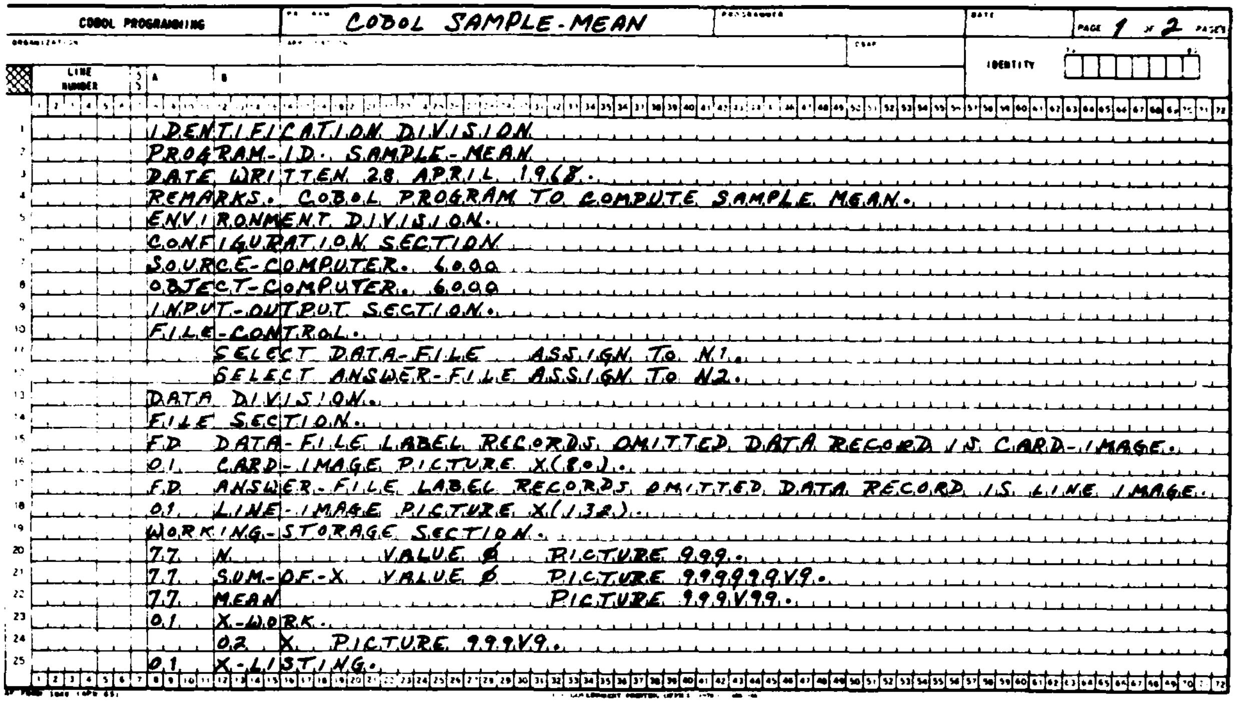
H6000 SERIES CONFIGURATION OF DIGITAL COMPUTER ELEMENTS

EXHIBIT IDP020-5

```
1      8       16
       LBL     .GBNRY
       TTL     BCD TO BINARY - ".GBNRY" & "BCDBNY"
       SYMDEF .GBNRY.BCDBNY
*
.GBNR YSTX1    .E.L..
       STI     .E.L..
       TRA     STMAIN
*
BCDNY STXQ     .E.L..
       STI     .E.L..
       ANAQ    =12H
*
STMAIN         LDAQ=12H000123456789
       STAQ    BNY803
       LDQ     0.DL
       LDA     BNY802
       STA     BNY801
*
BNY010         MPY10.DL
       ADQ     BNY801.SC
       TTF     BNY010
*
       LLS     36
       LDQ     =3H OK,DL
       MME     GEBORT
*
*****      WORKING STORAGE      *****
*
BNY801         BSS1
BNY802         TALLYBNY803,10,2

       EVEN
BNY803         BSS2
```

ASSEMBLER LANGUAGE CODE

EXHIBIT IDP030-1



COBOL PROGRAMMING

PROGRAM COBOL SAMPLE-MEAN

PROGRAMMER

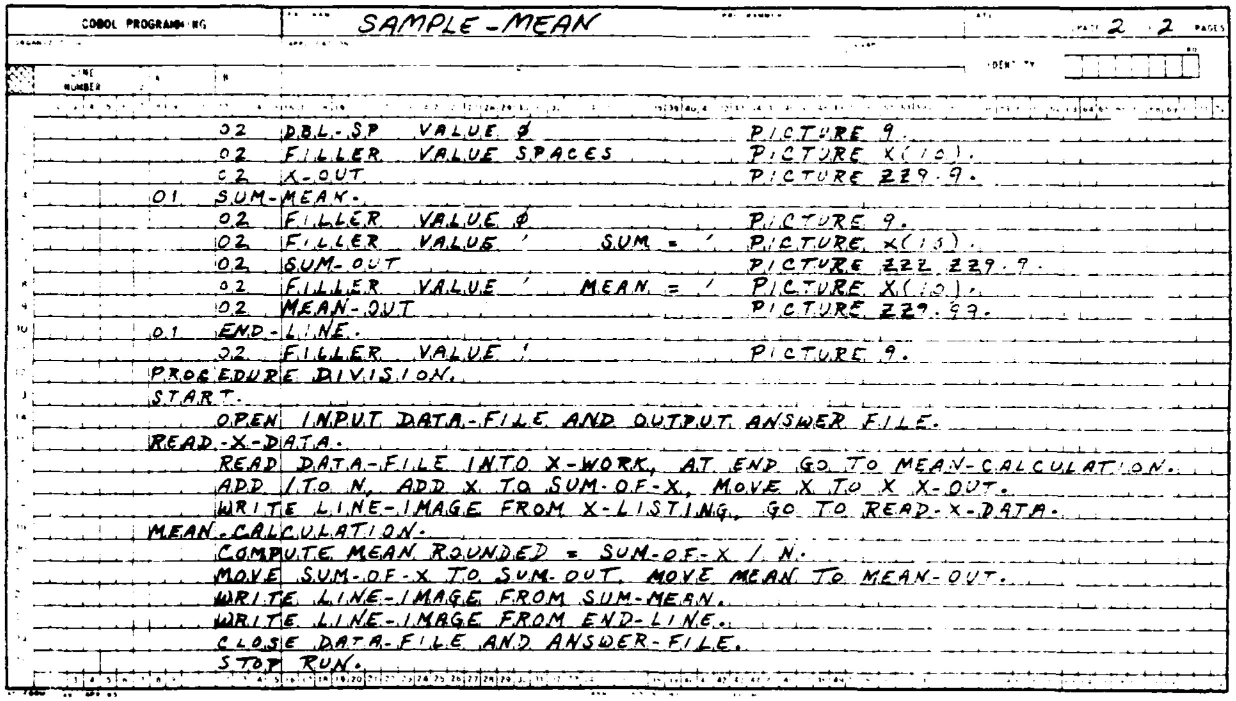
DATE

PAGE 1 of 2 PAGES

IDENTITY

```
       IDENTIFICATION DIVISION
       PROGRAM-ID. SAMPLE-MEAN
       DATE WRITTEN 28 APRIL 1968.
       REMARKS. COBOL PROGRAM TO COMPUTE SAMPLE MEAN.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION
       SOURCE-COMPUTER. 6000
       OBJECT-COMPUTER. 6000
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT DATA-FILE   ASSIGN TO N1.
           SELECT ANSWER-FILE ASSIGN TO N2.
       DATA DIVISION.
       FILE SECTION.
       FD  DATA-FILE LABEL RECORDS OMITTED DATA RECORD IS CARD-IMAGE.
       01  CARD-IMAGE PICTURE X(80).
       FD  ANSWER-FILE LABEL RECORDS OMITTED DATA RECORD IS LINE IMAGE.
       01  LINE-IMAGE PICTURE X(132).
       WORKING-STORAGE SECTION.
       77  N            VALUE Ø   PICTURE 999.
       77  SUM-OF-X     VALUE Ø   PICTURE 99999V9.
       77  MEAN                   PICTURE 999V99.
       01  X-WORK.
           02  X  PICTURE 999V9.
       01  X-LISTING.
```

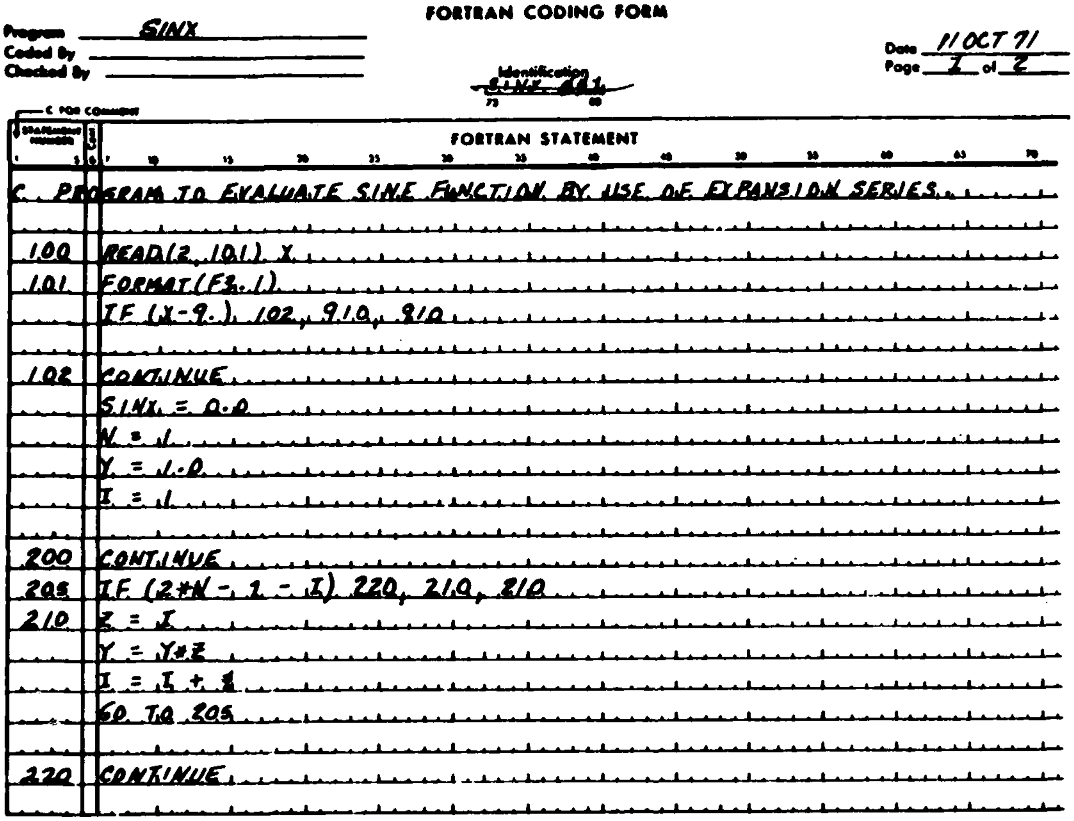
EXHIBIT IDP030-2



COBOL PROGRAMMING SAMPLE-MEAN PAGE 2 OF 2 PAGES

```
       02  DBL-SP   VALUE Ø                  PICTURE 9.
       02  FILLER   VALUE SPACES             PICTURE X(10).
       02  X-OUT                             PICTURE ZZ9.9.
    01 SUM-MEAN.
       02  FILLER   VALUE Ø                  PICTURE 9.
       02  FILLER   VALUE '       SUM = '    PICTURE X(13).
       02  SUM-OUT                           PICTURE ZZZ ZZ9.9.
       02  FILLER   VALUE '    MEAN = '      PICTURE X(10).
       02  MEAN-OUT                          PICTURE ZZ9.99.
    01 END-LINE.
       02  FILLER   VALUE !                  PICTURE 9.
    PROCEDURE DIVISION.
    START.
       OPEN INPUT DATA-FILE AND OUTPUT ANSWER FILE.
    READ-X-DATA.
       READ DATA-FILE INTO X-WORK, AT END GO TO MEAN-CALCULATION.
       ADD 1 TO N, ADD X TO SUM-OF-X, MOVE X TO X X-OUT.
       WRITE LINE-IMAGE FROM X-LISTING, GO TO READ-X-DATA.
    MEAN-CALCULATION.
       COMPUTE MEAN ROUNDED = SUM-OF-X / N.
       MOVE SUM-OF-X TO SUM-OUT. MOVE MEAN TO MEAN-OUT.
       WRITE LINE-IMAGE FROM SUM-MEAN.
       WRITE LINE-IMAGE FROM END-LINE.
       CLOSE DATA-FILE AND ANSWER-FILE.
       STOP RUN.
```

EXHIBIT IDP020-2 (Cont)



FORTRAN CODING FORM

Program SINX

Coded By

Checked By

Date 11 OCT 71

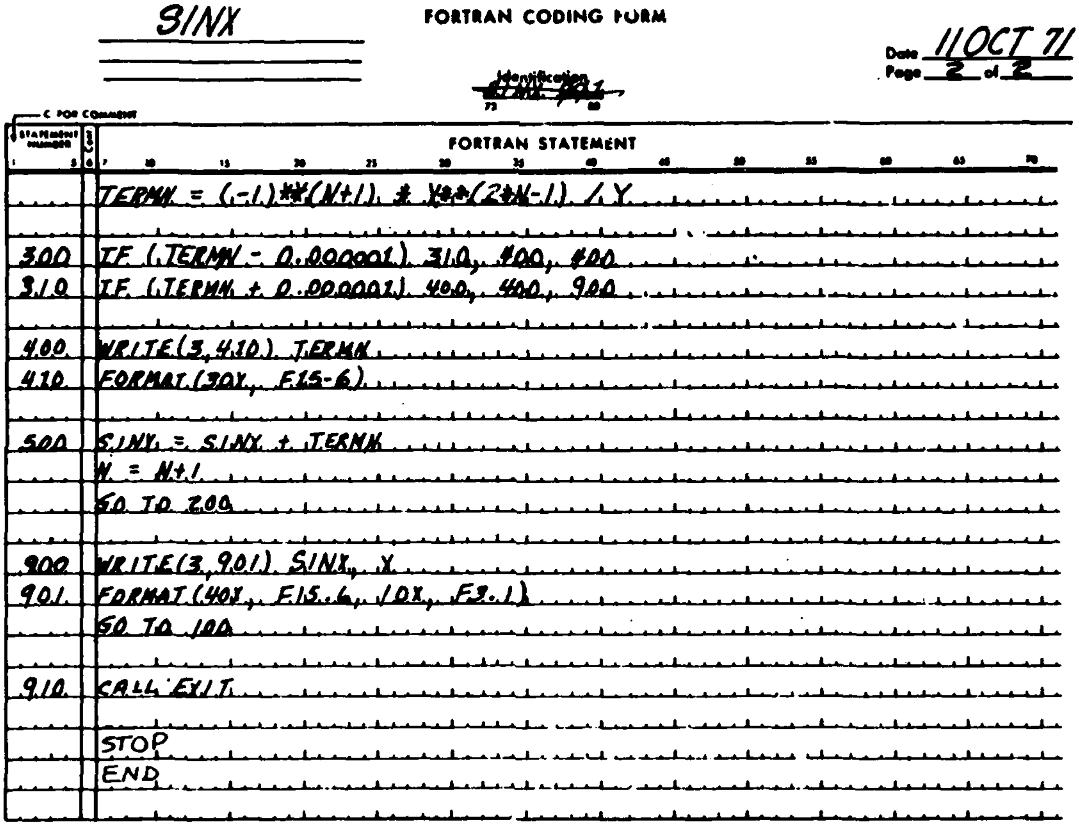
Page 1 of 2

Identification ~~SINX 001~~

73 80

```
C PROGRAM TO EVALUATE SINE FUNCTION BY USE OF EXPANSION SERIES.

  100 READ(2, 101) X
  101 FORMAT(F3.1)
      IF (X-9.) 102, 910, 910

  102 CONTINUE
      SINX = 0.0
      N = 1
      Y = 1.0
      I = 1

  200 CONTINUE
  205 IF (2*N - 1 - I) 220, 210, 210
  210 Z = I
      Y = Y*Z
      I = I + 1
      GO TO 205

  220 CONTINUE
```

EXHIBIT IDP030-3



SINX

FORTRAN CODING FORM

Date 11OCT 71
Page 2 of 2

Identification ~~[illegible]~~

C FOR COMMENT

STATEMENT NUMBER | FORTRAN STATEMENT

```
      TERMN = (-1)**(N+1) * Y**(2*N-1) / Y

300   IF (TERMN - 0.000001) 310, 400, 400
310   IF (TERMN + 0.000001) 400, 400, 900

400   WRITE(3,410) TERMN
410   FORMAT(30X, F15-6)

500   SINX = SINX + TERMN
      N = N+1
      GO TO 200

900   WRITE(3,901) SINX, X
901   FORMAT(40X, F15.6, 10X, F3.1)
      GO TO 100

910   CALL EXIT

      STOP
      END
```

EXHIBIT IDP030-3 (Cont)

HARDWARE SOFTWARE COST TRENDS

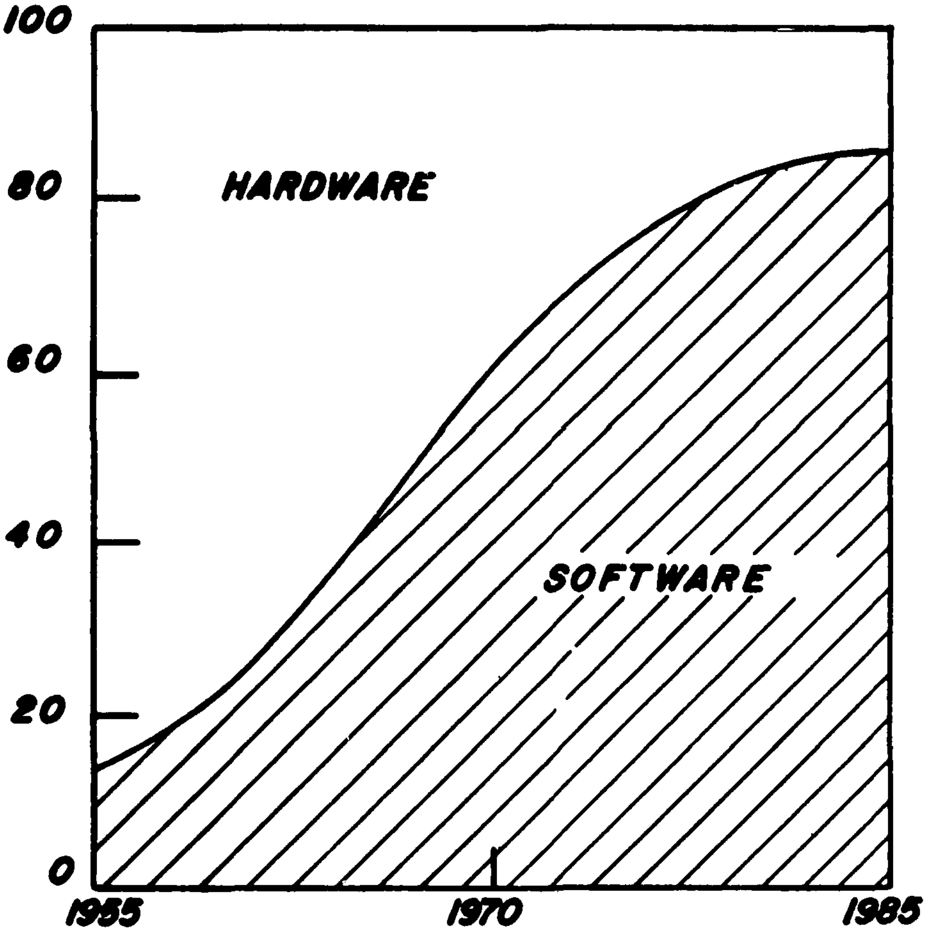


EXHIBIT IDP040-1

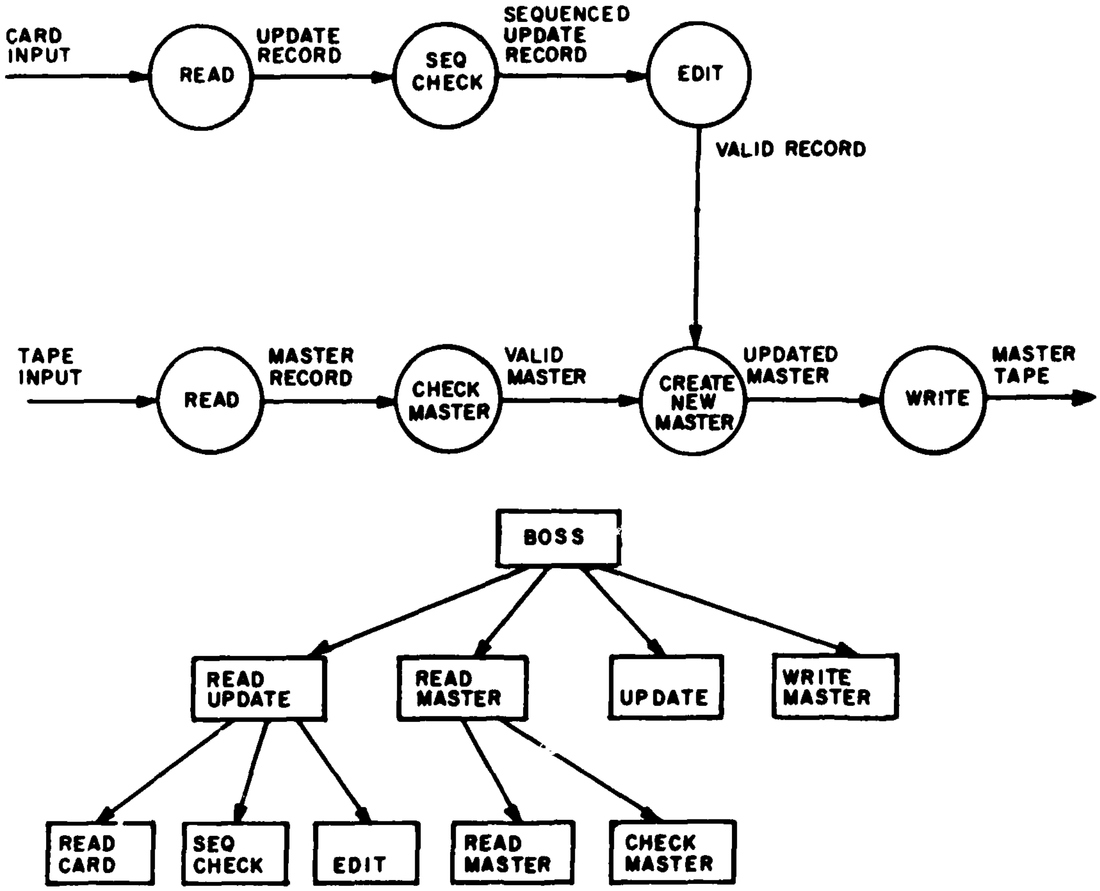


EXHIBIT IDP040-2

HIPO: DESIGN AID AND DOCUMENTATION TOOL

DRAWING HIPO DIAGRAMS

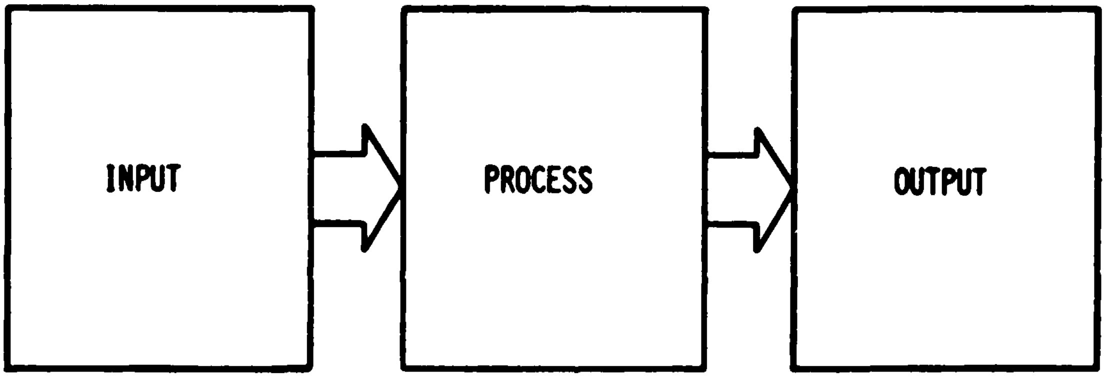
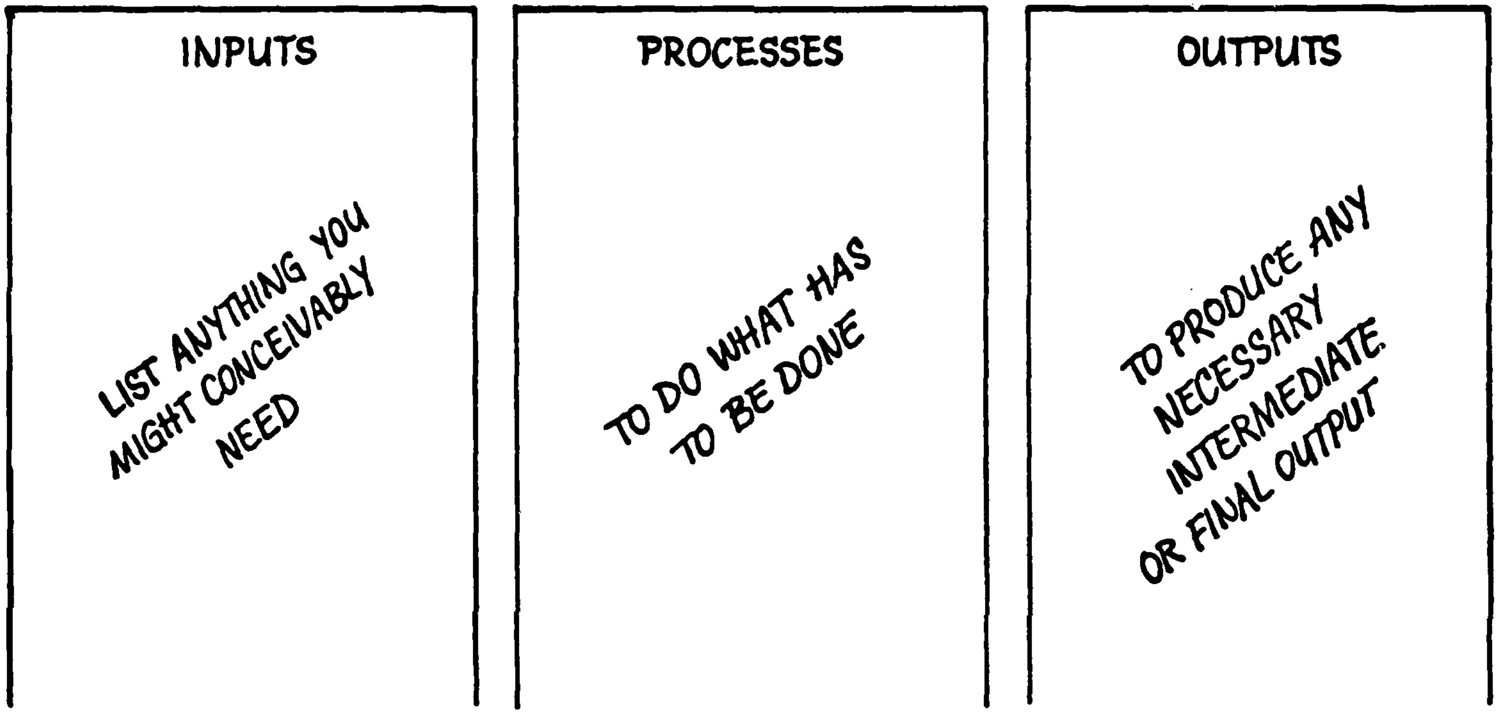


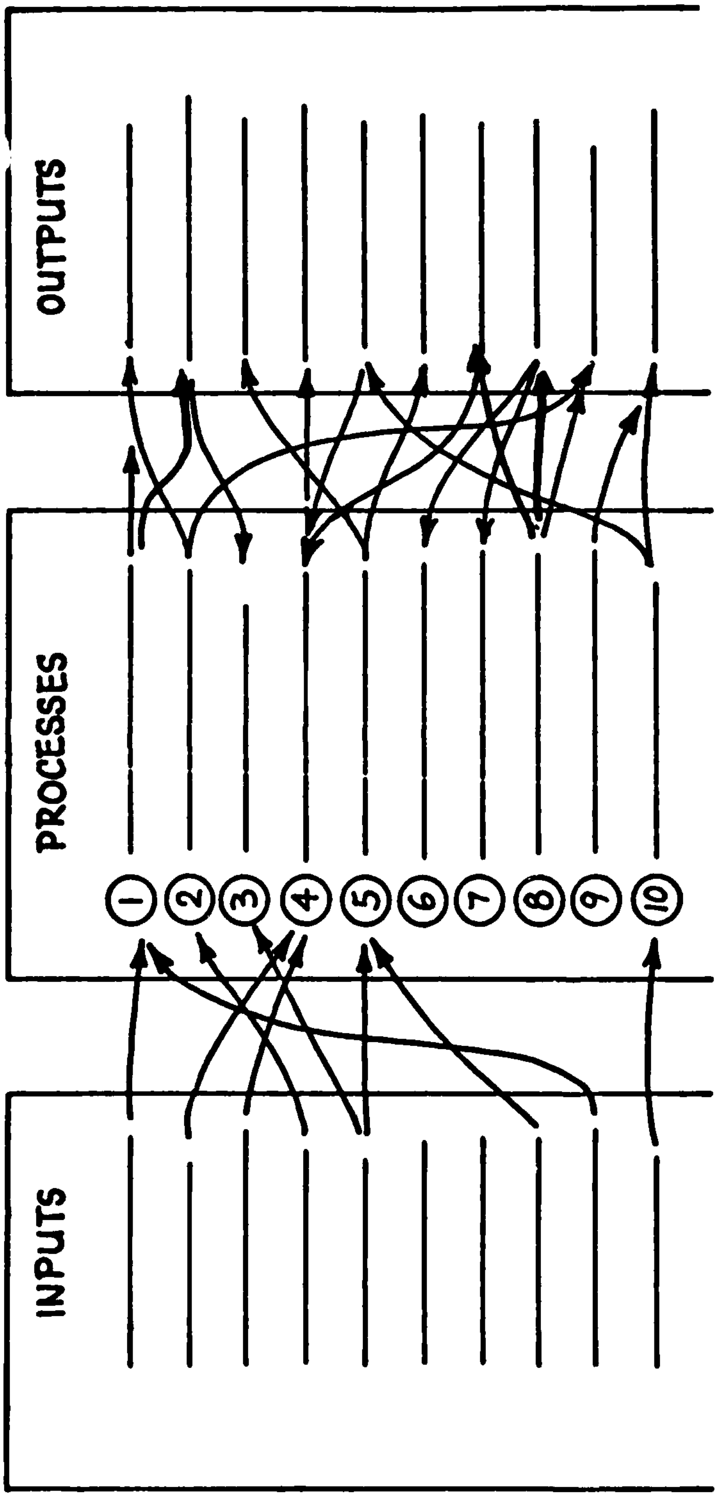
EXHIBIT IDP040-3


INPUTS
LIST ANYTHING YOU MIGHT CONCEIVABLY NEED
PROCESSES
TO DO WHAT HAS TO BE DONE
OUTPUTS
TO PRODUCE ANY NECESSARY INTERMEDIATE OR FINAL OUTPUT

- DRAW 3 OPEN-ENDED BOXES
- LIST INPUTS, PROCESSES, OUTPUTS
- CONNECT THEM

BUT DON'T WORRY ABOUT GRAPHICS

EXHIBIT IDP040-4



- DRAW 3 OPEN-ENDED BOXES
- LIST INPUTS, PROCESSES, OUTPUTS
- CONNECT THEM

BUT DON'T WORRY ABOUT GRAPHICS

EXHIBIT IDP040-5

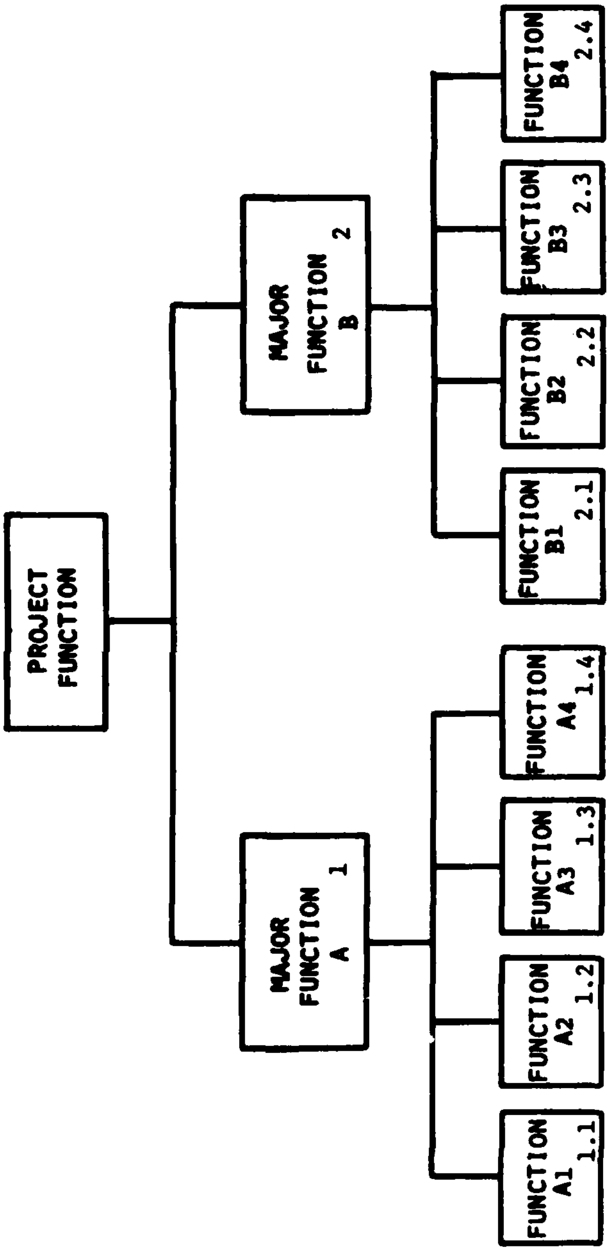
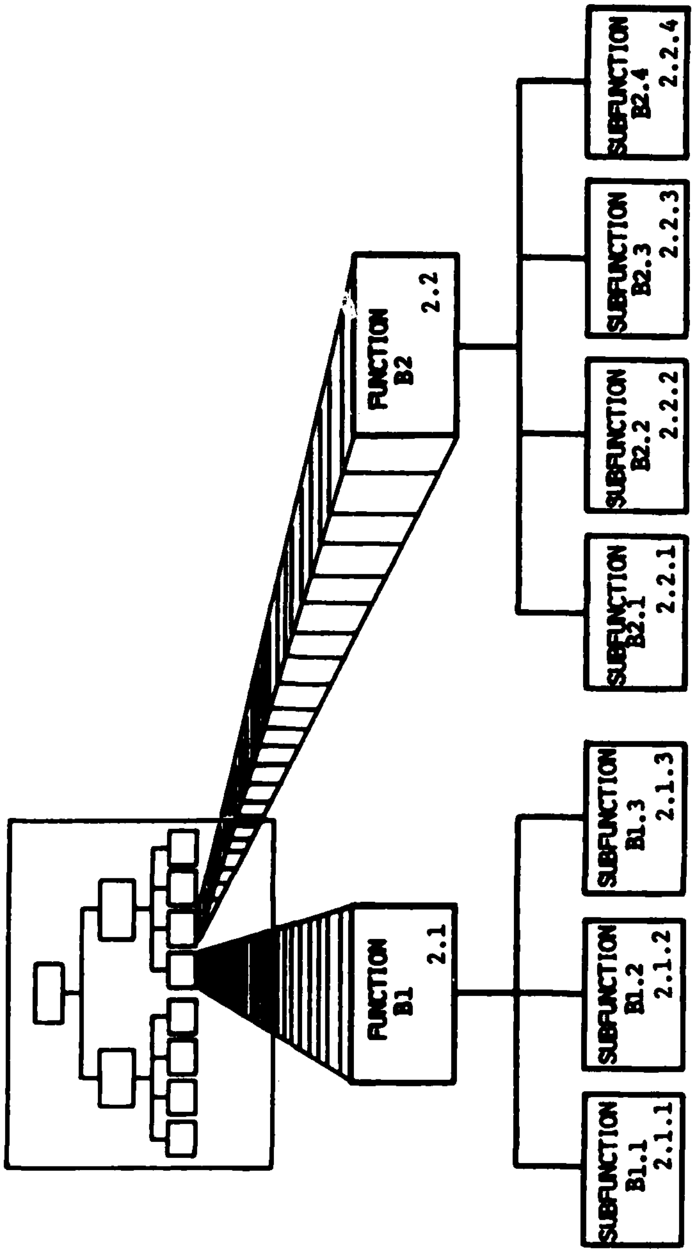
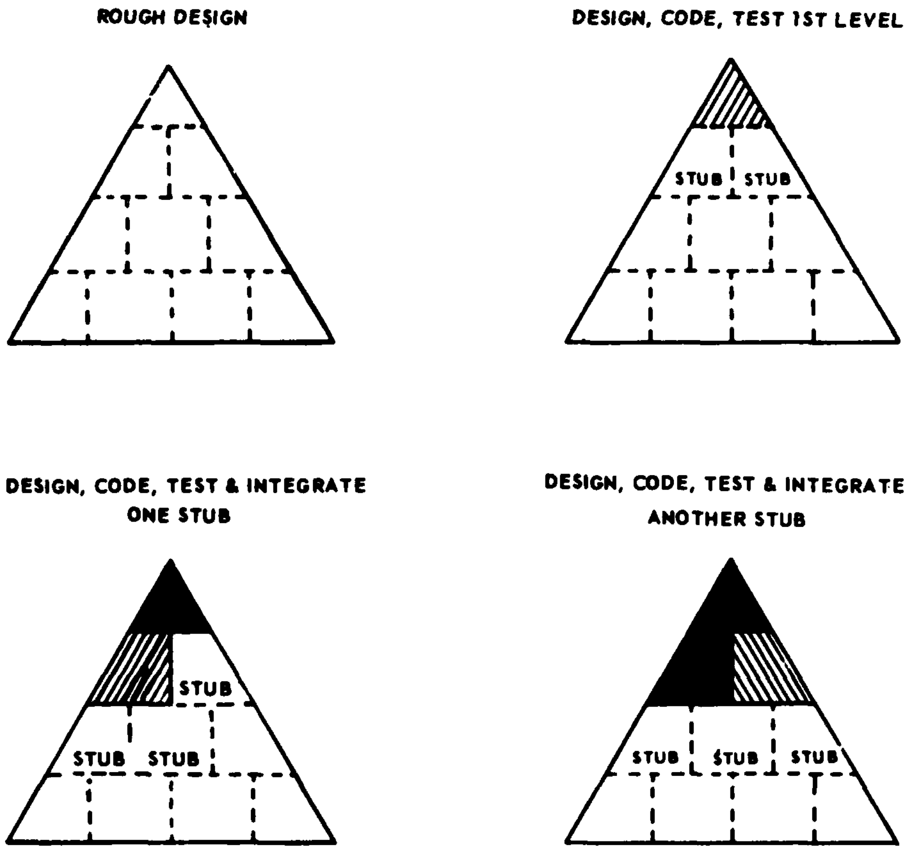


EXHIBIT IDP040-6



THE GROUP PLAN

EXHIBIT IDP040-7



ROUGH DESIGN

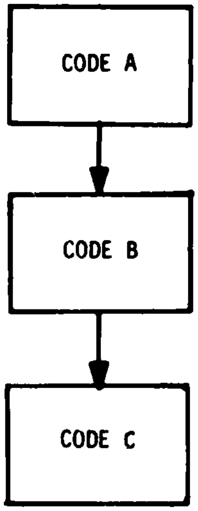
DESIGN, CODE, TEST 1ST LEVEL

DESIGN, CODE, TEST & INTEGRATE ONE STUB

DESIGN, CODE, TEST & INTEGRATE ANOTHER STUB

EXHIBIT IDP040-8

CONVENTIONAL FLOWCHART



P D L

```
CODE  A

CODE  B

CODE  C
```

IFTHENELSE

```
   IF       CONDITION-IS-TRUE
            CODE A
  ELSE
            CODE B
  ENDIF
            CODE C
```

CONVENTIONAL FLOWCHART

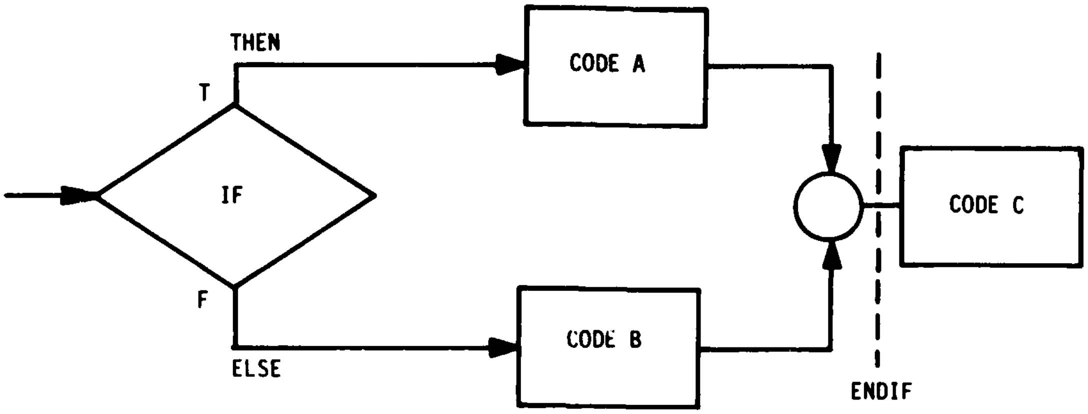
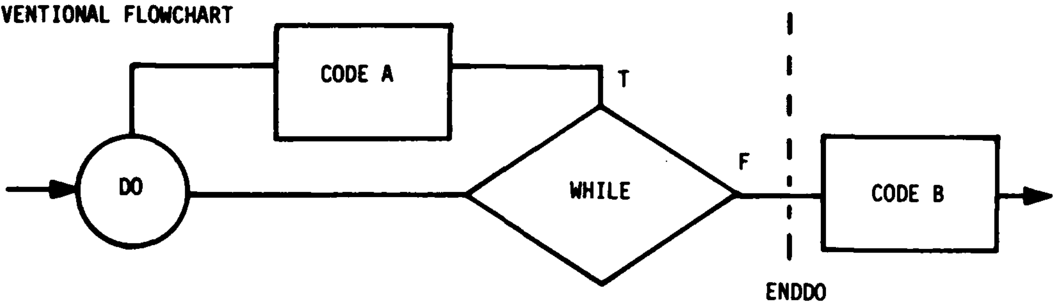


EXHIBIT IDP040-9

DOWHILE

```
DOWHILE           CONDITION-IS-TRUE
        CODE A
ENDDO
        CODE B
```

CONVENTIONAL FLOWCHART



DOUNTIL

```
DOUNTIL           CONDITION-IS-TRUE
        CODE A
ENDDO
```

CONVENTIONAL FLOWCHART

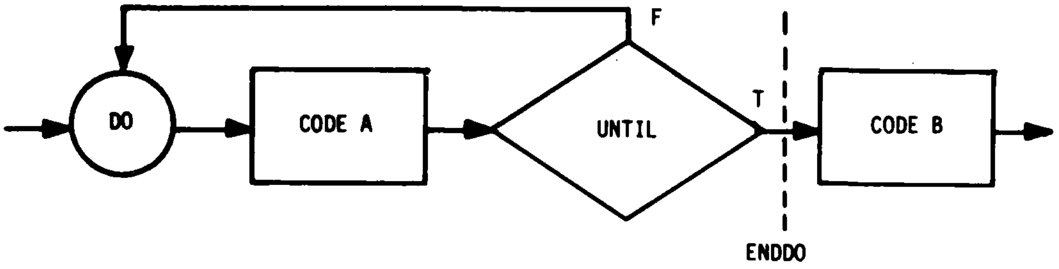


EXHIBIT IDP040-10

SECTION

1. Computer Models

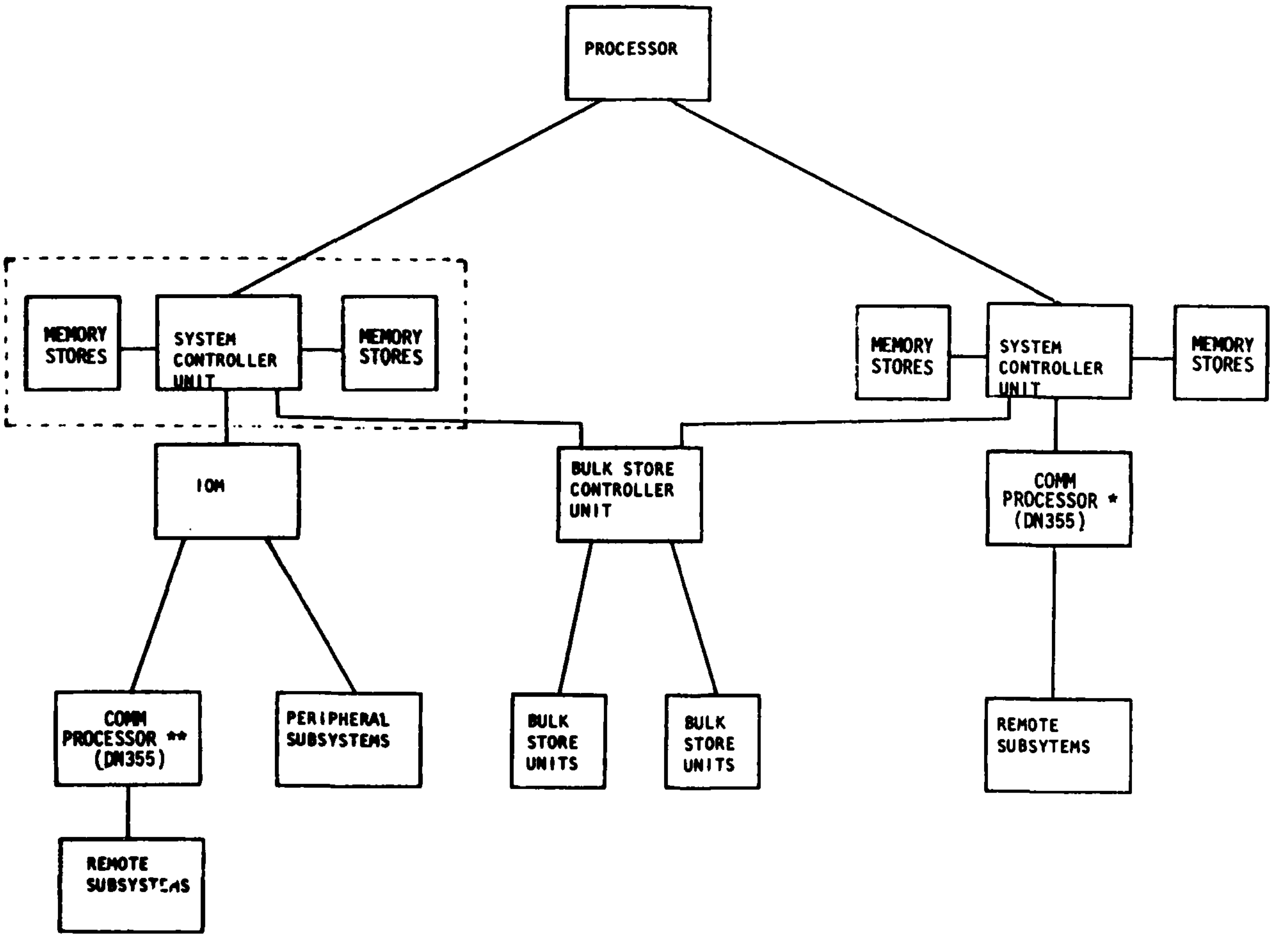
2. Input/Output Multiplexer

3. Communications Processor

4. Memory

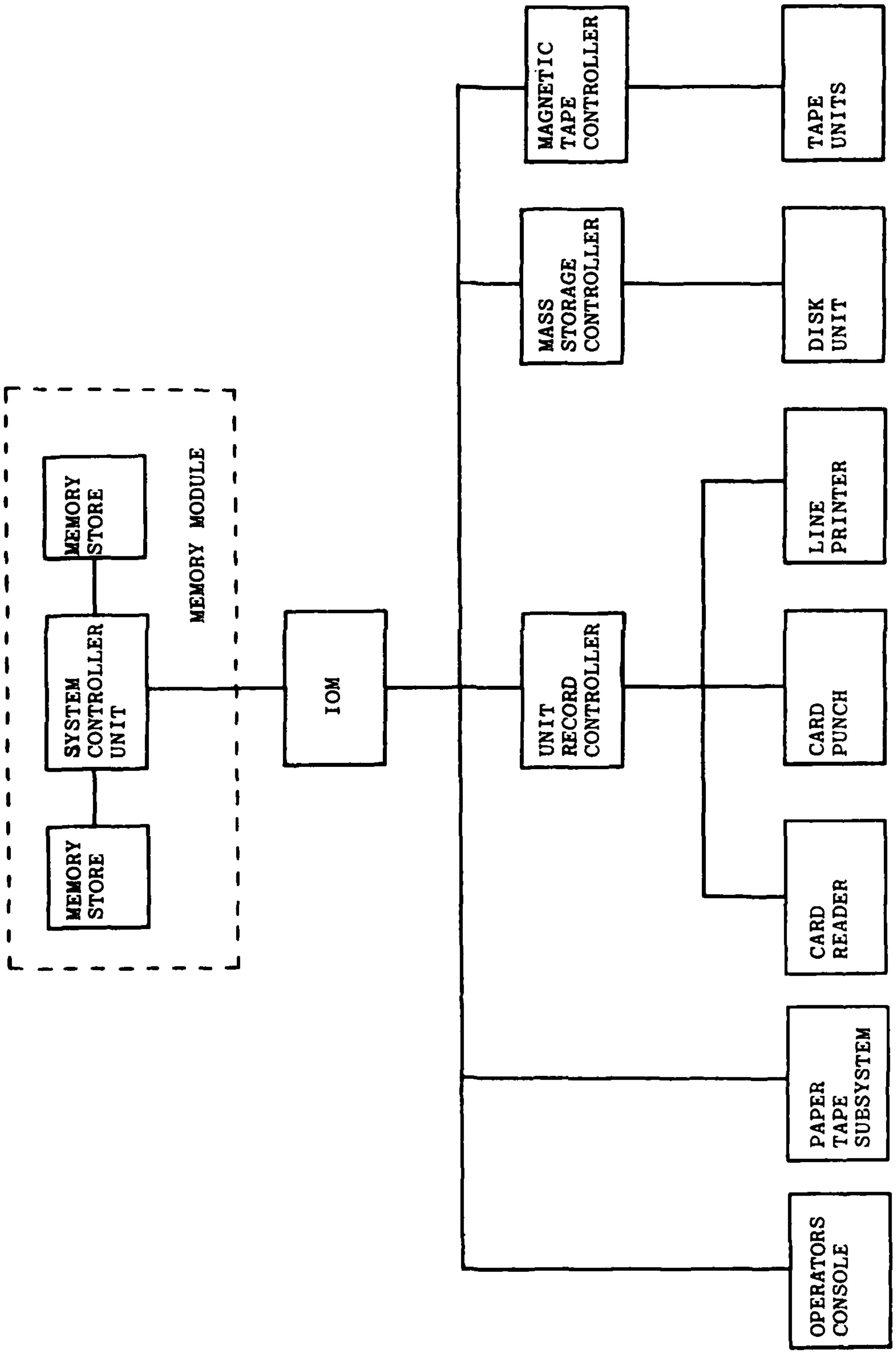
5. Processor

EXHIBIT IDP050-1



* Latest WWMCCS Configuration of the COMM Processor.
**Original WWMCCS Configuration of the COMM Processor.
NOTE: The maximum number of processor, memory, and IOM modules that can be configured at the largest WWMCCS ADP site is four each. The maximum number of COMM Processors is three.

EXHIBIT IDP050-2


MEMORY MODULE
MEMORY STORE
SYSTEM CONTROLLER UNIT
MEMORY STORE
IOM
MAGNETIC TAPE CONTROLLER
TAPE UNITS
MASS STORAGE CONTROLLER
DISK UNIT
UNIT RECORD CONTROLLER
LINE PRINTER
CARD PUNCH
CARD READER
PAPER TAPE SUBSYSTEM
OPERATORS CONSOLE

IDP050-3

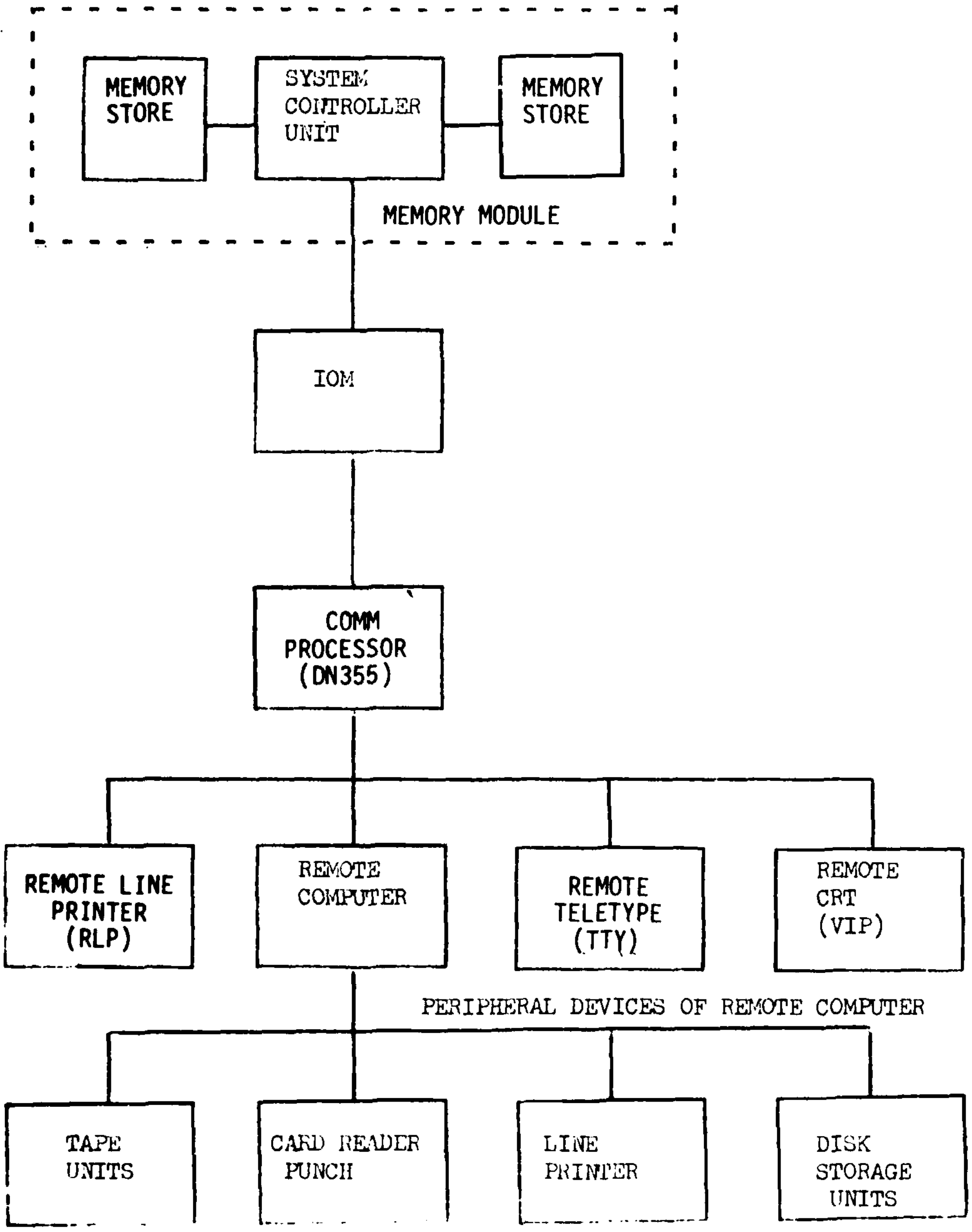


EXHIBIT IDP050-4

WWMCCS INTERCOMPUTER NETWORK (WIN) USER
(H6000-CDT) COURSE EXHIBITS

TELECONFERENCE (TLCF) TERMINAL LOGON PROCEDURES

The question-answer sequences shown below illustrate the three procedures which may be used to log onto teleconference. The sequences show are for VIP 786Ws, VIP 7705s, and teletype compatible terminals.

VIP 786W:

$6, TLCF

USERID$PASSWORD?

userid$password (Type USERID and PASSWORD separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., ZZZ, UZZ, SZZ, TZZ, etc.).)

VIP 7705:

$5, TLCF

USERID$PASSWORD?

userid$password (Type USERID and PASSWORD separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., UZZ, CZZ, SZZ, TZZ, etc.).)

EXHIBIT WIN020-1

TTY Compatible Terminal:

Hold control key while striking `A′.

PROGRAM NAME - TLCF

USERID$PASSWORD?

userid$password (Type USERID and PASSWORD separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., UZZ, CZZ, SZZ, TZZ, etc.).)

EXHIBIT WIN020-1 (Cont)

TELECONFERENCING MODES

Command Mode:

The Command Mode allows the user via commands to enter another mode, obtain information related to the conference, connect to direct access programs, send short communications, display terminal activity, and leave the conference. A participant is automatically placed in Command Mode upon joining the conference.

Listen Mode:

This is the passive state of the conference in which messages are received. A break sequence (the break key or $*$BRK for VIPs, followed by $BREAK for the Telnet users) is the only stimulus Listen Mode recognizes. It will cause the participant to be transferred to the Command Mode.

Talk Mode:

A participant enters Talk Mode by using the Command Mode command ‘TALK’. This is the message entry state of the conference. Upon completion of a message, a participant may specify the mode he wishes to enter. A message terminated with ‘$END’ or null input transfers the participant to Listen Mode. A message terminated ‘$TALK’ leaves the participant in Talk Mode. One terminated with ‘$COMMAND’ transfers the person to the Command Mode. The use of the break stimulus or ‘$DELETE’ causes the message to be cancelled and the participant to be transferred to Listen Mode.

EXHIBIT WIN030-1

Formatting Conventions

The following formatting conventions are used throughout the WIN Course:

CAPITAL LETTERS	FTS keyboards; if used must be exactly as shown.
<small letters>	Variable user-specified information. NOTE: Since lowercase letters are not available o CDTS, the carets themselves < > will be used to denote user-defined data.
Brackets [ ]	Optional word or phrase.
Braces { }	Options, one of which must be selected.
Parentheses ( )	Synonyms for FTS commands.
Ellipses ...	Information can be repeated.
Underlining	Where used, indicates the default value.

EXHIBIT WIN030-2

TALK MODE VERBS

?

The Talk Mode prompt (>) answered with a question mark will cause a brief description of Talk Mode, including all valid verbs, to be displayed. During the optional formatting of a message, the question mark can be used to request an explanation of the type of input desired.

$COMMAND

This verb marks a message as complete and transfers the terminal into Command Mode. Use of this verb during the optional formatting phase of message entry will cause all previous input to be ignored, and the message deleted. The terminal will be transferred to Command Mode.

$DELETE

This parameter deletes the message being entered and transfers the terminal to Listen Mode. Use of this verb during the optional formatting phase of message entry will cause all previous input to be ignored and the message deleted. The terminal will be transferred to Listen Mode.

$DESCRIBE verb

This verb will provide an explanation of how to use the Talk Mode verb entered as a parameter. If no parameter is given, a brief description of Talk Mode and a list of valid verbs will be provided.

$EDIT

A $EDIT prompt (-) permits changing lines in a message before it is completed through unique $EDIT commands. The commands are BACKUP, DELETE, DONE, FORWARD, INSERT, PRINT, and REPLACE. All of the commands except DONE may be abbreviated to the first letter. BACKUP, DELETE, FORWARD, PRINT, and REPLACE may be followed by a number for the number of lines the command applies to, such as B2 to back up two lines. An asterisk following DELETE, PRINT, or REPLACE applies the command to the rest of the message. F* moves the line pointer to the last line of the message. B* or B moves the line pointer to the first line of the message. A question mark following any of the $EDIT commands or prompts will obtain a description of the command (or $EDIT) and its functions. This command must be used before ending the Talk message.

EXHIBIT WIN040-1

$END

This verb marks a message as complete and transfers the terminal user to Listen Mode. A carriage return on a null line will serve the same purpose on a teletype. Use of this verb during the optional formatting phase of message entry will cause all previous input to be ignored and the message deleted. The terminal will be transferred to Listen Mode.

$INSERT catalog/file
$INSERT

This verb inserts a previously created file specified by the `catalog/file′ parameter into the current message. The file must be resident at the conference site, in TSS ASCII media code 6 format, and the lines should be no more than 80 characters in length. The participant can specify that only selected lines from the file be inserted by inputting special instructions when prompted. These take the form `TAKE n′, `TAKE REST′, `SKIP n′, and `SKIP REST′, repeated as necessary and separated by semicolons. This verb without an accompanying `catalog/file′ indicates the contents of a previous message are to be inserted into the current message. The participant will be prompted for the message number.

$LIST n
$LIST

This verb causes the last `n′ lines entered into the Talk message to be listed. If the `n′ parameter is not present, all lines are listed.

$SELECT username1; username2;...;username4

This verb allows a message to be sent only to the participant referenced by the `username′ parameter. Up to four usernames may be specified.

$SUBJECT text

This verb assigns a subject to the message being built. The text following the verb will be the subject of the message and may contain one or more keywords (see KEYWORD).

$TALK

This verb marks a message as complete and leaves the participant in Talk mode to begin another message. Use of this verb during the optional formatting phase of message entry will cause all previous input to be ignored and the message deleted. The terminal will remain in Talk mode.

COMMAND MODE COMMANDS (ALL PARTICIPANTS)

?

This command causes the user to be given a brief description of Command Mode, including available options.

BULLETIN

This command allows the participant to receive the next Talk Mode message without going to the Listen Mode.

BYE (QUIT, EXIT)

This command causes a participant to be disconnected from the conference. To participate further, the user will have to rejoin. QUIT and EXIT are synonyms for BYE.

CHAIR

A participant whose name is on the alternate chairman list may use this command to assume the chairmanship. It is only valid if there is not a current chairman.

COMMUNICATION $ALL
COMMUNICATION <username;username;...;username>

This command allows a participant to enter a 1- to 13-line message for transmittal. This message is unrecorded and is delivered to all participants specified in the delivery list regardless of their conference state (i.e., they do not have to be in Listen Mode). The delivery list is either one or more participants identified by 'USERNAMEs', or all participants ($ALL). Delivery list is limited to one line. Upon transmission of the message, a list of nonactive addressees will be displayed.

DAC program name

This command initiates a terminal line switch from TLCF to the named program, if it is active. If not, the user is notified and remains in command mode.

DEFINITION ACCESS

This command allows a participant to determine if the conference is open to joining participants. Using this command, the chairman may open or close the conference to participants who wish to join.

DEFINITION ALTERNATE

This command provides a list of these participants who may assume the conference chair when the chairman is absent. The chairman can use this command to update the list of those who may chair the conference in his absence.

DEFINITION ANNOUNCE

This command allows a participant to find out which messages currently comprise the announcement list.

DEFINITION DELIVERY

This command displays previously entered transcript delivery information (host name and $IDENT). The display is followed by a prompt inquiring if the information is to be changed. When a change is desired, the participant is prompted for new information.

DEFINITION FILE

This command allows a participant to view the conference transcript file name.

DEFINITION FLOOR

This command allows a participant to find out if floor recording is enabled. Using this command, the chairman may turn floor recording on or off.

DEFINITION INSERT

This command allows a participant to find out the current line count limit on inserted files.

DEFINITION KEYWORD

This command allows a participant to find out how many words are currently entered in the subject keyword list. The REVIEW SUBJECT command may be used to review the list of valid keywords.

DEFINITION LIST

This command allows a participant to obtain a list of the valid conference participant names. Only the chairman receives this list and the USERID associated with the individual participant names.

DEFINITION SCC

This command allows a participant to find out the classification of the conference. The chairman can use this command to change the conference classification. When the conference classification is changed, the file classification will also be changed.

DEFINITION

This command causes all information that is obtainable through the specific requests of this command (see above) to be printed at the participant's terminal. If a participant is not entitled to a specific type of information (for chairman only), he will not see it. This form cannot be used to change those parameters that are changeable.

FLOOR CANCEL

This command allows a participant who has previously requested the floor to cancel that request.

FLOOR

This command allows a participant to request the floor and optionally include one line of text. The participant will then be requested to continue with his activity until the chairman approves or disapproves the request (unless the requestor is the chairman).

GIVEUP

This command allows the floor holder to give up the floor.

HELP (DESCRIBE, MANUAL, VOCAB) command

This command will display a description of how to use the indicated command. If no parameter 'command' is entered, command mode description and the list of valid commands will be listed.

LISTEN

This command transfers the participant to Listen Mode. This command is invalid if the participant has the floor.

PDAC program name

This command logs a user onto a DAC program specified by the `program name' parameter. This parameter must be the name of a direct access program (e.g., TSS, TPE, TELNET, etc.). The connection to the DAC program is severed and the user returned to Command mode by using the DAC program disconnect command (e.g., BYE in TS1) or by inputting `^C' (up-arrow C) when the terminal is awaiting input. The connection can be suspended by responding `^S' (up-arrow S) when the terminal is awaiting input. To resume the connection at the point it was suspended, the participant would use the command mode verb PDAC with a blank program field. The receipt of messages from the conference while PCAC'd can be suspended by inputting '^N' (up-arror N) when the terminal is awaiting input. Receipt of these messages can be resumed by inputting `^P` (up-arrow P) when the terminal is awaiting input.

PRINT n;n;...;n ON id
PRINT n;n;...;n

This command causes conference messages to be printed on the remote line printer specified by the `id' parameter. If no `id' parameter is present, the conference messages will be printed on the main system printer. The numbers to be printed. Up to 20 messages may be specified. A prompt will be given for host name and $IDENT.

PRINT n-m ON id
PRINT n-m
PRINT n THRU m ON id
PRINT n THRU m

This command causes those messages in the range `n' through `m' to be printed on the remote line printer specified by the `id' parameter. If no`id' parameter is present, the conference messages will be printed on the main system printer. A prompt will be given for host name and $IDENT.

```
PRINT ALL BY username ON id        PRINT LAST n BY username ON id
PRINT ALL BY username              PRINT LAST n BY username
PRINT ALL ON id                    PRINT LAST n ON id
PRINT ALL                          PRINT LAST n
```

This command causes conference messages to be printed on the remote line printer specified by 'id' parameter. If no 'id' parameter is present, the conference messages will be printed on the main system printer. If the 'username' parameter is present, only messages entered by that user will be printed. If the 'LAST n' parameter is present, the last n messages will be printed. A prompt will be given for host name and $IDENT.

```
PRINT EVENTS ON id
PRINT EVENTS
```

This command causes recorded conference events (e.g., participant signon, signoff) to be printed on the remote line printer specified by the 'id' parameter. If no 'id' parameter is present, the conference messages will be printed on the main system printer. A prompt will be given for host name and $IDENT.

```
PRINT FLOOR FROM hhmm TO hhmm ON ddmmmyy ON id
PRINT FLOOR FROM hhmm TO hhmm ON ddmmmyy
PRINT FLOOR FROM hhmm TO hhmm ON id
PRINT FLOOR FROM hhmm TO hhmm
PRINT FLOOR ON ddmmmyy ON id
PRINT FLOOR ON ddmmmyy
PRINT FLOOR ON id
PRINT FLOOR
```

This command causes the recorded conference floor activity to be printed on the remote line printer specified by the 'id' parameter. If no 'id' parameter is present, the conference messages will be printed on the main system printer. If the 'ON ddmmmyy' parameter is present, only floor activity for the specified date will be printed. If the 'ON ddmmmyy' parameter is not present, today's date will be assumed. If the 'FROM hhmm TO hhmm' parameter is present, only floor activity between the specified times will be displayed. A prompt will be given for host namd and $IDENT.

```
PRINT MESSAGES BY username ON id
PRINT MESSAGES BY username
PRINT ESSAGES ON id
PRINT MESSAGES
```

This command is the same as PRINT ALL above.

EXHIBIT WIN050-1 (Cont)

PRINT SUBJECT keyword ON id
PRINT SUBJECT keyword

This command causes previously entered conference messages to be printed on the remote line printer specified by the 'id' parameter. Only those messages with the specified 'keyword' will be printed. If no 'id' parameter is present, the conference messages will be printed on the main system printer. A prompt will be given for host name and $IDENT.

REVIEW ALL BY username
REVIEW ALL
REVIEW LAST n BY username
REVIEW LAST n
REVIEW n;n;...;n
REVIEW n-m
REVIEW n THRU m

This command causes previously entered conference messages to be printed at the requestor's terminal. If the 'username' parameter is present, only those messages entered by that user will be printed. If the 'LAST n' parameter is present, the last n messages will be printed. The 'n;n;...;n' parameter allows the user to specify a message or series of message numbers to be printed. Input is limited to one line. The 'n-m' and 'n through m' parameters allow the user to specify a range of messages to be printed.

REVIEW SUBJECT keyword

This command causes previously entered conference messages to be printed at the requestor's terminal. Only those messages with the specified 'keyword' parameter is given, the list of possible subject keywords will be printed at the user's terminal.

REVIEW FLOOR FROM hhmm TO hhmm ON ddmmmyy
REVIEW FLOOR FROM hhmm TO hhmm
REVIEW FLOOR ON ddmmmyy
REVIEW FLOOR

This command causes conference floor activity to be printed at the requestor's terminal. If the 'ON ddmmmyy' parameter is present, only floor activity for the specified date will be displayed. If the 'ON ddmmmyy' parameter is not present, today's date will be assumed. If the 'FROM hhmm TO hhmm' parameter is present, only floor activity between the specified times will be displayed.

STATUS

This command is designed to provide general information about the conference. This information includes the full conference name, current time and date, host name, conference chairman's name, initiation time and date, if the floor is occupied, and if so, by whom, and a list of active participants including each participant's name, last joining time and date, and current mode.

TALK

This command transfers the participant to Talk Mode.

COMMAND MODE COMMANDS (CHAIRMAN ONLY)[*]

ADD userid/username

This command allows chairman to add to the valid participant list the participant specified by `userid/username' parameter. The `username' portion of the parameter must be unique, i.e., cannot be assigned to another userid. The chairman will be asked if he wishes to change the `userid' portion when the `username' portion is not unique but the `userid' portion is different.

ADJOURN

This command prevents new users from signing on, and if entered after all other participants have signed off, allows chairman to adjourn a conference.

ANNOUNCE n;n;...;n

This command allows chairman to specify or change the list of message numbers to be displayed to participants when they join the conference. The `n;n;...;n' parameters are optional and represent message numbers. If no message numbers are input, no messages will be displayed. A maximum of 10 messages may be announced.

CHAIR TO username

This command allows the current chairman to give the chair to another participant specified by the `username' parameter. The initiating chairman may reclaim the chair by entering `CHAIR'.

DELETE username
DELETE

This command without the `username' parameter will display those usernames which have not been used to access the conference. The `username' parameter is used to delete a name which has not been used from the participant list.

ERASE n

This command is used to erase the contents of a message from the transcript file. The `n' is the message number to be erased.

FLOOR FREE

This command allows the chairman to take the floor from the floor holder and leave it open.

[*] Chairman commands are available to any participant who has the c.air.

FLOOR TO username

This command displays to the named user a message stating that the chairman would like him to accept the floor. If the named user accepts, he will immediately be assigned the floor.

FORMAT CANCEL
FORMAT options[*]
FORMAT

This command is used to request a specific question/answer sequence be initiated for Talk mode messages. This sequence provides for a fixed format introduction to each Talk mode message. The options available will be listed upon detection of the command if they are not part of the request. The options available are: TO, FROM, SUBJECT, INFORMATION, REFERENCE, DATE-TIME-GROUP (DTG), and CLASSIFICATION. The CANCEL option is used to return talk messages to free form.

INSERT ELIMINATE
INSERT n

This command allows the chairman to change the maximum size file that may be inserted as a message. 'ELIMINATE' will eliminate the limit; n is a new maximum line limit (n $\leq$ 600).

KEYWORD

This command allows the chairman to create or add to a subject list by which Talk messages may be categorized for later review or print. No parameters are required.

TCALL

This command terminates the terminal connection to a conference of the named user (e.g., TCALL TOM) and can only be used by the chairman.

TERMINATE

This command is used by the chairman to prevent new users from signing on, and if entered when no other participants are signed on, to terminate the conference. Reconvening will not be possible if the conference is terminated.

[*] The user will automatically be prompted for these options when entering a talk mode message. Continuation lines may be entered when prompted by a dash (-). A null response will end th input to a particular option. When a response to all options has been given, the actual text of the message is entered in response to the talk mode prompt.

CONSOLIDATED LISTING OF TELECONFERENCING COMMANDS

Talk Mode Verbs

```
?
$COMMAND
$DELETE
$DESCRIBE verb
$EDIT
$END
$INSERT catalog/file
$INSERT
$LIST n
$LIST
$SELECT username1;username2;...;username4
$SUBJECT text
$TALK
```

Command Mode Commands (Chairman Only)

```
ADD userid/username
ADJOURN
ANNOUNCE n;n;...;n
CHAIR TO username
DELETE username
DELETE
ERASE n
FLOOR FREE
FLOOR TO username
FORMAT CANCEL
FORMAT option
FORMAT
INSERT ELIMINATE
INSERT n
KEYWORD
TCALL
TERMINATE
```

EXHIBIT WIN060-1 (Cont)

Command Mode Commands (All Participants)

```
?
BULLETIN
BYE (QUIT, EXIT)
CHAIR
COMMUNICATION $ALL
COMMUNICATION username;username;...;username
DAC program name
DEFINITION ACCESS
DEFINITION ALTERNATE
DEFINITION ANNOUNCE
DEFINITION DELIVERY
DEFINITION FILE
DEFINITION FLOOR
DEFINITION INSERT
DEFINITION KEYWORD
DEFINITION LIST
DEFINITION SCC
DEFINITION
FLOOR CANCEL
FLOOR
GIVEUP
HELP (DESCRIBE, MANUAL, VOCAB) command
LISTEN
PDAC program name
PRINT n;n;...;n ON id
PRINT n;n;...;n
PRINT n-m ON id
PRINT n-m
PRINT n THRU m ON id
PRINT n THRU m
PRINT ALL BY username ON id
PRINT ALL BY username
PRINT ALL ON id
PRINT ALL
PRINT LAST n BY username ON id
```

```
PRINT LAST n BY username
PRINT LAST n ON id
PRINT LAST n
PRINT EVENTS ON id
PRINT EVENTS
PRINT FLOOR FROM hhmm TO hhmm ON
  ddmmmyyid ON id
PRINT FLOOR FROM hhmm TO hhmm ON
  ddmmmyy
PRINT FLOOR FROM hhmm TO hhmm ON id
PRINT FLOOR FROM hhmm TO hhmm
PRINT FLOOR ON ddmmmyy ON id
PRINT FLOOR ON ddmmmyy
PRINT FLOOR ON id
PRINT FLOOR
PRINT MESSAGES BY username ON id
PRINT MESSAGES BY username
PRINT MESSAGES ON id
PRINT MESSAGES
PRINT SUBJECT keyword ON id
PRINT SUBJECT keyword
REVIEW ALL BY username
REVIEW ALL
REVIEW LAST n BY username
REVIEW LAST n
REVIEW n;n;...;n
REVIEW n-m
REVIEW n THRU m
REVIEW SUBJECT keyword
REVIEW FLOOR FROM hhmm TO hhmm ON ddmmmyy
REVIEW FLOOR FROM hhmm TO hhmm
REVIEW FLOOR ON ddmmmyy
REVIEW FLOOR
STATUS
TALK
```

EXHIBIT WIN060-1 (Cont)

TELNET TERMINAL LOGON PROCEDURES

The question-answer sequences shown below illustrate the three procedures which may be used to log onto TELNET. The sequences shown are for VIP 786Ws, VIP 7705s, and teletype compatible terminals.

VIP 786W:

$6,TELNET

USERID$PASSWORD?

userid$password (Type userid and password separated by dollar sign ($).)

IDENT?

xxxxx (Enter valid account number and user identifier.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., UZZ, CZZ, SZZ, TZZ, etc.).)

VIP 7705:

$5,TELNET

USERID$PASSWORD?

userid$password (Type userid and password separated by dollar sign ($).)

IDENT?

xxxxx (Enter valid account number and user identifier.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., UZZ, CZZ, SZZ, TZZ, etc.).)

EXHIBIT WIN070-1

Teletype Compatible Terminal:

Hold Control Key while striking 'A'.

PROGRAM NAME - TELNET

USERID$PASSWORD?

userid$password (Type userid and password separated by dollar sign ($).)

IDENT?

xxxxx (Enter valid account number and user identifier.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code (i.e., UZZ, CZZ, SZZ, TZZ, etc.).)

EXHIBIT WIN070-1 (Cont)

TELNET INPUT MODES

Command Mode

Telnet users are placed in command mode immediately upon connecting a terminal to the Telnet program. While in command mode, all terminal input is interpreted as a Telnet command. After processing the command, Telnet will normally print a prompt character (default prompt character is `@') requesting another command. If the command entered from the terminal causes Telnet to successfully establish a network connection to a direct access (DAC) program (i.e., TSS, TPE, TLCF, etc.), the Telnet user is placed in Telnet remote mode.

Remote Mode

While in *remote mode*, terminal input will be transmitted to the terminal accessible program to which the Telnet user has connected. The only Telnet commands recognized in remote mode are the break key (or $*$B) and $BREAK. Depressing the break key causes the Telnet user to be placed in Telnet command mode (See Break Key). Entering $BREAK in response to a request for input from the remote DAC program causes a break to be transmitted to the remote DAC program (see $BREAK). The Telnet user may enter Telnet Command mode by activating the break key or by logging off the remote program.

TELNET COMMANDS[*]

BREAK KEY

Depressing the break key at a TTY compatible terminal or typing $*$B at a VIP terminal causes Telnet to return the Telnet prompt sign (@) in the local Telnet command mode. If a break is initiated while a remote host connection is open, the terminal will return to the local TELNET command mode, but the connection to the remote host DAC program will remain open. To return to that same remote DAC program, the user must use the $BREAK or CONTINUE Telnet commands.

$BREAK

This command, whether entered in Telnet command or remote mode (when terminal is awaiting user input) causes Telnet to transmit a break to the remote DAC program. The character string that comprises this command may be changed by the user (see BREAK below).

[*] The full form of each command is shown. The user need only input the first four characters for a command to be recognized.

BREAK $xxx

This command allows the Telnet user to change the command mode break string from the default '$BREAK' to a string specified by the parameter '$xxx'.

BYE (EXIT, QUIT)

This command causes Telnet to close the open network connection and log the user off Telnet. EXIT and QUIT are synonyms for BYE.

CHOLD

This command causes Telnet to cancel a previous HOLD command.

CLOSE

This command causes Telnet to close the open network connection.

CONTINUE

If a condition exists, this command will place the Telnet user in remote mode. If a $*$B was initiated to return the user's terminal to Telnet, and the connection is still open, this command will return the terminal to the remote host at the point the $*$B interrupted it.

DAC program name

This command causes Telnet to close any network connection that may be open for this user and switch the user terminal to the local host DAC program (i.e., TS1, TPE, TLCF, etc.) specified by the 'program name' parameter, if it is active. If not, the user terminal will remain in command mode.

HELP (DESCRIBE, MANUAL, VOCAB) command

If the optional 'command' parameter is not present, this command causes Telnet to list the Telnet command vocabulary. If the 'command' parameter is present, Telnet displays a description of that command and associated parameters. DESCRIBE, MANUAL, and VOCAB are synonyms for HELP.

HOLD

This command causes Telnet to maintain the network connection to a remote host when the remote DAC program (i.e., TS1, TPE, etc.) logs off the user.

HOSTS hostl;host2;...;hostn
HOSTS

This command will cause Telnet to display the status of the hosts on the WIN. The status will include which hosts are currently active and the WIN software (NCP, FTS, TELNET, TLCF) in execution at the particular host. When the command is followed by a host name(s), the status of only the specified host(s) will be displayed. This status is not "realtime", but represents the most recent information available.

OPEN host/program name

This command causes Telnet to establish a network connection to the host specified by the `host′ parameter and to log the user′s terminal onto the remote DAC program specified by `program name′. Telnet will use the USERID specified during local LOGON processing for remote LOGON processing.

OPENH host/program name

This command causes the same actions as the OPEN command except if the remote DAC program should log off the user′s terminal, the network connection to the remote host will be maintained. The hold option is cancelled by the CHOLD command.

PROMPT x

This command allows the Telnet user to change the command mode prompt character from the default ′@′ to a character specified by the parameter ′x′.

STATUS

This command causes Telnet to display the current state of a Telnet connection, including remote host name and remote DAC program name.

Telnet Commands (Consolidated Listing)

Break Key
$BREAK
BREAK $xxx
BYE (EXIT, QUIT)
CHOLD
CLOSE
CONTINUE
DAC program name
HELP (DESCRIBE, MANUAL, VOCAB) [command]
HOLD
HOSTS [hostl; host2;...;hostn]
OPEN host/program name
OPENH host/program name
PROMPT x
STATUS

EXHIBIT WIN070-2 (Cont)

CDTS WIN080 Simulation

OBJECTIVE: To use Teleconferencing commands taught in Lessons WIN020 through WIN060 and TELNET Log-on from Lesson WIN070.

NOTE: You must perform these steps in order.

1. Use the following information to log-on to TELNET and open a connection to Teleconferencing at LANT.

Log-on	$5,TELNET
USERID	CDTS
PASSWORD	WIN
IDENT	PWIN
Classification of Output	Unclassified
Conference Name	CONFTEST
Participant's Name	WINNER
Host for Delivery of Transcript	NMCC
IDENT	PWIN

2. Once you have joined the conference, take note of which messages are flagged as announcements by the chairman. Find a list of all active members in the conference. Review all messages before proceeding to the Talk Mode. (See available commands in step 4.)

3. Build a message in the Talk Mode concerning a subject that should be of interest to all conference members. Return to the Command Mode.

4. For this section, the following commands are available:

```
REVIEW SUBJECT                        DEFINITION
REVIEW ALL                            DEFINITION ALTERNATE
REVIEW SUBJECT keyword                DEFINITION ANNOUNCE
REVIEW n (n is an integer)            DEFINITION KEYWORD
TALK                                  COMMUNICATION $ALL
STATUS                                CHAIR
HELP                                  BYE
```

Using these commands, find the following pieces of information:

How many keywords there are.
What the keywords are.
Name of the transcript file.
Who the alternate chairmen are.

5. Send a communication to all participants stating any problems you have encountered. (For purposes of improving this CDT lesson, you must keep each line short; maximum 30 characters per line.)

6. At this point, enter the command to assume the chairmanship.

*** STEPS 7 AND 8 ARE OPTIONAL ***

EXHIBIT WIN080-1

7. For this section, the following commands are available for your use.

KEYWORD
FORMAT
TALK
LISTEN
HELP

DEFINITION
CHAIR
STATUS
DEFINITION KEYWORD
REVIEW SUBJECT

Once chairman, add the word WEATHER as a keyword. The impose formatting as follows:

TO:
FROM:
SUBJECT:

8. Finally, send a Talk Mode message to all participants with WEATHER as a keyword in the subject line. Your message should state that you must leave early because a severe thunderstorm is threatening. After sending the message, give the chairmanship to Chris.

9. Log off TLCF and TELNET.

EXHIBIT WIN080-1 (Cont)

FILE TRANSFER SERVICE (FTS) TERMINAL LOGON PROCEDURES

The question answer sequences shown below illustrate the three procedures which may be used to log onto FTS. The sequences show are for VIP 786Ws, VIP 7705s, and Teletype compatible terminals.

VIP 786W:

$6,FTS

USER$PASSWORD?

user$password (Type userid and password separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ, TZZ, etc.)

CLASSIFICATION OF FILES YOU WILL CREATE?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ, TZZ, etc.)

VIP 7705:

$5,FTS

USERID$PASSWORD?

userid$password (Type userid and password separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ TZZ, etc.)

CLASSIFICATION OF FILES YOU WILL CREATE?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ, TZZ, etc.)

EXHIBIT WIN090-1

TTY Compatible Terminal:

Hold control key while striking 'A'.

PROGRAM NAME - FTS

USERID$PASSWORD?

userid$password (Type USERID and PASSWORD separated by dollar sign $.)

IDENT?

xxxxx (Enter valid account number and user identification.)

CLASSIFICATION OF YOUR OUTPUT?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ, TZZ, etc.)

CLASSIFICATION OF FILES YOU WILL CREATE?

xxx (Enter valid security classification code i.e., UZZ, CZZ, SZZ, TZZ, etc.)

FTS COMMANDS: WIN INTERACTIVE SERVICE LANGUAGE (WISL) (WITH SYNONYMS)

APPEND

Transfers a file from one location to another within the WIN community. The origin file defined in the origin phrase is appended to the end of the last file indicated in the target phrase. If the target is not defined, the file is created.

BYE (EXIT, QUIT)

Terminates the current FTS session, gracefully logs off the user, and clears the screen of a VIP.

CANCEL (KILL)

Terminates the execution of a previously entered WISL command or command stream.

CHECK

Submits the indicated WISL command stream file to the RUN compiler for syntax checking only.

DAC

Performs a terminal line switch from FTS to the named direct access program, if it is active. If not, the line switch is not performed.

DELETE (REMOVE)

Deletes a specified catalog or file.

EDIT (ED, TS1, BUILD)

Performs a line switch to transfer the FTS user to time-sharing (TS1). To return, the user must use the TSS command "JDAC FTS".

END (EXIT)

Indicates the logical end of a WISL command stream file (optional command). Can be used in FTSBIP programs.

EXECUTE

Initiates execution of the command in the work space. If the command to be executed contains no syntax errors, the user is returned a unique Command ID (CMID) and execution begins.

HELP (DESCRIBE, MANUAL, VOCAB)

Provides the FTS user with WISL vocabulary and information regarding the WISL command structure.

IF

Permits execution of WISL commands in a command stream file based on the success or failure of previously executed WISL commands in the same command stream. Can be used in FTSBIP programs.

MAIL

Permits a user to send messages to specified user mailboxes at various WIN sites. The mailboxes may then be queried with the READMAIL command.

MODIFY

Modifies (changes) a command name and/or origin file name without reinitializing the current command parameter values.

NETJOB

Serves for remote job entry (RJE). It receives a job file from a given host, submits the job for execution at a second host, directs the output to the indicated locations, and sends the first $MSG2 in the jobs to the FTS mailbox of specified USERIDs.

READMAIL

Invokes an FTS subsystem which allows the user to access and manipulate messages in his own uniquely identified mailbox. The user can selectively list messages at a terminal or the local system printer. These messages can then be purged from a user's mailbox or saved on a file. (See READMAIL commands.)

REPLACE (COPY)

Transfers a file from one host to another, with the origin file replacing the target file. If the target file does not exist, a file will be created with the target file name.

RESTART

Causes execution of a previously suspended WISL command to be resumed.

REVIEW (ECHO)

Has two functions: If a CMID is not supplied, the current command in the work area is displayed. If a CMID is supplied, the indicated WISL command is displayed at the user's terminal, if the command is in the system.

RUN

Submits a referenced WISL command stream file for execution. May also be used when setting up an FTSBIP job with the INLINE option.

SAVE

Appends the command in the work area to a permanent file for later use. The command is saved as an ASCII character string.

SEND (MOVE, TRANSFER, RELOCATE)

Transfers a file from one location to another within the WIN community. The data transferred is used to create a file at the target location. If the target file already exists, the command is aborted.

SKIP (GO)

Permits forward branching within a WISL command stream. Can be used in FTSBIP programs.

STATUS (PS)

Provides an FTS user with a wide range of information about the status of commands under his USERID and the status of the network.

SUSPEND (SLEEP)

Suspends the execution of a WISL command. The command remains suspended until a RESTART command is entered.

WAIT

Delays the execution of a WISL command stream until the passage of a specified length of time or time of day or until the termination of the preceding WISL commands in the same command stream. Can be used in FTSBIP programs.

EXHIBIT WIN090-1 (Cont)

READMAIL COMMANDS

GO

Resumes the listing of messages after an End-of-Message-Pause caused by the EOMP option for LIST.

LIST

Displays the specified messages from the user's mailbox. "ALL" must be used if listing a mail message that has been previously scanned or listed.

OUTPUT

Causes the specified messages to be written to the specified device.

PURGE

Purges the specified messages from the user's mailbox. However, purge does not begin until after QUIT ends the read session.

QUIT

Ends a READMAIL session and returns the user to the FTS command mode.

RSAVE

Saves the specified messages in ASCII on the specified file. These messages will replace the previous contents of the file.

SCAN

Reviews messages by host, author, subject, time, or text contents. "ALL" must be used if scanning a mail message that has been previously scanned or listed

EXHIBIT WIN100-1

READMAIL COMMAND FORMATS

```
GO

LIST  {<number range>}    [EOMP]  +

OUTPUT <number range> [ONLINE        ]
                      [              ]
                      [<tape number> ]

PURGE <number range>   *

QUIT

RSAVE <number range> <catalog/file string>

     [                              [FROM <host>          ]]
     [                              [ID = <userid>        ]]
SCAN [<number range>                ["<text>"             ]]
     [                              [<SUBJECT>            ]]
     [                              ["<subject line>"     ]]
     [                              [TIME= <time>         ]]
```

+ NOTE: EOMP = End of Message Pause. This will cause readmail messages to be displayed with pauses inserted between them. The "GO" option must be used to display the next message in the number range.

*NOTE: PURGE can be entered at any time during use of READMAIL, but the command will not be executed until the QUIT command is entered.

MAIL COMMAND FORMATS

```
                       { <user 1> [AT <host 1>]          }
     { <file 1> }      { {BRDCST   }                     }
MAIL { INLINE   }  TO  { {BROADCAST}    <file>           }
     { *        }      { {B        }                     }
                       { OPERATOR [AT <host 2>]          }

[    { <user> [AT <host 3>]  } ]
[    { {BRDCST}              } ]
[ CC { {BRDCST}<file 2>      } ]   [SUBJECT [IS] "<text>"]
[    { {B     }              } ]
```

EXHIBIT WIN100-1 (Cont)

```
APPEND <origin> TO <target> [FORMAT [IS] <format>]

                        ( <file> [AT <host>]            )
DELETE (REMOTE          { ( BRDCST    )                 }
                        { { BROADCAST } <file>          }
                        ( ( B         )                 )

REPLACE (COPY) <target> WITH <origin>
                    [FORMAT [IS] <format>]

SEND (MOVE, TRANSFER, RELOCATE) <origin> TO <target> [FORMAT [IS] <format>]

                     ( <file name> [AT <hostname>]       )
                     { { TAPE7 } [NUMBER <tapeid>d       }
<origin>   =         { { TAPE9 } [<tape options>]        }
                     {           [AT <host>]             }
                     {                                   }
                     { INLINE                            }
                     (  *                                )

                     ( <file name> [AT <hostname>]       )
                     { { TAPE7 } [NUMBER] <tapeid>u      }
                     { { TAPE9 }                         }
<target>   =         {          [<tape options>]         }
                     {           [AT <host>]             }
                     { { BRDCST    }                     }
                     { { BROADCAST }  <file>             }
                     ( { B         }                     )

                     [ { PHYSICAL       SEQUENTIAL } ]
<format>   =         [ { PHYS           SEQ        } ]
                     [ { LOGICAL        RANDOM     } ]
                     [ { LOG               RAN     } ]

                     [ { BLKSIZ    }  <BLOCK SIZE> ]
                     [ { BLOCKSIZE }               ]

                        [RECSIZE <record size>]

                     [ { MAXIMUMSIZE } <maximum file size> ]
                     [ { MAXSIZ      }                     ]

                     [ { FILESIZE }    <current file size> ]
                     [ { FILSIZ   }                       ]

                     [  STANDARD      ]
                     [ { NONSTANDARD }]
                     [ { NONSTD      }]

                      [ LABELS   ]   [ BINARY ]     [ DG        ]
                      [ NOLABELS ]   [ BCD    ]     [ DOWNGRADE ]
                                     [ ASCII  ]
```

EXHIBIT WIN110-1

SERVICE COMMANDS

```
BYE (EXIT, QUIT)

CANCEL (KILL) <cmid>

CHECK <command stream file>

DAC <program name>

EDIT (ED, TSS, BUILD)

EXECUTE

HELP (DESCRIBE, MANUAL, VOCAB, ?) [<keyword>]

MODIFY <string>

READMAIL

RESTART <cmid>

REVIEW (ECHO) [<cmid>]

RUN <file>

SAVE <file>

                ┌                                                      ┐
                │ LIMITS                ┌ <display> ┐                  │
                │ [<userid,][,cmid>]    │ MONITOR   │                  │
                │                       └ M         ┘                  │
STATUS (PS)     │      [<userid<]  [AT <host>]  [<display>]            │
                │ { HOST  }   ┌ <host> ┐                               │
                │ { HOSTS }   └ ALL    ┘                               │
                └                                                      ┘

SUSPEND (SLEEP)  <cmid>

             { ALL    }
<display> =  { LAST   }
             { ACTIVE }
```

EXHIBIT WIN120-1

FTS SIMULATION

OBJECTIVE: To use File Transfer Service commands taught in Lessons WIN090 through WIN120.

***NOTE: Your local host for this simulation is MAC. Enter all commands to mail or transfer files on a single line. Perform the tasks listed below in the order given.

1. Log-on to FTS using the following information:

USERID	CDTS
PASSWORD	WIN
IDENT	PWIN
CLASSIFICATION OF YOUR OUTPUT	DO NOT MARK
CLASSIFICATION OF FILES YOU WILL CREATE	UNCLASSIFIED

2. Mail a message to USERID STUDENT located at LANT. The text of this message is stored in a disk file called CDTS/TRNMSG. Review the contents of your workspace, then execute the command and monitor its progress through execution.

3. Mail an inline message to USERID TRAINER located on the FORCE host computer. This message is to be from one to four lines long. Review your workspace, then execute it.

4. Enter the READMAIL subsystem and review all the mail in your mailbox. (Assume that all but the last message have already been read.) After reading the mail, purge the first message and return to FTS.

5. Using SEND, transfer a disk file called CDTS/DISKFILE located on the local host. The file will be transferred to the REDCOM host, and is to have the same name as the origin file. After entering the SEND command line, review the contents of your workspace. Finally, execute the command and watch its progress through execution.

6. Transfer a disk file named WINT/TRNFILE located at NMCC to the TAC host. The target file already exists, and is named STUDENT/DATA. After you enter the command, review your workspace, then execute it.

7. Log off of FTS.

EXHIBIT WIN130-1

NETJOB COMMAND

The NETJOB command allows remote job entry (RJE); it retrieves a job file from a given host, submits it for execution at a second host, directs the output to the specified hosts, and memos the mailboxes of specified individuals.

FORMAT:

```
        NETJOB   { INLINE                       }
                 { <jobfile> [AT <host1> ]      }   [ON < host2>]

  [                { <opfile>       }               ] [         { <userid> [AT <host4>]       } ]
  [                { SYSOUT         }               ] [         { { BRDCST    }               } ]
  [  OUTPUT [IS]   { JOUT           } [AT <host3>]  ] [  MEMO   { { BROADCAST } <file2>       } ]
  [                { FLUSH          }               ] [         { { B         }               } ]
  [                { RLP  <rlpid>   }               ] [                                         ]
```

REQUIRED PARAMETERS:

<jobfile> Catalog/file of a BCD file containing the job to be executed.

INLINE The job to be executed will be entered following this command.

OPTIONAL PARAMETERS:

<hostn> A WIN host

<opfile> Catalog/file of file to which output is to be directed.

SYSOUT Output is directed to SYSOUT of the indicated host.

JOUT Output is to be released (i.e., not printed).

FLUSH Output is to be released (i.e., not printed).

RLP Designates a remote line printer.

<rlpid> Two character identification for remote line printer designation.

<userid> User to which the first $MSG2 card is to be mailed when the indicated job is done executing.

<file2> File containing a list of users to which the first $MSG2 card is mailed when the indicated job is done executing.

EXHIBIT WIN140-1

DEFAULTS:

If not specified, the "AT hostn" and "ON hostn" for each default to the user's host and the OUTPUT defaults to SYSOUT at the user's host.

USAGE CONDITIONS:

If the jobfile does not exist, an error message will be sent to the user. The job must be on the file in BCD.

If INLINE is specified and INLINE data has previously been entered for any command (not just NETJOB), the user will be asked if he wishes to reuse the previous INLINE data for this command.

If a break is entered while entering INLINE data, the user is returned to command level ready to re-execute the current command or to enter a new command. The INLINE data file is destroyed and will not be available for reuse with the next INLINE command.

INLINE data is ended by entering the QUIT command or a null line of input (i.e., a carriage return).

The use of quotes surrounding filenames is optional and may be used to avoid conflict with keywords. If quotes are not used, the filename must be in the standard GCOS format.

The first $MSG2 card in the job file will be sent as a memo if the MEMO option is used.

COMMAND STREAM COMMANDS

CHECK Statement

```
FORMAT:  CHECK <file>
```

END Statement

```
FORMAT:  END
```

IF Statement

```
FORMAT:  IF <label 1> [NOT] ERROR  { SKIP TO <label 2> }
                                   { [SKIP TO] END     }
```

RUN Statement

```
FORMAT:  RUN <file>
```

SKIP (GO) Statement

```
FORMAT:  SKIP [TO]  { <label> }
                    { END     }
```

WAIT Statement

```
FORMAT:  WAIT  { ON <label>               }
               { UNTIL <time of day>      }
               { [FOR] <length of time>   }
```

EXHIBIT WIN150-1

SAMPLE COMMAND STREAM

```
TRANS1 SEND CATA/FILE1 AT USAFE TO CATA/FILE2 AT REDCOM
 IF TRANS1 ERROR SKIP TO FAIL1
MALE1 MAIL INLINE TO USER1 AT REDCOM
INLINE
 THE TRANSFER OF FILE "FILE1" WAS SUCCESSFUL, CHECK YOUR CATALOG/FILE OF
 "CATA/FILE2" FOR YOUR UPDATED COPY.
QUIT
 SKIP TO TRANS2
FAIL1 MAIL * TO USER1 AT REDCOM CC USER2 AT PACAF
INLINE
 THE TRANSFER OF FILE "FILE2" DID NOT TAKE PLACE.  WILL RETRY AT 0700 ON THE
 6TH OF NOVEMBER.
QUIT
TRANS2 REPLACE CATA/FILE3 AT JDA WITH CATA/FILE4 AT NAVEUR
 IF TRANS2 ERROR SKIP TO FAIL2
MALE2 MAIL INLINE TO USER3 AT JDA CC USER4 AT PACAF
INLINE
 THE TRANSFER OF FILE "FILE4" WAS SUCCESSFUL.
QUIT
 SKIP TO PURGE1
FAIL2 MAIL * TO USER3 AT JDA CC USER4 AT PACAF
INLINE
 THE TRANSFER OF FILE "FILE4" DID NOT TAKE PLACE, WILL RETRY.
QUIT
PURGE1 DELETE CATA/FILE5 AT NMCC
 IF PURGE1 ERROR SKIP TO FINI
MALE3 MAIL INLINE TO USER4 AT PACAF
INLINE
 THE FILE "FILE5" HAS BEEN DELETED.
QUIT
FINI END
```

EXHIBIT WIN150-2

UNCLASSIFIED

PERSONNEL STRENGTH PROFILE BY ASSIGNED MISSION
BASED ON AUTHORIZED AND ASSIGNED STRENGTHS
(GIPSY SYSTEM TEST NO. 1)

	AUTHORIZED	ACTUAL	OFFICER	ENLISTED	STR-RATIO	MSN-READY	READINESS	SIGNIFICANCE
AA01	48	45	5	49	0.9375000	3.2000000	3	1
AB02	40	31	1	30	0.7750000	3.8709677	3	4
AC03	65	75	25	50	1.1538462	2.6000000	3	9
AC04	65	29	0	29	0.4461538	6.7241379	3	16
AC05	75	55	13	42	0.7333333	4.0909091	3	25
AC06	113	73	5	68	0.6460177	3.0958904	2	36
AC07	15	23	4	19	1.5333333	0.6521739	1	49
AD09	23	22	21	1	0.9565217	5.2272728	5	64
AD09	35	35	0	35	1.0000000	4.0000000	4	81
AD10	40	52	6	46	1.3000000	2.3076923	3	100
AD11	50	55	1	54	1.1000000	1.8181818	2	121
AD12	105	103	3	100	0.9809524	1.0194175	1	144
AE13	29	13	1	12	0.4482759	6.6923077	3	169
AE14	39	29	13	16	0.7435897	6.7241379	5	196
AE15	6	7	7	0	1.1666667	4.2857143	5	225
AE16	29	24	22	2	0.8275862	6.0416667	5	256
AE17	49	42	10	32	0.8571429	5.8333333	5	289
AF18	150	135	1	134	0.9000000	4.4444444	4	324
AF19	133	100	10	90	0.7518797	6.6500000	5	361
AF20	170	168	4	164	0.9882353	5.0595238	5	400
AF21	50	60	25	35	1.2000000	4.1666667	5	441
AF22	88	95	5	90	1.0795455	4.6315789	5	484
AG23	115	115	12	103	1.0000000	5.0000000	5	529

UNCLASSIFIED

TABULAR REPORT

EXHIBIT GIP010-1

UNCLASSIFIED

PERSONNEL STRENGTH PROFILE BY ASSIGNED MISSION
BASED ON AUTHORIZED AND ASSIGNED STRENGTHS
(GIPSY SYSTEM TEST NO. 1)

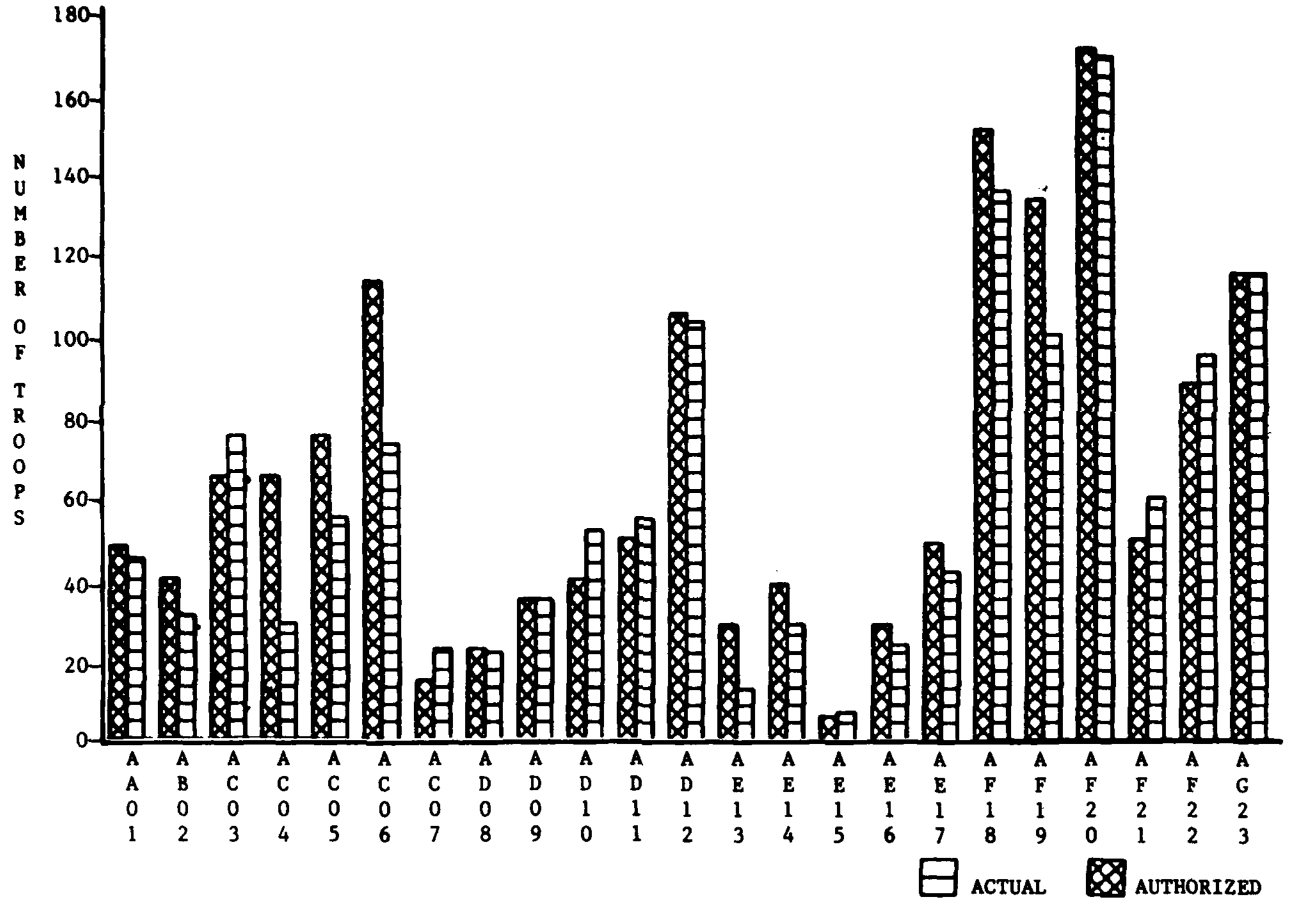
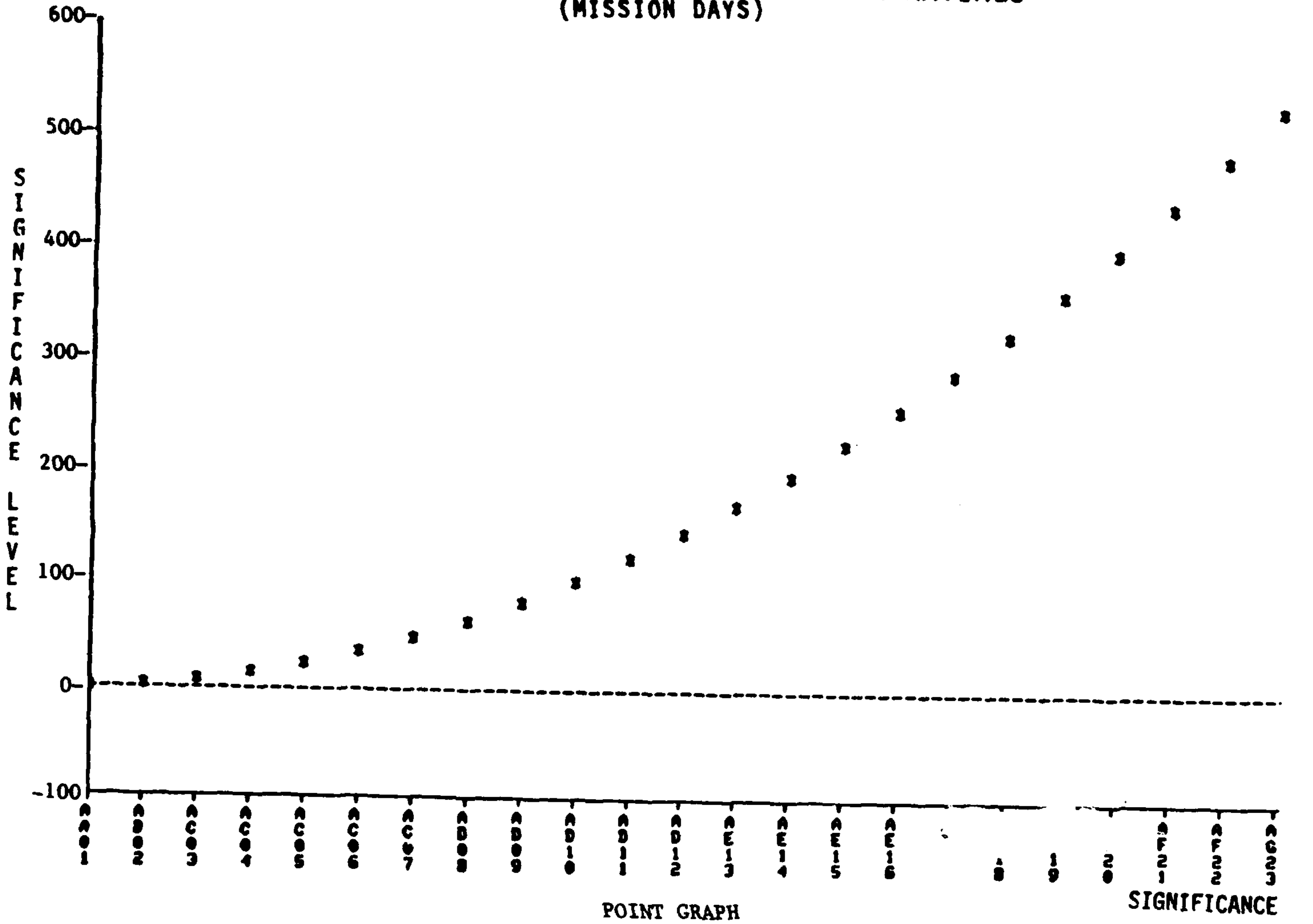


EXHIBIT GIP010-2

UNCLASSIFIED

LIKELIHOOD OF MISSION COMPLETION RATINGS
(MISSION DAYS)

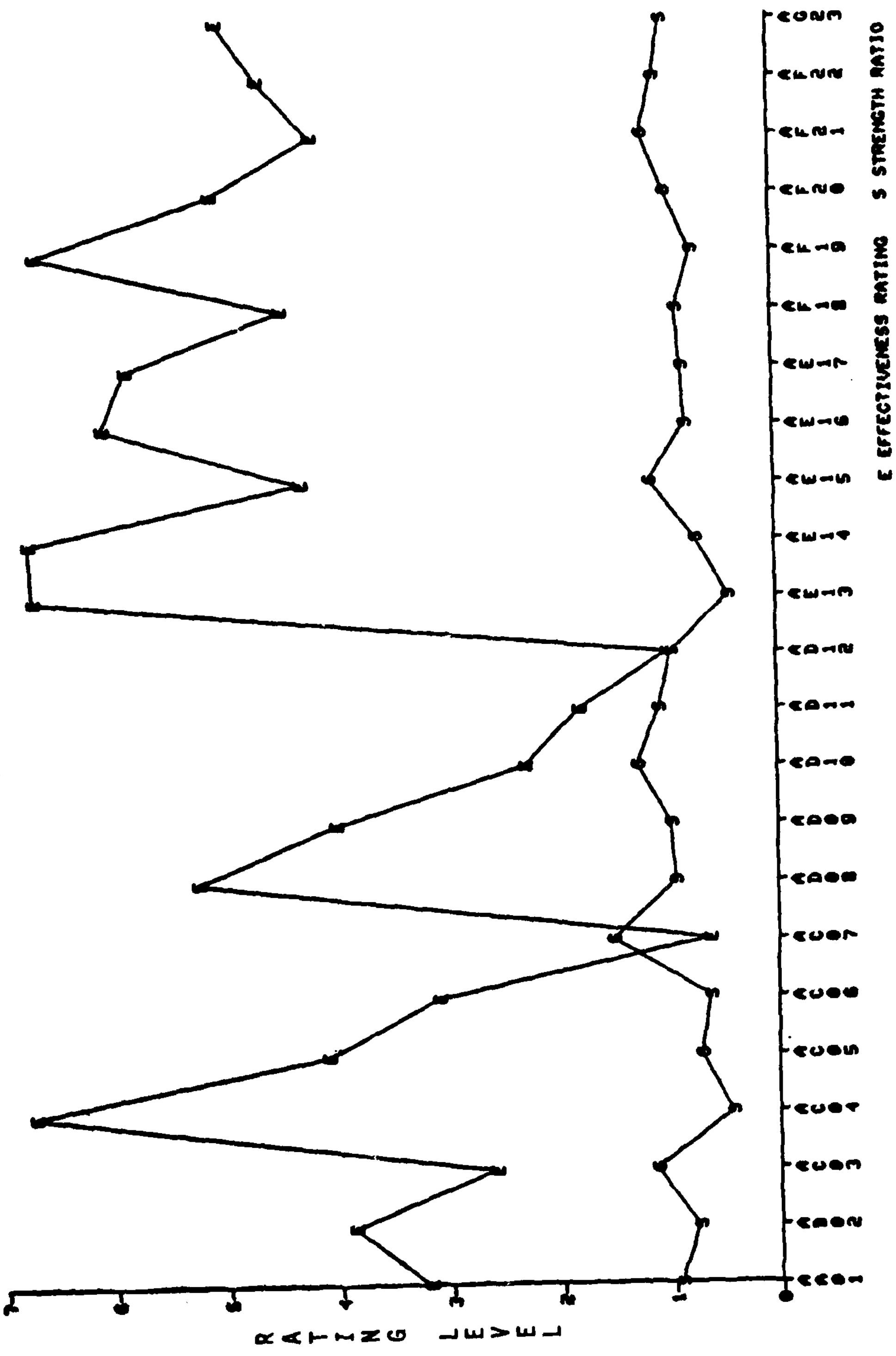


POINT GRAPH

EXHIBIT GIP010-3

UNCLASSIFIED

LIKELIHOOD OF MISSION COMPLETION RATINGS
(MISSION DAYS)



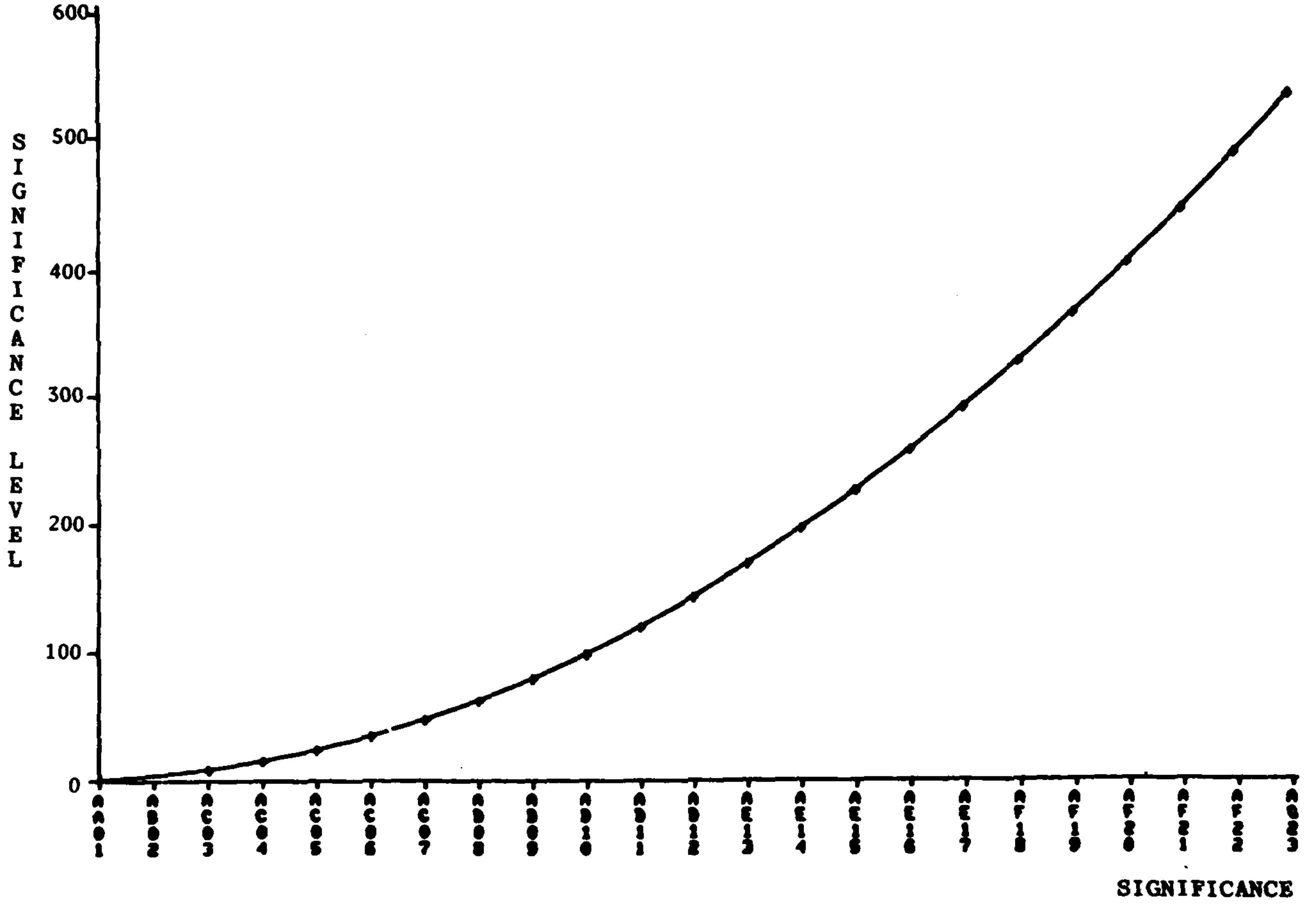
LINEGRAPH WITH MULTIPLE LINES

EXHIBIT GIP010-4

UNCLASSIFIED

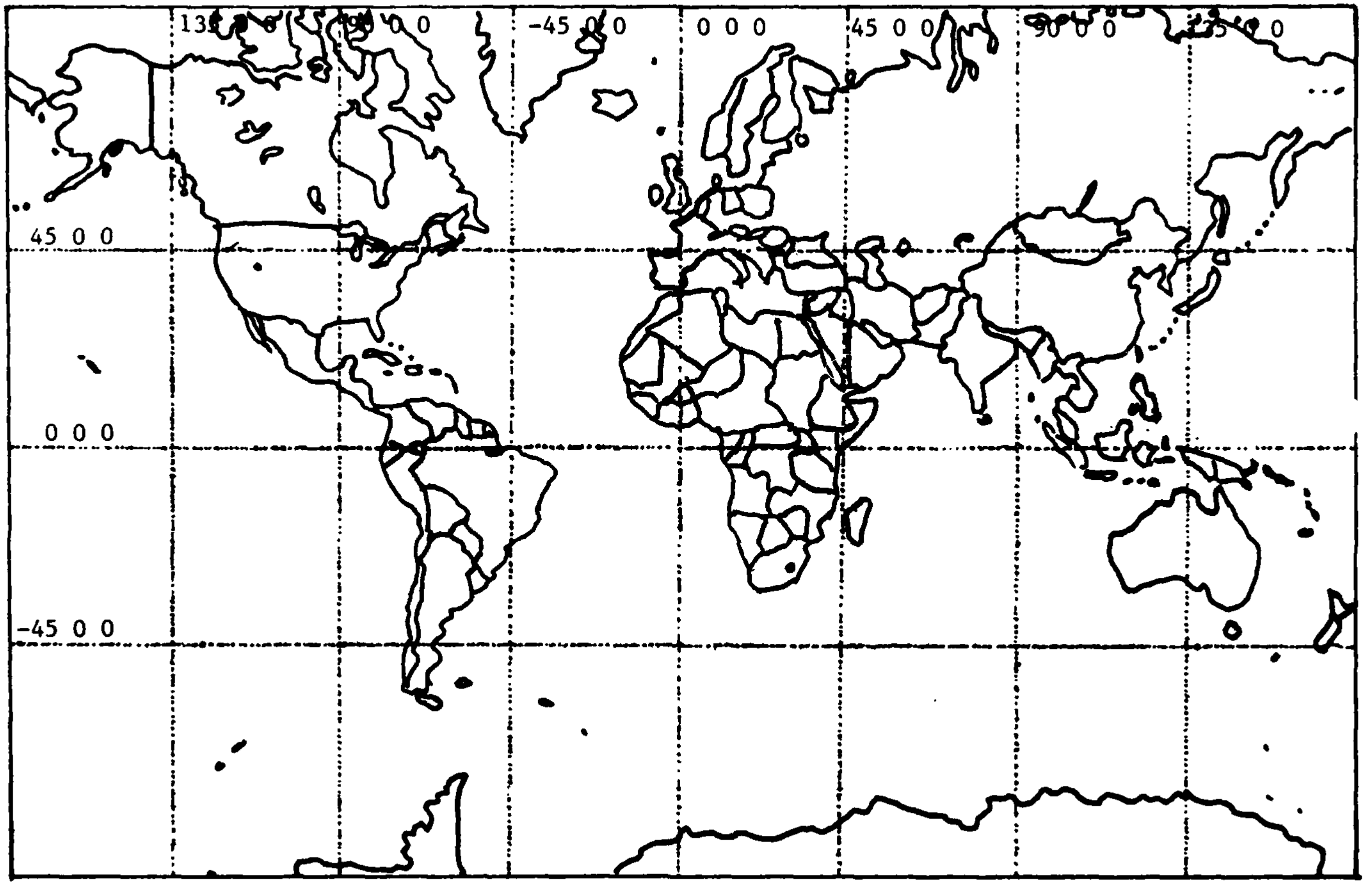
LIKELIHOOD OF MISSION COMPLETION RATINGS

(MISSION DAYS)



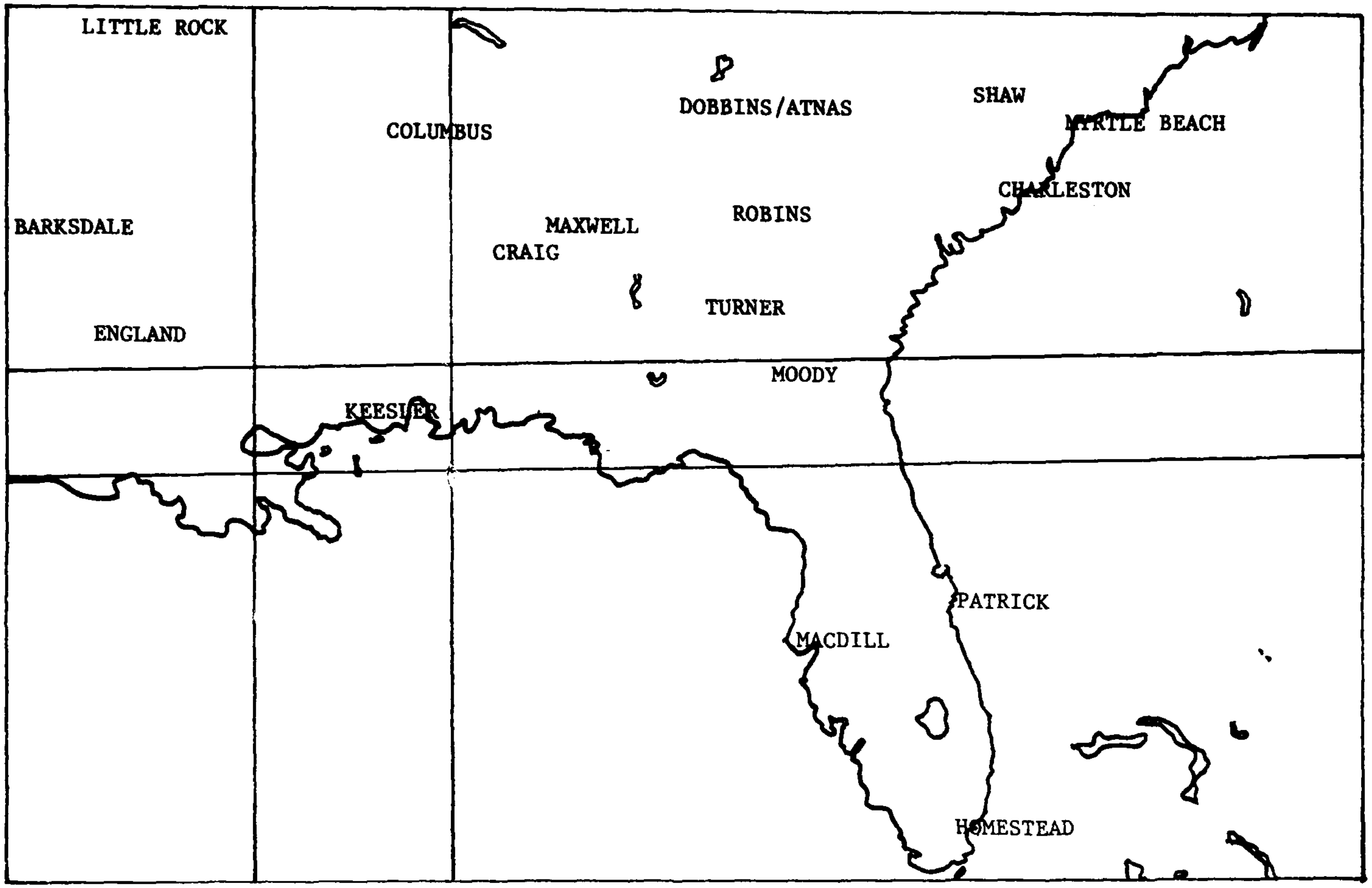
LINEGRAPH WITH SINGLE LINE

EXHIBIT GIP010-5



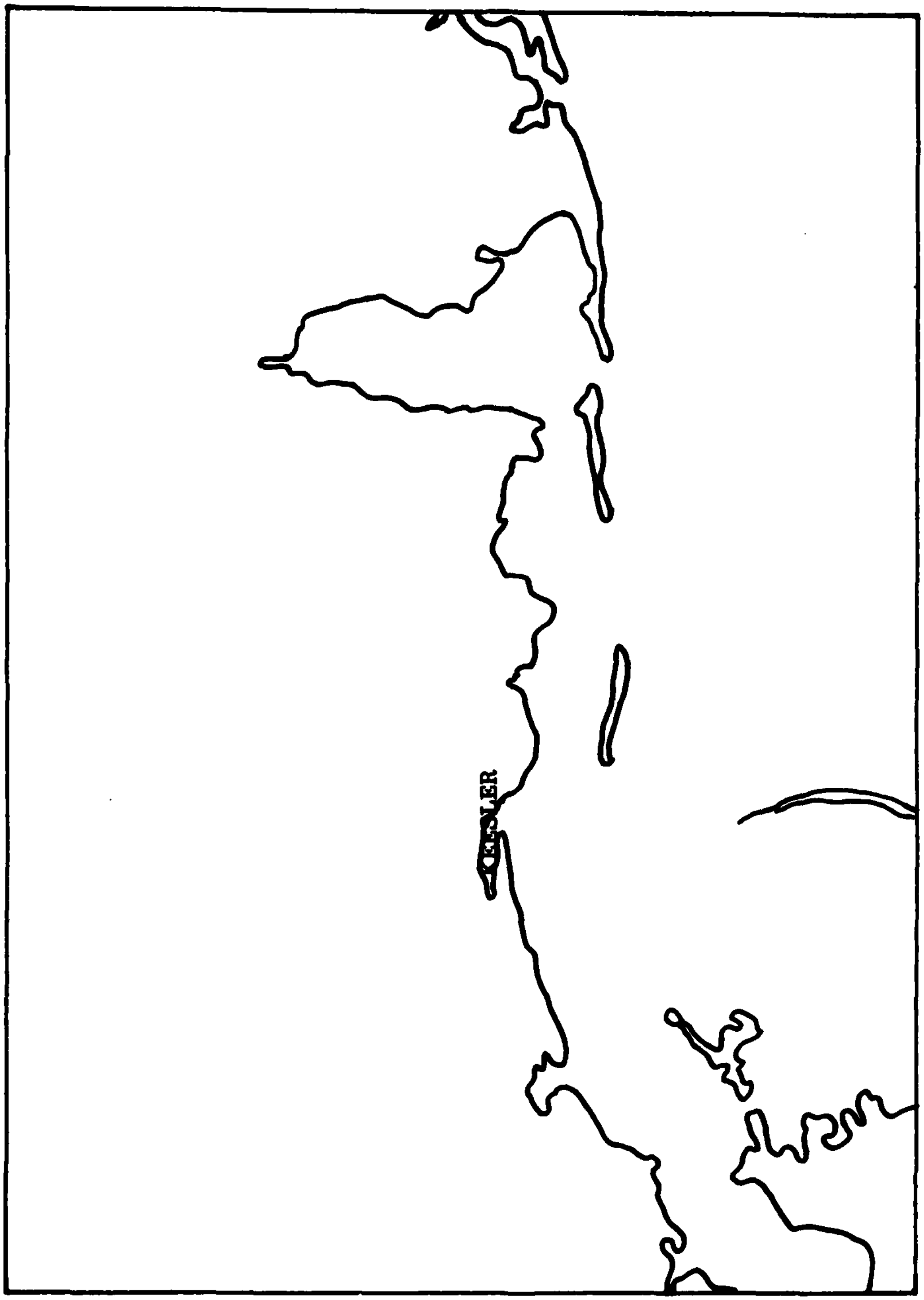
GIPSY WORLD MAP WITH GRID LINES

EXHIBIT GIP010-6



PREPARING THE ZOOM COMMAND

EXHIBIT GIP010-7



RESULT OF ZOOM

EXHIBIT GIP010-8

COURSE OUTLINE

A. Introduction
   1. Prerequisites
   2. Course Outline

B. Required Information for GIPSY
   1. Rules of GIPSY Language
   2. Log-on Procedures
   3. Basic GIPSY Statements
   4. How to Create Files

C. Tabular Report
   1. Commands to Build Report
   2. Display Report

D. Graph Display
   1. Bargraph
   2. Histobar
   3. Histogram
   4. Point Graph
   5. Line Graph

E. Geographical Display
   1. Map
   2. Window
   3. Track
   4. Symbol

EXHIBIT GIP010-9

GIPSY METALANGUAGE

Braces	{ }
Brackets	[ ]
Carets	< >
UPPERCASE LETTERS	
lowercase letters	
Ellipsis	... or ⋮

__Examples:__

```
TITLE [(SIZE <size option>)] "line of title".

TITLE [(SIZE <size option>)] "line of title";
                  .
                  .
                  .
      [(SIZE <size option>)] "line of title".

CLASS [(SIZE <size option>)] ⎧ <classification code>        ⎫
                             ⎩ "<textual classifications>"  ⎭
```

EXHIBIT GIP020-1

INFORMATION FOR LOG ON SIMULATION

SIGN-ON:	$*$LOG22L,TSS
USERID$PASSWORD:	BAT$MAN
IDENT:	ROBIN
FILES:	UNCLASSIFIED
OUTPUT:	CONFIDENTIAL

EXHIBIT GIP020-2

SYNTAX RULES FOR THE CLASS AND TITLE STATEMENTS

```
CLASS [(SIZE <size option>)] {<classification code>   }
                             {<textual classification>}

TITLE [(SIZE <size option<)] "<line of title>".
```

Size Options		Codes			Displayed Classification
JUMBO	J	ZZZ			NONE
LARGE	L	UZZ	UNC	U	UNCLASSIFIED
MEDIUM	M	CSS	CONF	C	CONFIDENTIAL
SMALL	S	SZZ	SEC	S	SECRET
		TZZ	TS	T	TOP SECRET

THIS IS SMALL

THIS IS MEDIUM

THIS IS LARGE

THIS IS JUMBO

EXHIBIT GIP030-1

```
TITLE [(SIZE <size option>)] "<line of title>".
                   OR
TITLE [(SIZE <size option>)] "<line of title>";

              {COPY     }
SET [AUTO]    {WAIT     }  {ON }
              {ECHO     }  {OFF}
              {RECORDING}

SET [AUTO] WAIT TO <number> {SECONDS}
                            {MINUTES}

SET SIZE TO <size option> .
```

EXHIBIT GIP030-2

GIPSY INTERRUPT COMMANDS

//LIST n - will list n preceding lines from last command.

//LIST - causes all statements entered during the current session (i.e., contents of temporary PCS) to be listed at the terminal.

//PREVIEW - allows user to preview unprocessed PCS statement by listing all statements in the current PCS which have not yet been processed.

//PREVIEW n - allows user to preview next n lines of current PCS which have not yet been processed.

//LAST - displays last inpu⁺ ᵌtatement that is incomplete or not yet acted upon.

//COPY ON - turns GIPSY's automatic hard-copy feature on.

//COPY OFF - turns GIPSY's automatic hard-copy feature off.

//CLEAR PAGE - clears the screen to start a new page, if the automatic copy feature is on; a copy will be taken prior to clearing the screen.

//COPY - automatically copies the contents of the current display regardless of the automatic copy parameter.

//ECHO ON -causes all noninterrupt statements from a PCS file to be echoed back to the terminal.

//ECHO OFF - terminates the echo of PCS supplied statements back to the terminal. Only statements containing errors are displayed on the terminal.

//PURGE - purges all remaining statements from the current PCS input file.

//PURGE GROUP - purges the last group of statements supplied by a PCS statement. This command would be used to purge a group of inputs which were found to have too many errors to bother attempting to fix.

//PURGE n - purges the next n lines from the current PCS input file.

//RETURN - causes GIPSY to return to the previous input mode; i.e., if reading from a PCS return to terminal mode, if in terminal mode return to PCS mode, if reading PCS file was nc; completed.

//NOTE text - causes the text following NOTE to be displayed on the terminal as a message to the user of the PCS; text may be up to 126 characters.

//HUH - causes GIPSY to repeat last message, including prompted error messages.

EXHIBIT GIP030-3

//CONTINUE - causes GIPSY to assume that the current process is complete, then GIPSY continues with the next command.

//PAUSE n - causes GIPSY to pause for n seconds before continuing with additional activities.

//FDT - causes GIPSY to display the current File Descriptor Table.

//QDT - lists the current contents of the QDT; i.e., all fields.

EXHIBIT GIP030-3 (Cont)

Here are the four field you are to describe in your FDT.

FIELD NAME	START POSITION	TYPE	LENGTH
DATE	16	A	7
DAY	16	I	2
MONTH	18	A	3
YEAR	21	I	2

I is for Integer and A is for Alphanumeric.

EXHIBIT GIP040-1

The following are the valid relational operators and their meanings:

RELATIONAL OPERATOR	MEANING
EQ	EQUALS
NE	NOT EQUAL
BT	BETWEEN (INCLUSIVE)
LT	LESS THAN
GT	GREATER THAN
LE	LESS THAN OR EQUAL TO (INCLUSIVE)
GE	GREATER THAN OR EQUAL TO
=	EQUALS
>	GREATER THAN
<	LESS THAN
CHANGES (UNARY OPERATION)	PREVIOUS OCCURRENCE OF FIELD IS DIFFERENT FROM PRESENT

EXHIBIT GIP040-2

Now our GIPSY procedure looks like this:

```
CLASS ZZZ.

TITLE "GIPSY TEST" (SIZE JUMBO);

(SIZE MEDIUM) "PENTAGON";

"WASHINGTON D.C.".

SET COPY ON.

SET WAIT ON.

FDT BAT/FDT

FILE BAT/DATA

RETRIEVE IF YEAR EQ 78 AND MONTH EQ "DEC".
```

EXHIBIT GIP040-3

PROJECT FOR LESSON GIP040

1. Create a GIPSY procedure which uses a TITLE, a CLASS,and a SET command and save it to a PCS file.

2. Create an FDT which describes the following fields:

Field Name	Posn	Length	Type
Infotype	11	7	Alphanumeric
Location	26	15	Alphanumeric
Unittype	45	9	Alphanumeric
Authstr	55	4	Integer
Actlstr	59	4	Integer
Enlisted	68	4	Integer
Officer	77	4	Integer

(You may use your own field names.)

Save the FDT to another file.

3. Enter the command DONE to get out of the GIPSY system. Then, reenter the GIPSY system. Access the FDT with an FDT command and list it with the interrupt command //FDT. Then do the same thing with your PCS file using the PCS command.

EXHIBIT GIP040-4

Here is a very simple tabular report.

X	Y1	Y2	Y3
1	3	5	7
2	5	8	11
3	7	11	15

Here is a tabular report created in GIPSY showing the number of units in various readiness states by service.

	ARMY	AIR FORCE	MARINES	NAVY
READY1	30	29	38	28
READY2	50	37	28	35
READY3	70	46	18	72
READY4	50	29	7	29
READY5	30	19	10	15

EXHIBIT GIP050-1

CALCULATE DATA STATEMENT

The format is:

```
CALCULATE DATE = <ARITHMETIC EXPRESSION> [;<ARITHMETIC EXPRESSION>]...
                  [IF <CONDITIONAL EXPRESSION>].


MATH TABLE.

<EQUATION 1> = <ARITHMETIC EXPRESSION 1>.

<EQUATION 2> = <ARITHMETIC EXPRESSION 2>.

             .
             .
             .
<EQUATION N> = <ARITHMETIC EXPRESSION N>.

END.

LOGIC TABLE.

<LOGIC NAME 1>  : <CONDITIONAL EXPRESSION1>.

<LOGIC NAME 2>  : <CONDITIONAL EXPRESSION 2>.
                .
                .
                .
<LOGIC NAME N>  :  <CONDITIONAL EXPRESSION N>.

END.
```

EXHIBIT GIP050-2

PROJECT FOR LESSON GIP050

1. Enter the GIPSY system and retrieve your PCS file that was saved in GIP040. Add to this procedure the command to access the file LIBRARY/GIPSY/..TSTFIL. Also add the command that will allow the procedure to utilize the FDT you saved in GIP040. Finally, add the commands which will build a Tabular Report using the SELECT verb to describe the rows and columns as follows:

   ROWS - Officer, Enlisted
   COLUMNS - Authstr, Actlstr

   Resave the procedure to your PCS file and enter the command DONE.

2. Reenter the GIPSY system and build a second Tabular Report that defines the columns the same way, but that uses the field UNITTYPE for the row and describe it with the USE verb. Save this procedure to a second PCS file.

EXHIBIT GIP050-3

DISPLAY REPORT

```
     FDT LIBRARY/GIPSY/..TSTFDT
     FILE LIBRARY/GIPSY/..TSTFIL
     >TITLE "GENE'S TEST";"25 FEB 81 PM";#14".
     >BUILD TABULAR REPORT.
     >ROWS.
     >USE UNITTYPE.
     >COL.
     >SELECT AUTHSTR,ACTLSTR,OFFICER,ENLISTED,SYMBOLCODE.
     >END.
     >RUN
     DATA SELECTION ENDED:
     RECORDS READ    RECORDS QUALIFIED
          23                23
      DISPLAY REPORT.

                                                              1 OF 1
                                GENE'S TEST
                                25 FEB 81 PM
                                         #14

            AUTHSTR   ACTLSTR        OFFICER   ENLISTED   SYMBOLCODE
AIR FORCE       253       267             31        236          -30
ARMY            333       255             47        208            0
COASTGARD       152       115             53         62            0
MARINES         591       558             45        513            0
NATLGUARD        88        76              6         70            0
NAVY            115       115             12        103            0
```

EXHIBIT GIP060-1

RESULT OF A DISPLAY REPORT WITH THE COMPRESSED OPTION

```
DISPLAY REPORT (COMPRESSED).

                                                          1 OF 1
                                          24 FEB 81 PM

                          AUTHSTR    ACTLSTR    OFFICER    ENLISTED
              AIR FORCE       253        267         31         236
              ARMY            333        255         47         208
              COASTGARD       152        115         53          62
              MARINES         591        558         45         513
              NATLGUARD        88         76          6          70
              NAVY            115        115         12         103
```

EXHIBIT GIP060-2

THE ASSIGN STATEMENT

```
FDT LIBRARY/GIPSY/..TSTFDT
FILE LIBRARY/GIPSY/..TSTFIL
   BUILD TABULAR REPORT
ROW.
SELECT AUTHSTR.
COLS.
USE UNITID.
END.
RUN
  BUILD NEW REPORT.
ASSIGN
ROW
PEOPLE = AUTHSTR - 20.
<END ASSIGN.
<END NEW REPORT.


DISPLAY REPORT.

                                                           1 OF 1
           13TH FLEET   18N3BDECOA   1BN3BDECOC   MAJ1KFLUTE   SPACLFORC1
AUTHSTR           115           48          373          152          591
PEOPLE             95           28          353          132          571
```

EXHIBIT GIP060-3

THE DELETE STATEMENT

```
CLASS UZZ.
TITLE "TEST PCS 24 FEB 81".
BUILD TABULAR REPORT.
ROWS.
USE UNITTYPE.
COLS.
SELECT AUTHSTR,ACTLSTR,OFFICER,ENLISTED.
END.
RUN
DATA SELECTION ENDED:
RECORDS READ    RECORDS QUALIFIED
     23                23
DISPLAY REPORT                   UNCLASSIFIED                           UNCLAS
                              TEST PCS 24 FEB 81              1 OF 1

                          AUTHSTR   ACTLSTR   OFFICER   ENLISTED
              AIR FORCE       253       267        31        236
              ARMY            333       255        47        208
              COASTGARD       152       115        53         62
              MARINES         591       558        45        513
              NATLGUARD        88        76         6         70
              NAVY            115       115        12        103

BUILD NEW REPORT.
DELETE COL AUTHSTR.
END NEW REPORT.
DISPLAY REPORT.
                                                              1 OF 1

                              TEST PCS 24 FEB 81

                               ACTLSTR   OFFICER   ENLISTED
                   AIR FORCE       267        31        236
                   ARMY            255        47        208
                   COASTGARD       115        53         62
                   MARINES         558        45        513
                   NATLGUARD        76         6         70
                   NAVY            115        12        103
```

EXHIBIT GIP060-4

THE SUBSET STATEMENT

```
FDT LIBRARY/GIPSY/..TSTFDT
FILE LIBRARY/GIPSY/..TSTFIL
CLASS UZZ
TITLE "TEST PCS 24 FEB 81".
BUILD TABULAR REPORT.
ROWS.
USE UNITTYPE.
COLS.
SELECT AUTHSTR,ACTLSTR,OFFICER,ENLISTED.
END.
RUN
DISPLAY REPORT.
                                                                   1 OF 1

                                    TEST PCS 24 FEB 81

                         AUTHSTR    ACTLSTR    OFFICER    ENLISTED
             AIR FORCE       253        267         31         236
             ARMY            333        255         47         208
             COASTGARD       152        115         53          62
             MARINES         591        558         45         513
             NATLGUARD        88         76          6          70
             NAVY            115        115         12         103

BUILD NEW REPORT
 SUBSET
COL
INCLUDE OFFICER, ENLISTED, ACTLSTR.
END SUBSET.
END NEW REPORT.
DISPLAY REPORT.

                             TEST PCS 24 FEB 81

                               OFFICER    ENLISTED    ACTLSTR
                 ARMY               47         208        255
                 NAVY               12         103        115
                 MARINES            45         513        558
                 COAST GARD         53          62        115
```

EXHIBIT GIP060-5

PROJECT FOR LESSON GIP060

1. Enter the GIPSY system and retrieve one of the PCS files you saved in GIP050. Then enter a RUN and DISPLAY command and observe the results.

2. Append ROW Totals and COLUMN Sequence to the report and DISPLAY it again. The delete the ROW Totals and DISPLAY the report a third time.

3. Build a new report, making the following changes:

   a. Delete a row or column from your report. (You can list the rows of your current report by using the interrupt command //ROWS and the columns by using the interrupt command //COLS.

   b. Subset your report.

   c. Rename the columns of your report.

   After each modification (C1, C2, and C3) use the REVIEW command to check the report.

4. After the last change has been made, and you have entered the command END NEW REPORT, display your new report with the DISPLAY REPORT command. Finally, exit GIPSY with a DONE command and reenter and repeat each step with the other PCS file from GIP050.

EXHIBIT GIP060-6

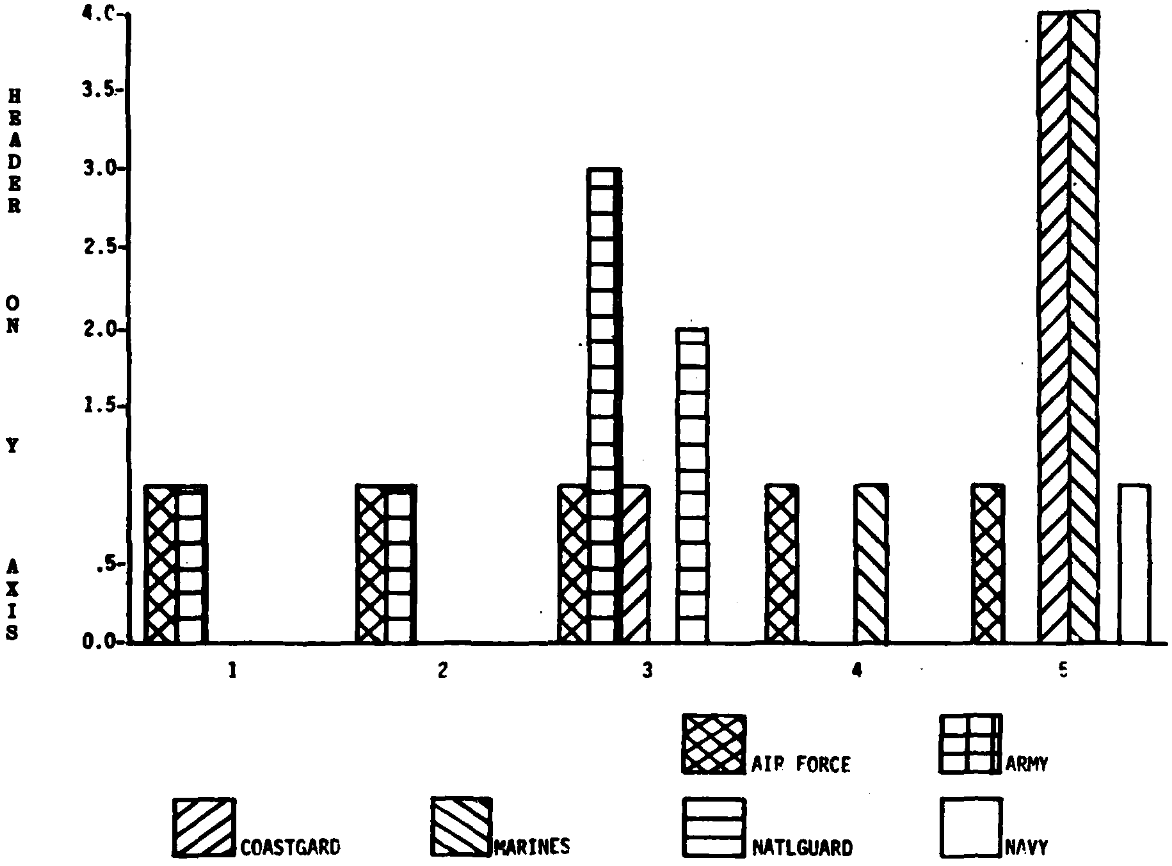


EXHIBIT GIP070-1

DISPLAY HISTOBAR "HEAD" USING "13TH FLEET".

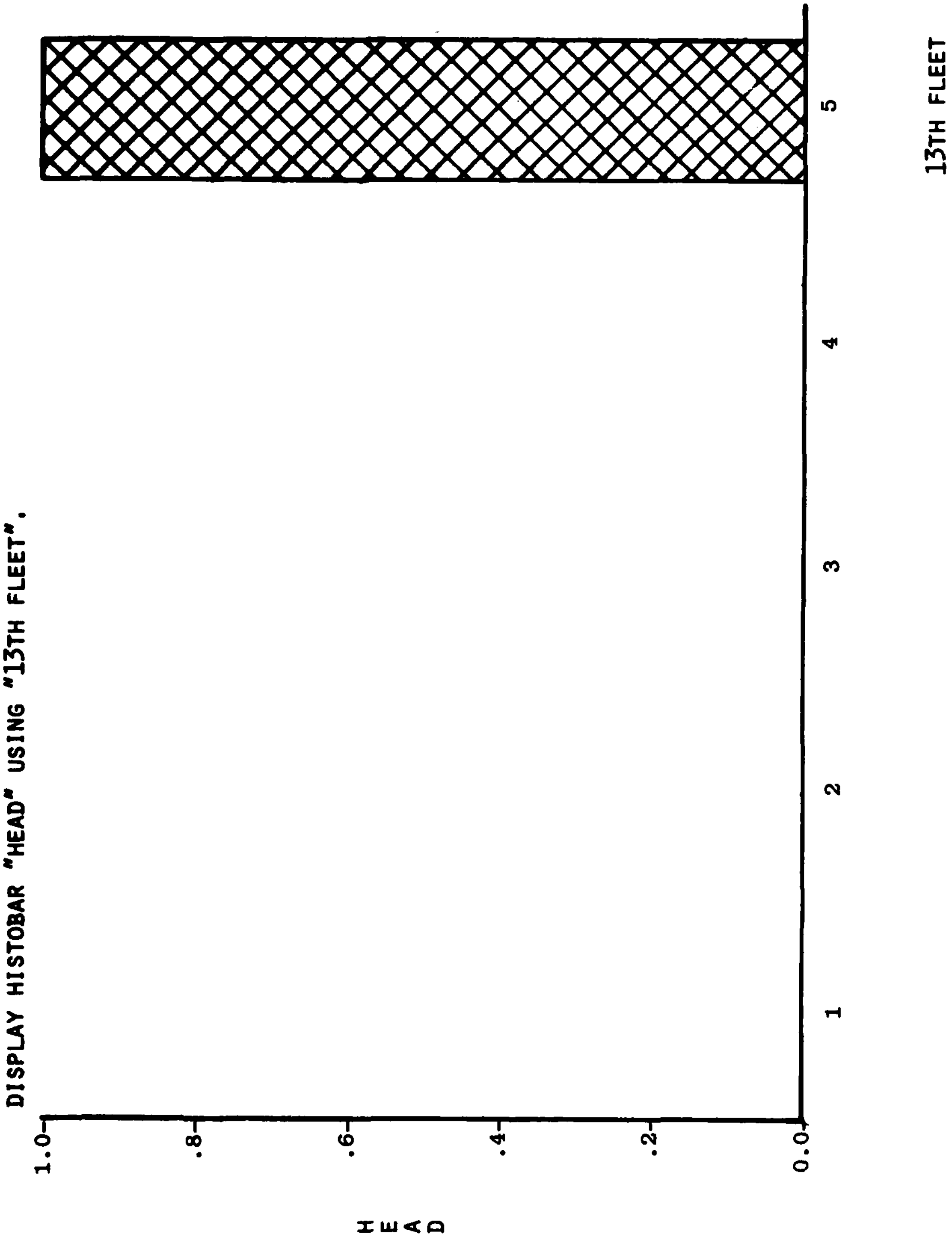


EXHIBIT GIP070-2

UNCLASSIFIED
TEST PCS 24 FEB 81

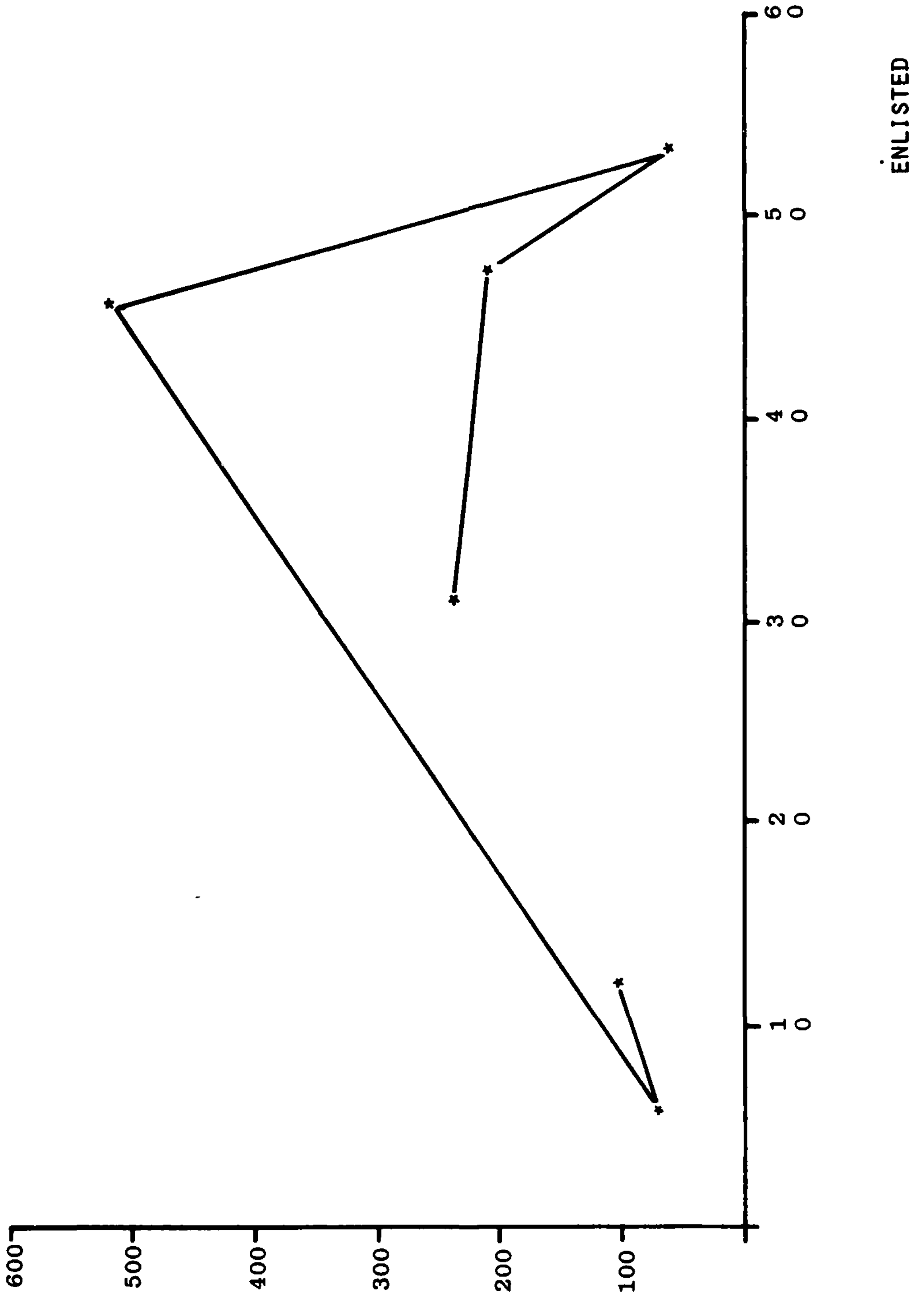


EXHIBIT GIP070-3

DISPLAY BAR STACKING AUTHSTR, ACTLSTR
24 FEB 81 PM

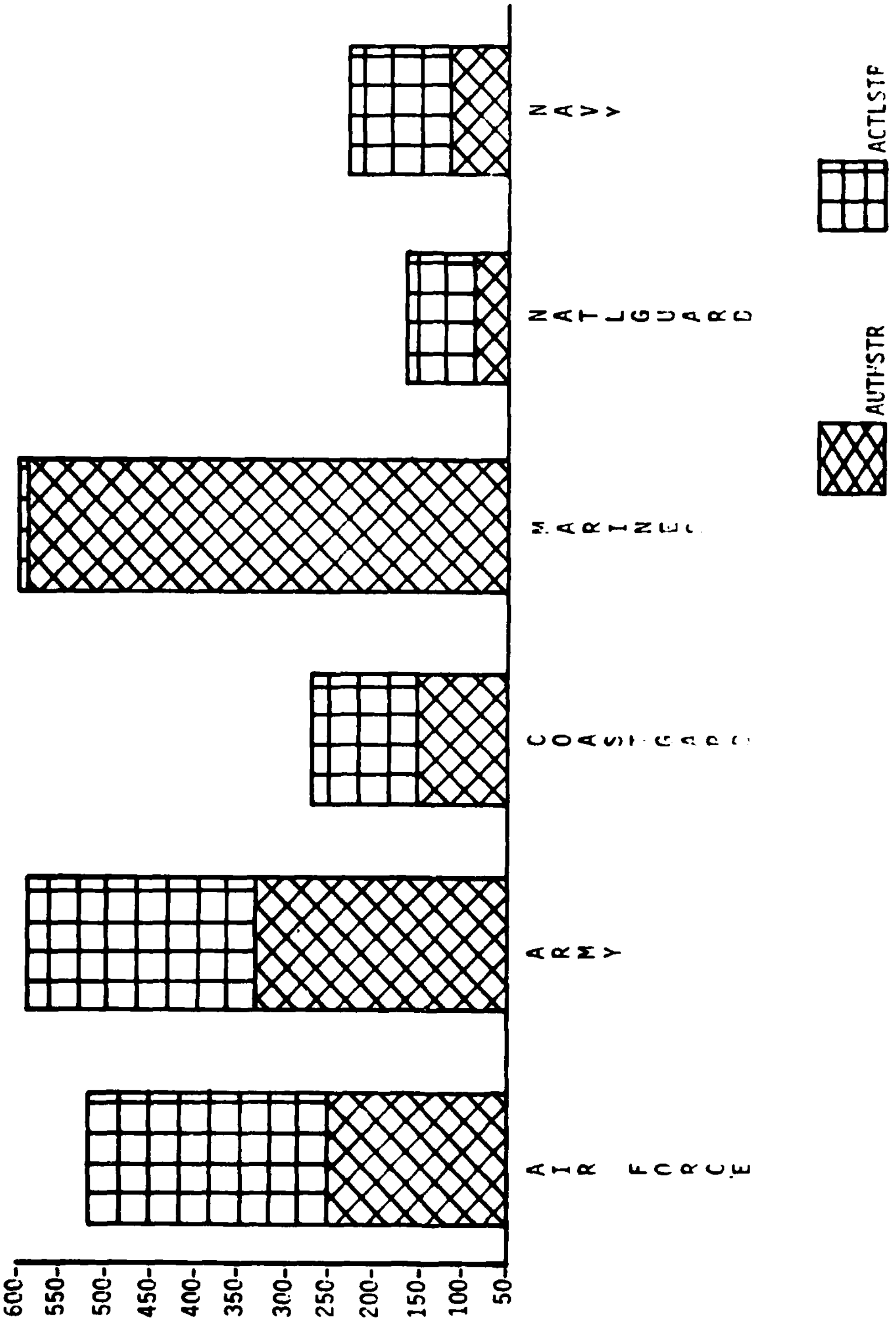


EXHIBIT GIP070-4

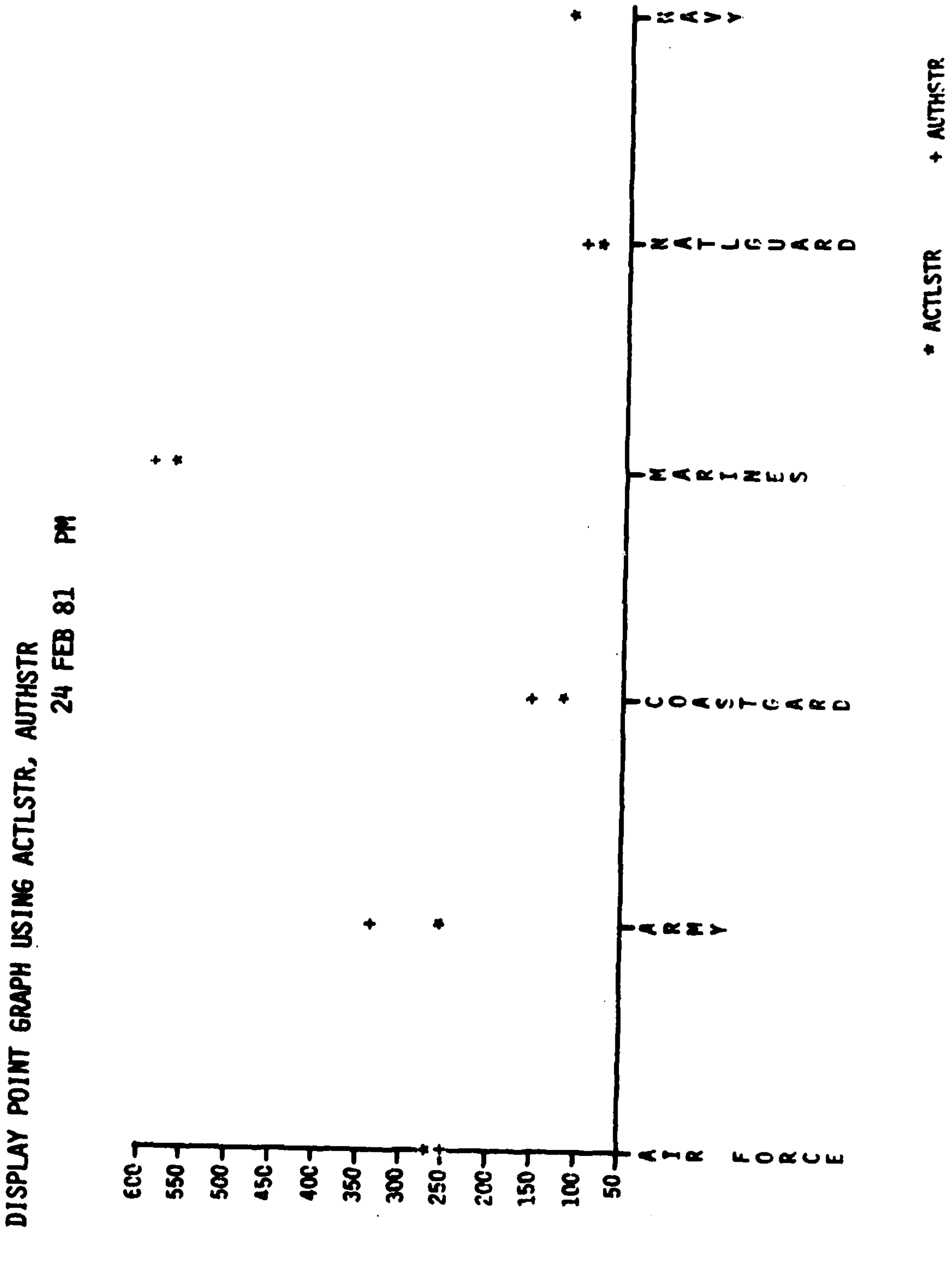


EXHIBIT GIP070-5

```
... Ready for GIPSY command...Enter EXPLAIN command for details          24 FEB 81   PM
 DISPLAY LINE STACKING AUTHSTR, ACTLSTR
```

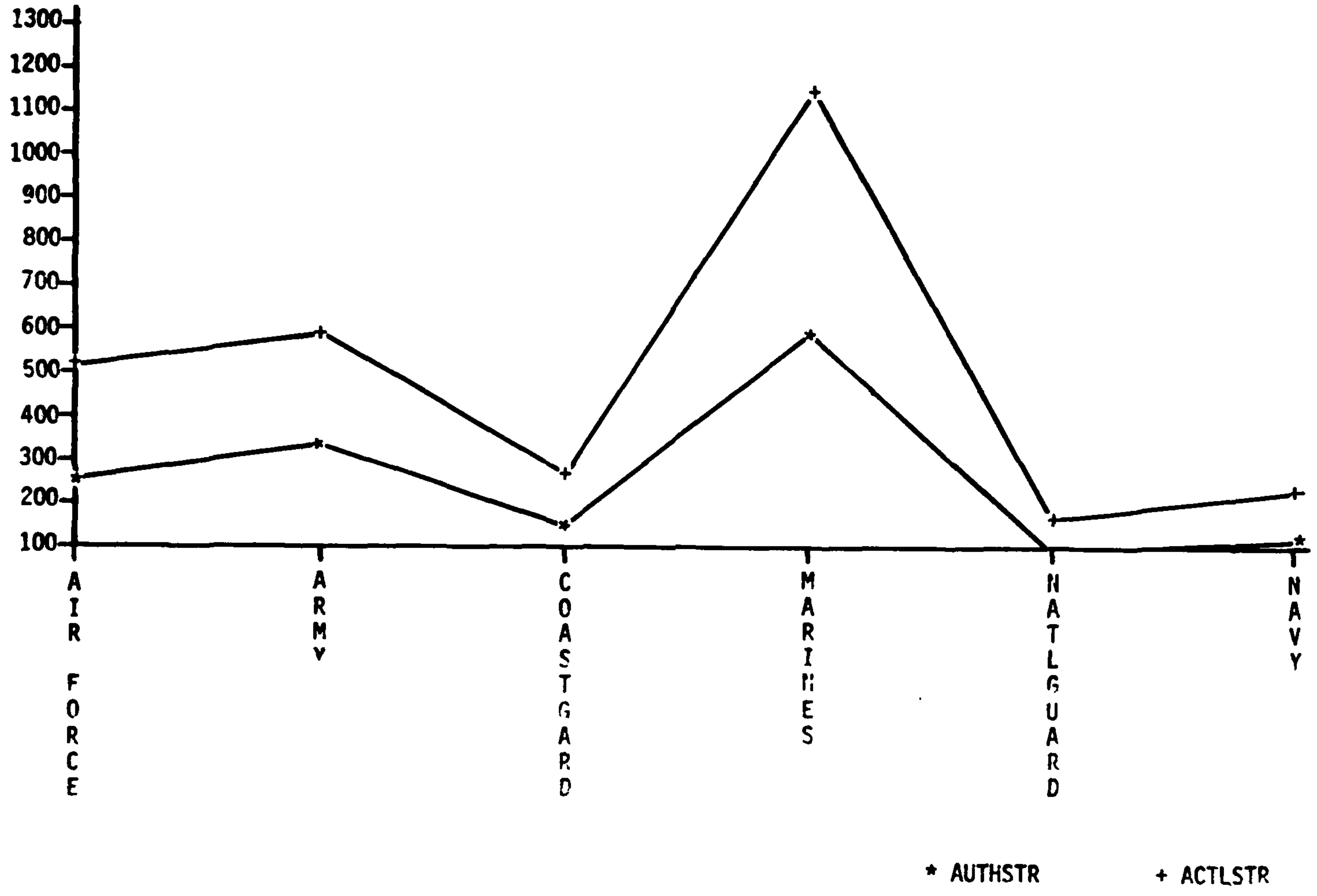


EXHIBIT GIP070-6

PROJECT FOR LESSON GIP070

1. Under GIPSY, enter a SCALE, a LOCK, and an UNLOCK command.

2. If you have access to a graphics terminal, use the Tabular Report from GIP050 to display at least one of each of the following types of graphs:

   a. Bargraph

   b. Histogram

   c. Point Graph

   d. Line Graph

   e. Curve

   If you do not have access to a graphics terminal, you may still enter the appropriate commands. The system can then check your entries for syntax errors without going into the display function.

EXHIBIT GIP070-7

THE MAP COMMAND

```
                  FILE     { WORLD  }
                           { USA    }
                           { WORLD2 }

                  WITH  <map detail options>

                           { <lower left coord>  ,  <upper right coord> }
                           { <area name>                                }
MAP             { WINDOW   { CONTAINING <names of places>               } }
                           { <location name>                            }
                           { CURRENT                                    }
                           { *                                          }

                  COLOR  <color name>

                  PROJECTION   { MERCATOR }
                               { LAMBERT  }
```

Colors

RED
BLUE
GREEN
BLACK
Other colors defined
for your terminal

Line types

```
SOLID    __________
DOTTED   ..........
DASHED   ----------
DOT-DASH ._._._._._.
LDASH    __ __ __ __ __
```

Coordinate Forms

DDHDDDH
DDMMHDDDMMH
DDMMSSHDDDMMSSH

EXHIBIT GIP080-1

USA MAP FILE

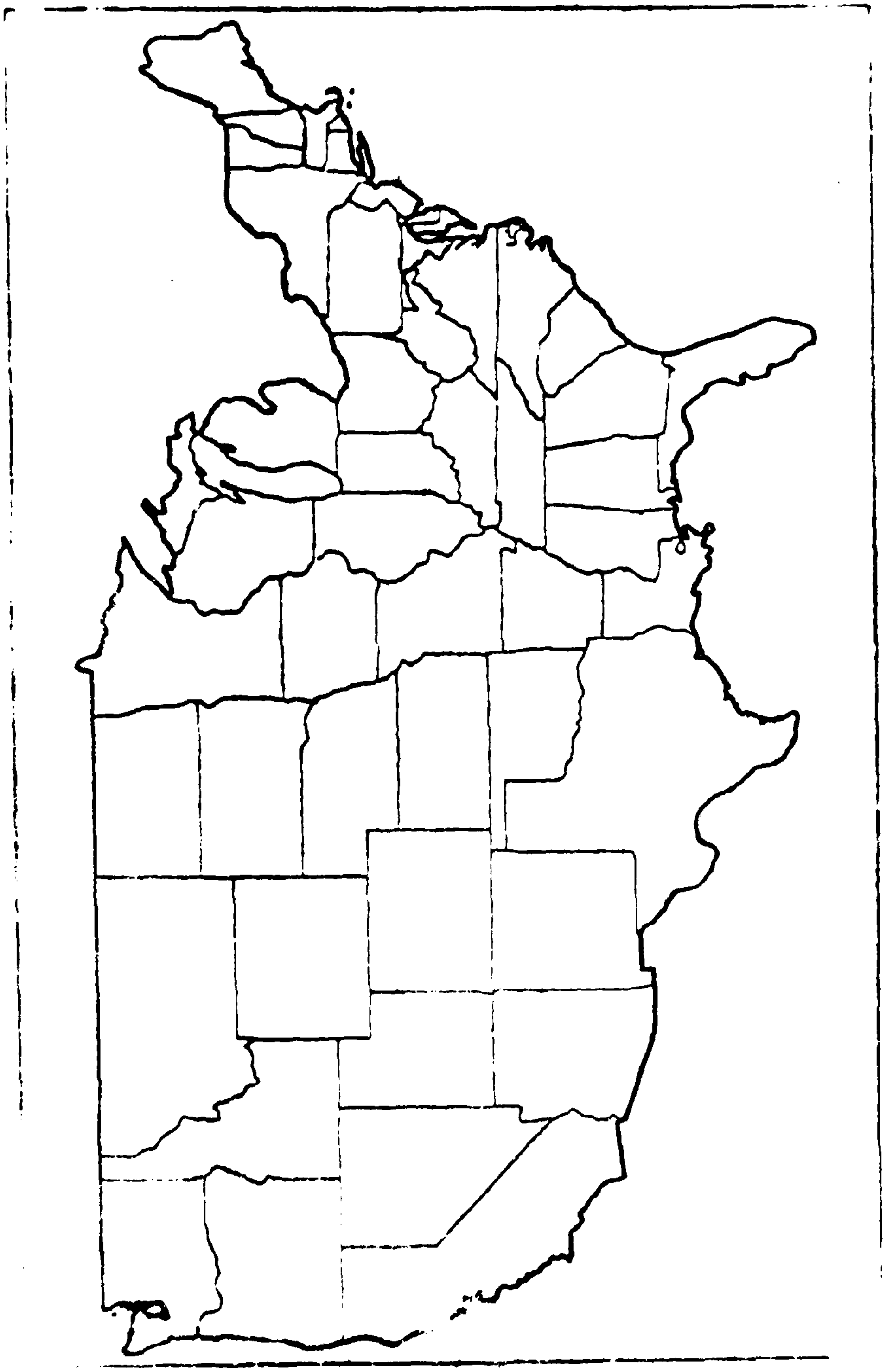


EXHIBIT GIP080-2

WORLD2 POLITICAL AND TOPOGRAPHIC MAP DETAILS

ADMIN (administrative boundaries)

ALL

BOUNDARIES
- COUNTRY BOUNDARIES
- DELIMITED BOUNDARIES
- DISPUTED BOUNDARIES
- INDEFINITE BOUNDARIES
- OTHER BOUNDARIES

CANAL
- IRRIGATION CANALS
- LESSER CANALS
- MAJOR CANALS

COAST

GLACIERS

ICE SHELVES
- MAJOR ICE SHELVES
- MINOR ICE SHELVES

ISLANDS
- MAJOR ISLANDS
- MEDIUM ISLANDS
- MINOR ISLANDS

NORMAL
- COAST
- COUNTRY BOUNDARIES
- MAJOR ISLANDS
- MAJOR LAKES
- MAJOR RIVERS

REEFS

RIVERS
- ADDITIONAL MAJOR RIVERS
- ADDITIONAL RIVERS
- DOUBLE LINED RIVERS
- INTERMITTENT ADDITIONAL RIVERS
- INTERMITTENT MAJOR RIVERS
- INTERMITTENT MINOR RIVERS
- INTERMITTENT RIVERS
- MAJOR RIVERS
- MINOR RIVERS
- PERMANENT RIVERS
- PERMANENT MAJOR RIVERS

SALT PANS
- MAJOR SALT PANS
- MINOR SALT PANS

STATES (U.S. State boundaries)

EXHIBIT GIP080-3

	WINDOW	
WINDOW NAME	LOWER LEFT	UPPER RIGHT
AFARS AND ISSAS	105403N0414949E	124745N0432216E
AFGHANISTAN	202355N0602925E	382935N0745138E
AFRICA	000036S0000026W	000038N0000052E
ALABAMA	301236N0882930W	350131N0845427W
ALASKA	525100N1685556W	712048N1295926W
ALBANIA	393756N0191641E	423942N0210330E
ALGERIA	185913N0083355W	370747N0114245E
ANDORIA	422629N0012706E	424003N0014830E
ANGOLA	180200S0113416E	042450S0241115E
ANTARCTICA	783429S1774029W	542900S0315830W
TOGO	060320N0001144W	110915N0015844E
TOKELAU ISLANDS	102601S1714817W	085810S1574424W
TONGA ISLANDS	211751S1753224W	210605S1751114W
TRINIDAD AND TOBAGO	100244N0615300W	112041N0603045W
TUNISIA	301938N0072133E	372027N0111942E
TURKEY	354849N0260330E	420655N0444924E
UGANDA	012230S0294455E	040924N0350146E
UNITED ARAB EMIRATES	222939N0510058E	260426N0561723E
UNITED KINGDOM, NORTH ICELAND	495912N0075836W	603853N0014849E
UNITED STATES	184610N1782703W	712027N0665529W
WISCONSIN	422909N0925434W	465806N0865509W
WYOMING	405955N1110336W	450012N1040124W
YEMEN ARAB REPUBLIC	124124N0423706E	171811N0461721E
YUGOSLAVIA	405255N0132404E	465404N0225657E
ZAIRE	132925S0120959E	052232N0311554E
ZAMBIA	180112S0220053E	081003S0334128E

EXHIBIT GIP080-4

MERCATOR PROJECTION

MERCATOR PROJECTION

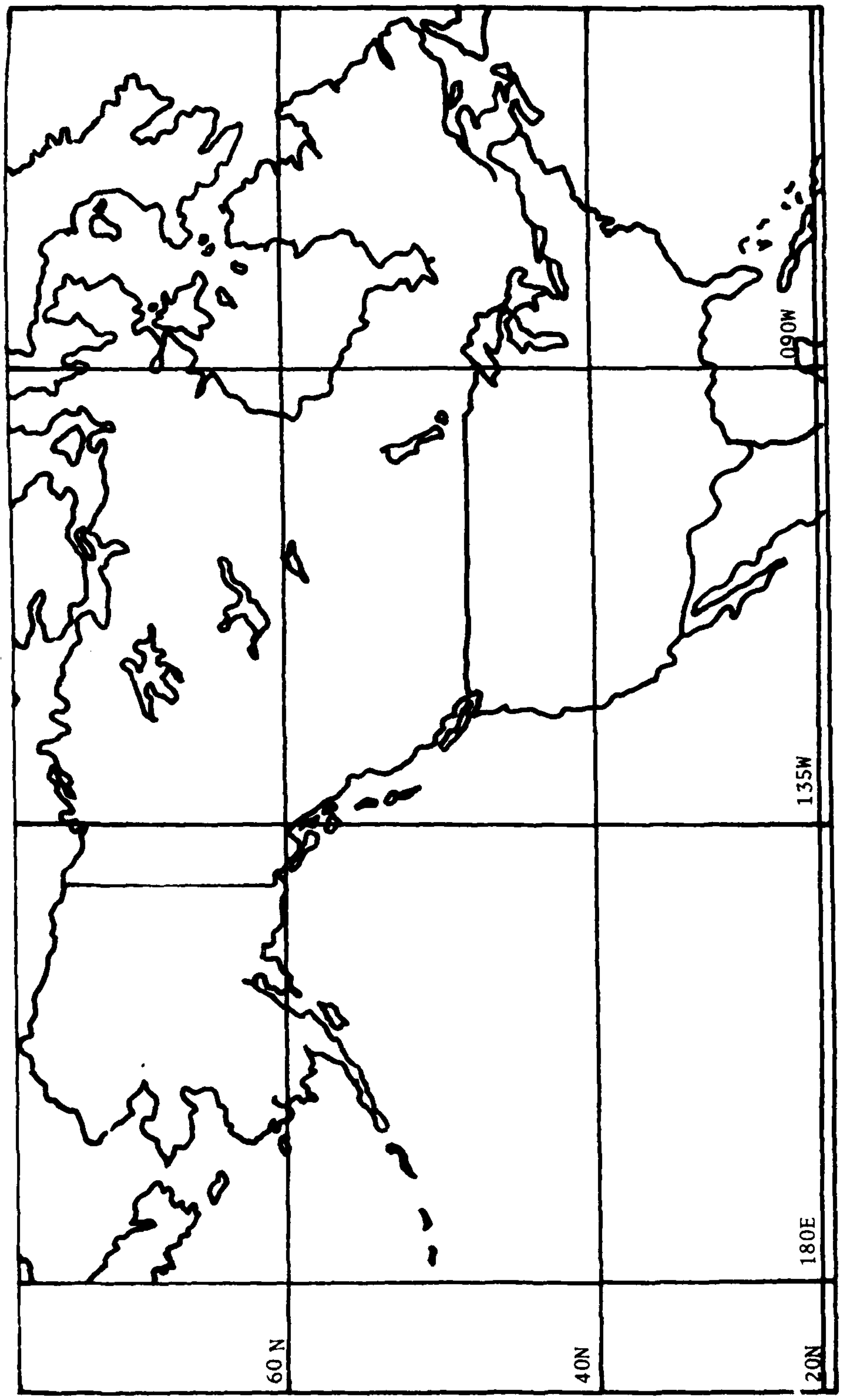
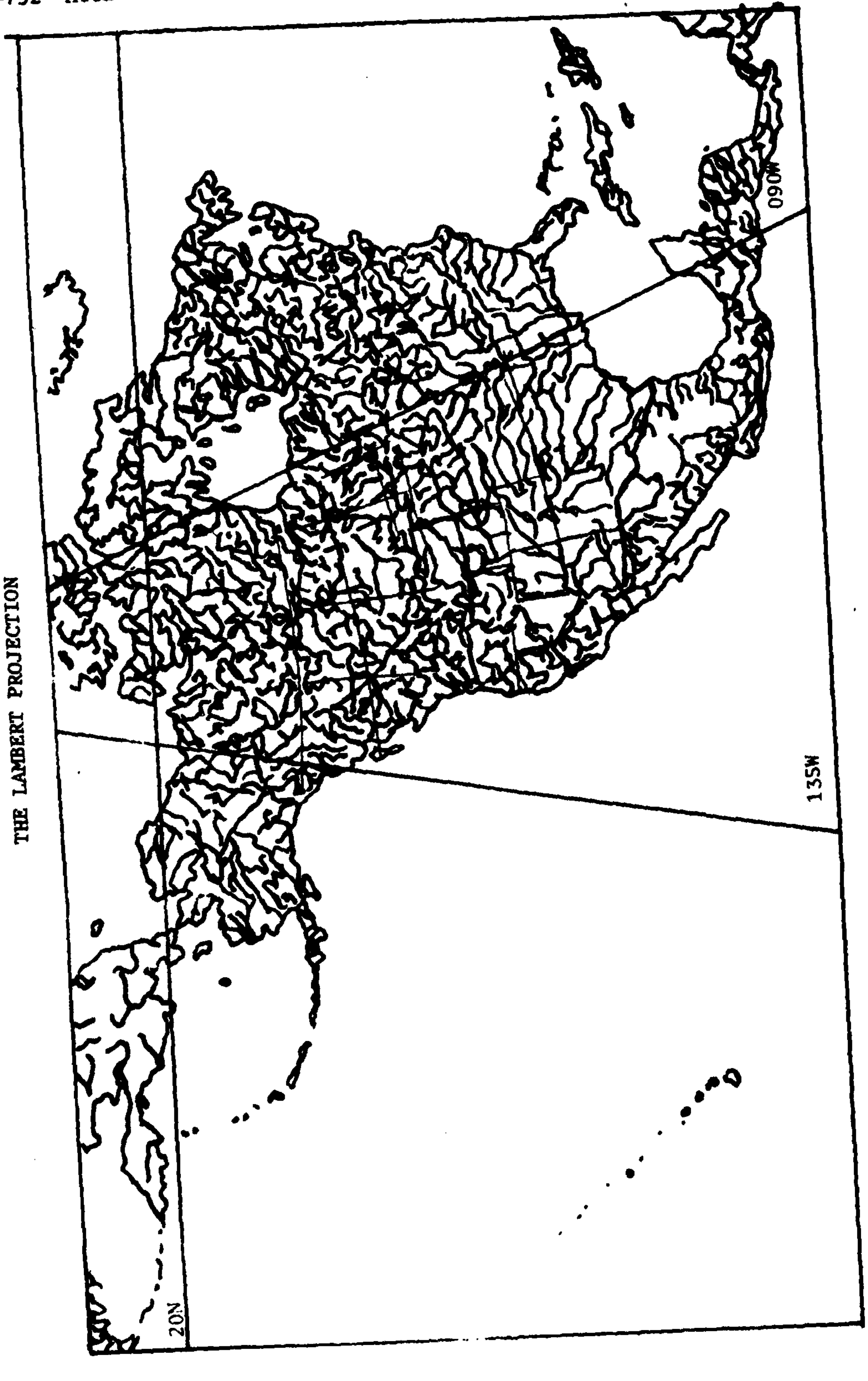


EXHIBIT GIP080-5



THE LAMBERT PROJECTION

EXHIBIT GIP080-6

PROJECT FOR LESSON GIP080

Under the GIPSY System, enter the commands to display a map with the following specifications:

a. MAP FILE: WORLD

b. WINDOW: NORTH AMERICA

c. PROJECTION: MERCATOR

d. Display the map.

Save your commands to a PCS file. If you are not on a graphics terminal, you will have to omit the display command.

EXHIBIT GIP080-7

FORMAT FOR WINDOW AND LOCATION TABLES

```
WINDOW  <area name> = <lower left coord>, <upper right coord>.

WINDOW TABLE.
      <area name>  =  <lower left coord>, <upper right coord>.
                .
                .
                .
END.
```

Example:

```
WINDOW "DEEP SOUTH" = 2957N09004W,3345N08423W.
```

```
LOCATION  <location name> = <geographic coordinate>.

LOCATION TABLE.
   <location name> = <geographic coordinate>.
                .
                .
                .
END.
```

Example:

```
LOCATION MACDILL = 2757N08227W.
```

EXHIBIT GIP090-1

LOCATION NAME	LOCATION
ALBANY	4340N07345W
ALBUQUERQUE	3505N10647W
AMARILLO	3511N10150W
ANCHORAGE	6113N14954W
ATLANTA	3345N08423W
ATLANTIC CITY	3922N07425W
AUSTIN	3929N11704W
BAKER	4447N11750W
BALTIMORE	3918N07638W
BISMARCK	4648N10047W
BOISE	4336N11613W
BOSTON	4221N07105W
BUFFALO	4255N07850W
CALGARY	5101N11401W
CARLSBAD	3226N10415W
NEW HAVEN	4119N07255W
NEW ORLEANS	2957N69004W
NEW YORK CITY	4047N07358W
NOGALES	3121N11056W
NOME	6425N16530W
NORTH PLATTE	4108N10046W
OKLAHOMA	3526N09728W
OTTAWA	4524N07543W
PHILADELPHIA	3957N07510W
PHOENIX	3329N11204W
PIERRE	4422N10021W
PITTSBURGH	4027N07957W
PORT ARTHUR	4830N08917W
PORTLAND ME	4340N07015W
PORTLAND ORE	4531N12241W

Location Names
(Excerpts from Appendix H of the GIPSY User's Manual

EXHIBIT GIP090-2

THE GRID AND SET COMMANDS

```
GRID   [STARTING AT <lower left coord>]
       [INCREMENT BY <grid interval>].

SET  { MAP                          }  { ON  }
     { GRID                         }  { OFF }
     { PROTECTION                   }
     { PROTECTED   display items    }
```

<Display Items>

MAP
LINES
TRACKS
CIRCLES

EXHIBIT GIP090-3

THE ZOOM COMMAND

```
ZOOM [ [ IN  ] AT  <lower left coord>, <upper right coord>                  ]
     [ [ OUT ]     <new center coord> [BY <magnification factor>]           ]
     [                                                                      ]
     [ BY  <magnification factor>                                           ]
```

Magnification Factor	Result of Zoom
1	The same scale will be used, but with a new center.
>0 and <1	A greater area will be covered on the map with a smaller scale.
>1	A smaller area will be covered on the map, but with a larger scale.

EXHIBIT GIP090-4

PROJECT FOR LESSON GIP090

1. Retrieve the PCS file of GYPSY commands that you saved in lesson GIP090.

2. Then make the following modifications:

a. Temporarily remove the GRID lines from your map.

b. Display the map again.

c. Include a comment that explains that the next command will cause a zoom to be performed on the area whose lower left coordinate is defined by New Orleans and whose upper right coordinate is defined by Atlanta.

d. Enter the ZOOM command that would cause the zoom described in step c to be performed.

EXHIBIT GIP090-5

DATA FILE AND SAMPLE FILE DESCRIPTOR TABLE (FDT)

FILE LIBRARY/GIPSY/..TSTFIL

```
1BN3DBEDOACURRENT7701053*392000N0764100WAA01NATLGUARDB  48   23MI     55CLT45  25
1BN3BDECOCCURRENT7701053*420135N0874028WAB02NATLGUARDB  40   31MI     30CLT45  40
1BN3BDECOCHISTORY7701013+385730N0770000WAC03ARMY        65   25MI6    16MI    128
1BN3BDECOCHISTORY7701023+402930N0800000WAC04ARMY        65   29M16    40MI     64
1BN3BDECOCHISTORY7701033+390800N0844000WAC05ARMY        65   65M16    42MI     32
1BN3BDECOCHISTORY7701042+384500N0911830WAC06ARMY        65   73M16    68MI     64
1BN3BDECOCCURRENT7701051+415500N0874328WAC07ARMY        65   82M16    70MI    128
W34SQUAD03HISTORY77010150341000N1193000WAD08AIR FORCE   23   40F-111   4B-52    3
W34SQUAD03HISTORY77010240404500N1115500WAD09AIR FORCE   35   37F-111   5B-52   13
W34SQUAD03HISTORY77010330324800N0965000WAD10AIR FORCE   40   52F-111   6B-52   17
W34SQUAD03HISTORY77010420350958N0900400WAD11AIR FORCE   50   65F-111   7B-52   15
W34SQUAD03CURRENT77010510415500N0874328WAD12AIR FORCE  105  103F-111  13B-52   17
MAJIKFLUTEHISTORY7701013X422558N0825330WAE13COASTGARD   29   13PT-35  32CLT45  18
MAJIKFLUTEHISTORY7701025X432100N0823300WAE14COASTGARD   29   29PT-35  16CLT45  17
MAJIKFLUTEHISTORY7701035X450000N0815500WAE15COASTGARD   29   31PT-35   8CLTAS  28
MAJIKFLUTEHISTORY7701045X463000N0852930WAE16COASTGARD   29   24PT-35  32CLTAS  45
MAJIKFLUTECURRENT7701055X415700N0873000WAE17COASTGARD   29   42PT-35  32CLTAS  28
SPACLFORC1HISTORY7701014=474000N1222500WAF18MARINES    150  135CH53E   8HOWTZ   0
SPACLFORC1HISTORY7701025=450000N0931800WAF19MARINES    150  150CH53E  16HOWTZ   2
SPACLFORC1HISTORY7701035=415500N0874328WAF20MARINES    150  168CH53E  64HOWTZ   4
SPACLFORC1HISTORY7701045=415500N0874328WAF21MARINES    150  142AH53E  32HOWTZ   8
SFACLFORC1CURRENT7701055=415500N0874528WAF22MARINES    150  121CH53E   8HOWTZ  16
13TH FLEETCURRENT7701055:343000N0743300WAG23NAVY       175  201DESTR  14CARRI  12
```

SAMPLE FDT:

Field Name	Start Position	Length	Type
NAME	1	10	A
INFOTYPE	11	7	A
LOCATION	26	15	C
SYMBL	41	4	A
UNITTYPE	45	9	A

EXHIBIT GIP100-1

GIPSY UNIT TRACKING AND IDENTIFICATION DISPLAY

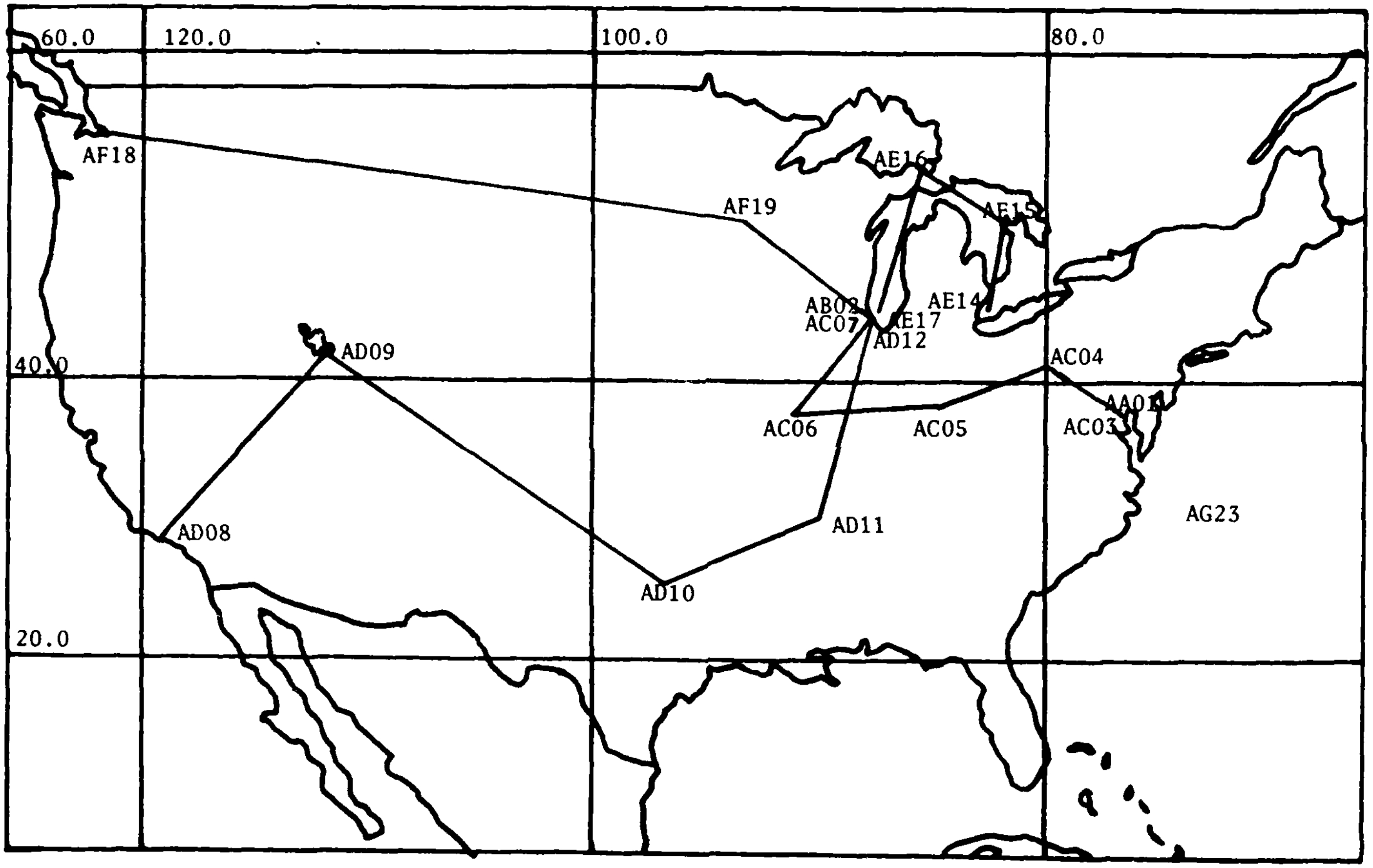


EXHIBIT GIP100-2

THE SYMBOL TABLE

```
SYMBOL TABLE [CONTINUED][(<Symbol Options>)] .

    ⎧<Symbol>                          ⎫ ⎡<Symbol Options>      ⎤ 2
    ⎩\CIRCLE RADIUS <radios> <units>   ⎭ ⎣IF <cond expression>  ⎦  0 .

          .
          .
          .
    END.

SYMBOL OPTIONS:
     (SIZE <Size Options>)
     (COLOR <Color Name>)
     (PROTECT)
     (GROUP SEPARATELY)
     (NOT GROUPED)
     (CENTER)
     (MARKER)
     (NAME <Symbol Entry Name>)
     (LINE TYPE <Line Type>)
     (COORDINATE = <Fieldname>)
```

EXHIBIT GIP100-3

PROTECTED SYMBOI PLOT

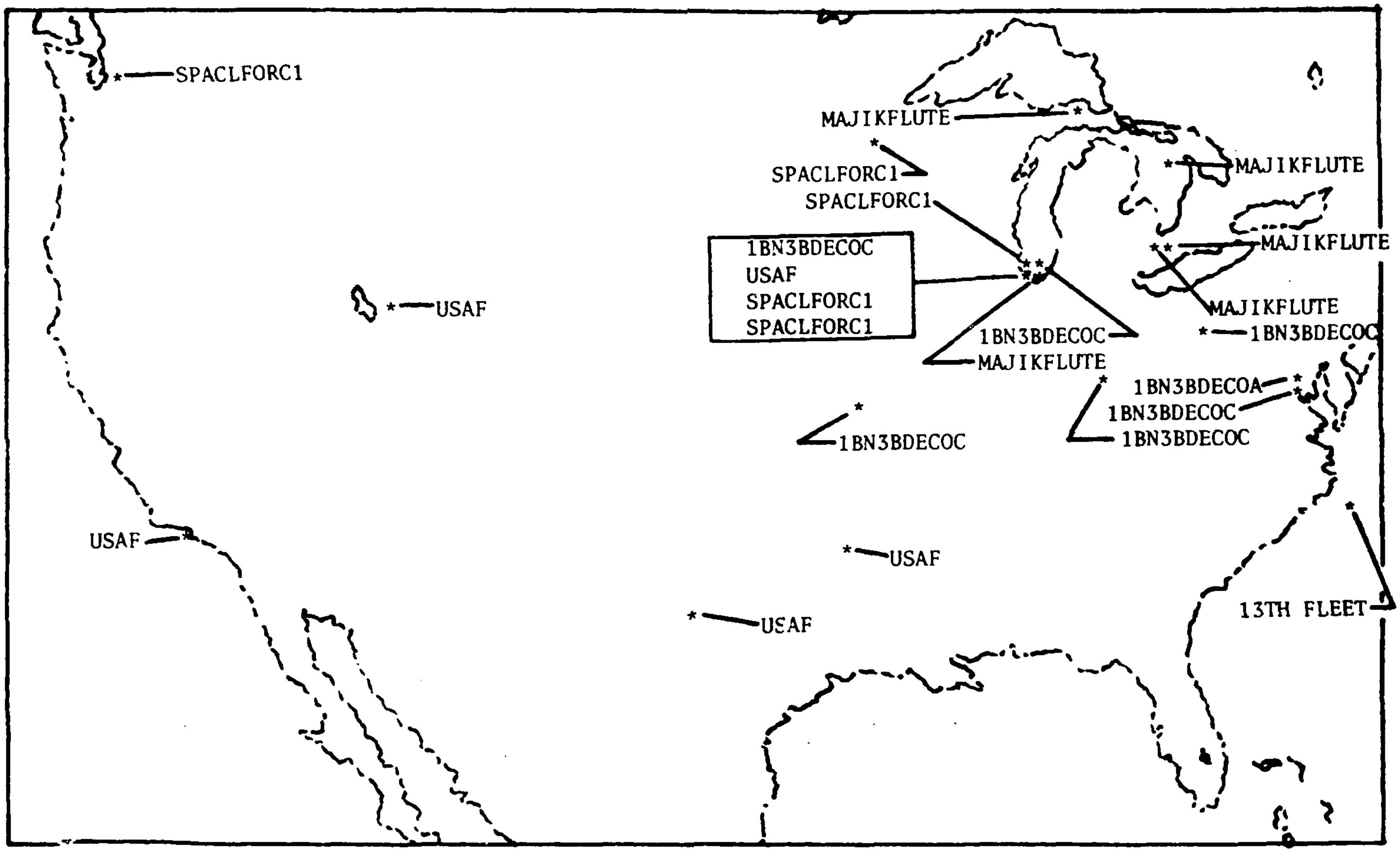


EXHIBIT GIP100-4

SPECIAL GIPSY GRAPHIC CHARACTERS

IDENTIFIER	CHARACTER IDENTIFIED
"\A"	+
"\B"	◇
"\C"	△
"\D"	✡
"\E"	○
"\F"	✱
"\G"	▯
"\H"	•

EXHIBIT GIP100-5

GEOGRAPHIC CIRCLES

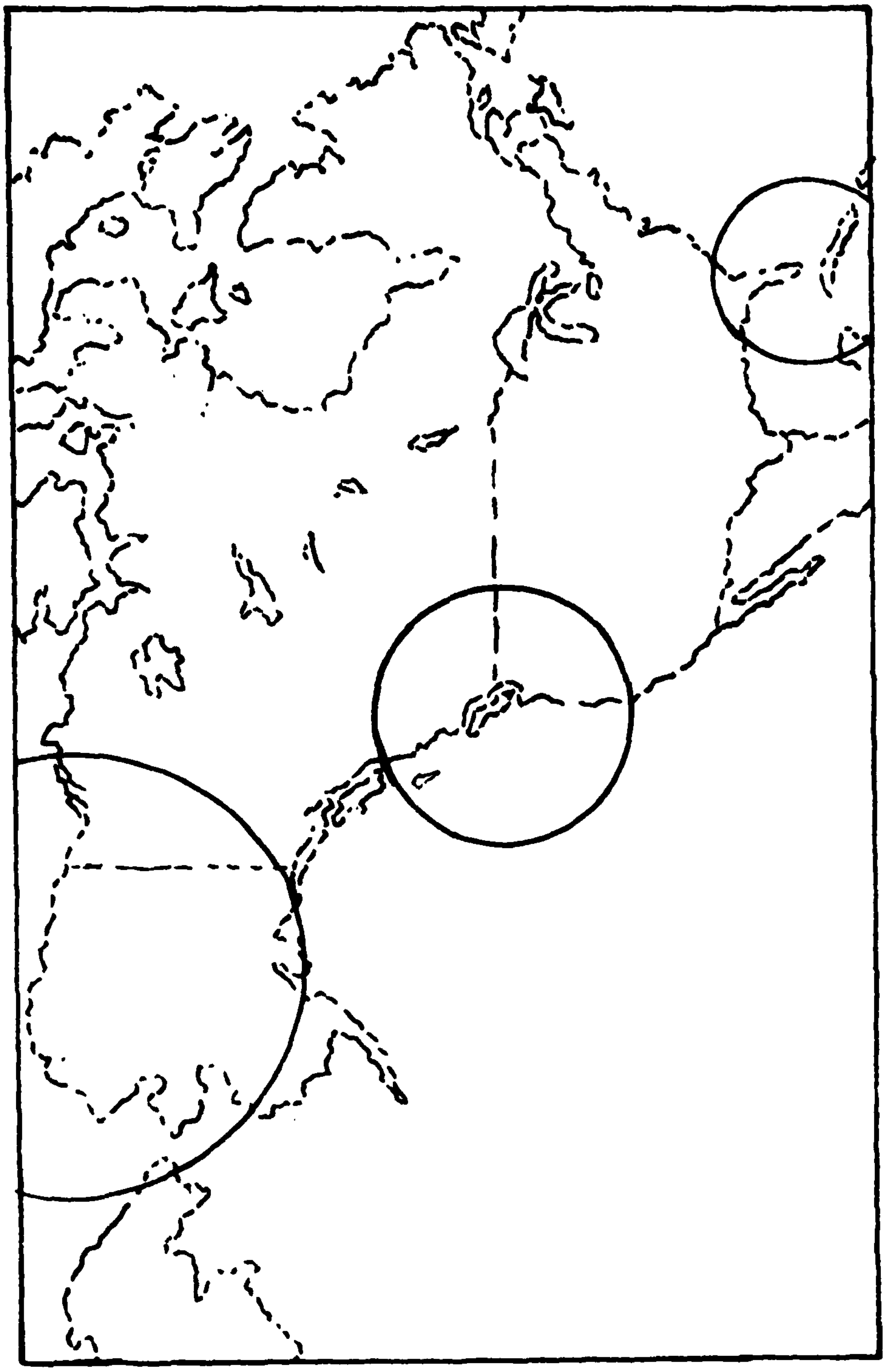


EXHIBIT GIP100-6

RADIUS UNITS FOR GEOGRAPHIC CIRCLES

UNIT	MEANING
NM	- NAUTICAL MILES
MI	- STATUTE MILES
KM	- KILOMETERS
M	- METERS

EXHIBIT GIP100-7

GIP100 PROJECT

Build and display a symbol plot using the following specifications:

FILE: LIBRARY/GIPSY/..TSTFIL
FDT: LIBRARY/GIPSY/..TSTFDT
MAP FILE: WORLD
MAP WINDOW: "UNITED STATES"

1. Include a title on your map.

2. Display the symbol "HISTORY" if the INFOTYPE is equal to HISTORY; otherwise, display a star.

3. Display your map and make a hard copy of it to show your training monitor.

EXHIBIT GIP100-8

THE TRACK TABLE

```
TRACK TABLE [CONTINUED][(<track options>)].

                      ┌                       ┐ 2
<Track Identifier>    │ <track options>)      │
                      │ IF <cond expression>  │
                      └                       ┘ 0

        .
        .
        .
END.
```

TRACK OPTIONS:

```
(SIZE <size option>)                  (DIRECTED ON  POINT
                                                    TRACK
(COLOR <color name>)                  (GREAT CIRCLE)
(PROTECT)                             (DISTANCE)
(GROUP SEPARATELY                     (CUMULATIVE DISTANCE)
(NOT GROUPED)                         (TOTAL DISTANCE)
(NAME <track name>)                   (AZIMUTH)
(CENTER)                              (LOCATION)
(MARKER)
(COORDINATE = <fieldname>)
(LINE TYPE <line type>)
```

EXHIBIT GIP110-1

USING THE AZIMUTH OPTION

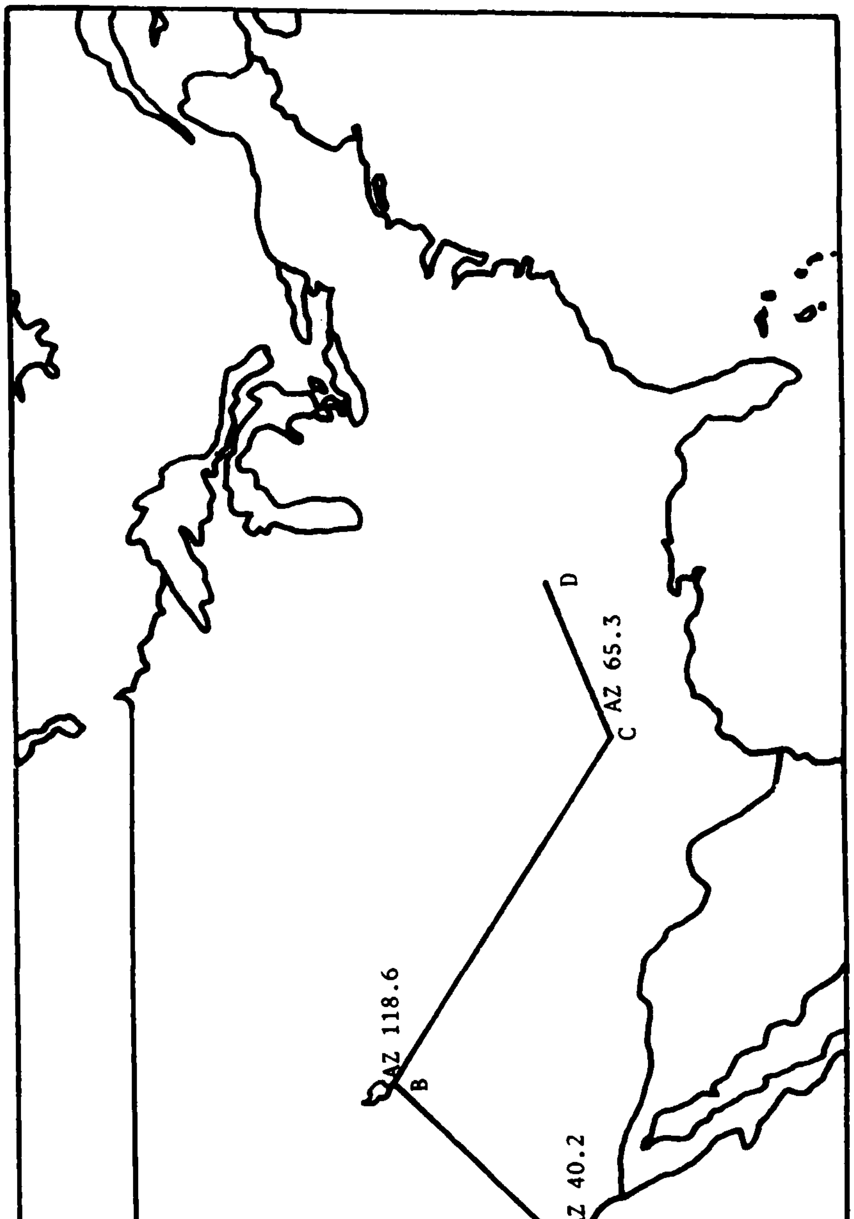


EXHIBIT GIP110-2

USING THE LOCATION OPTION

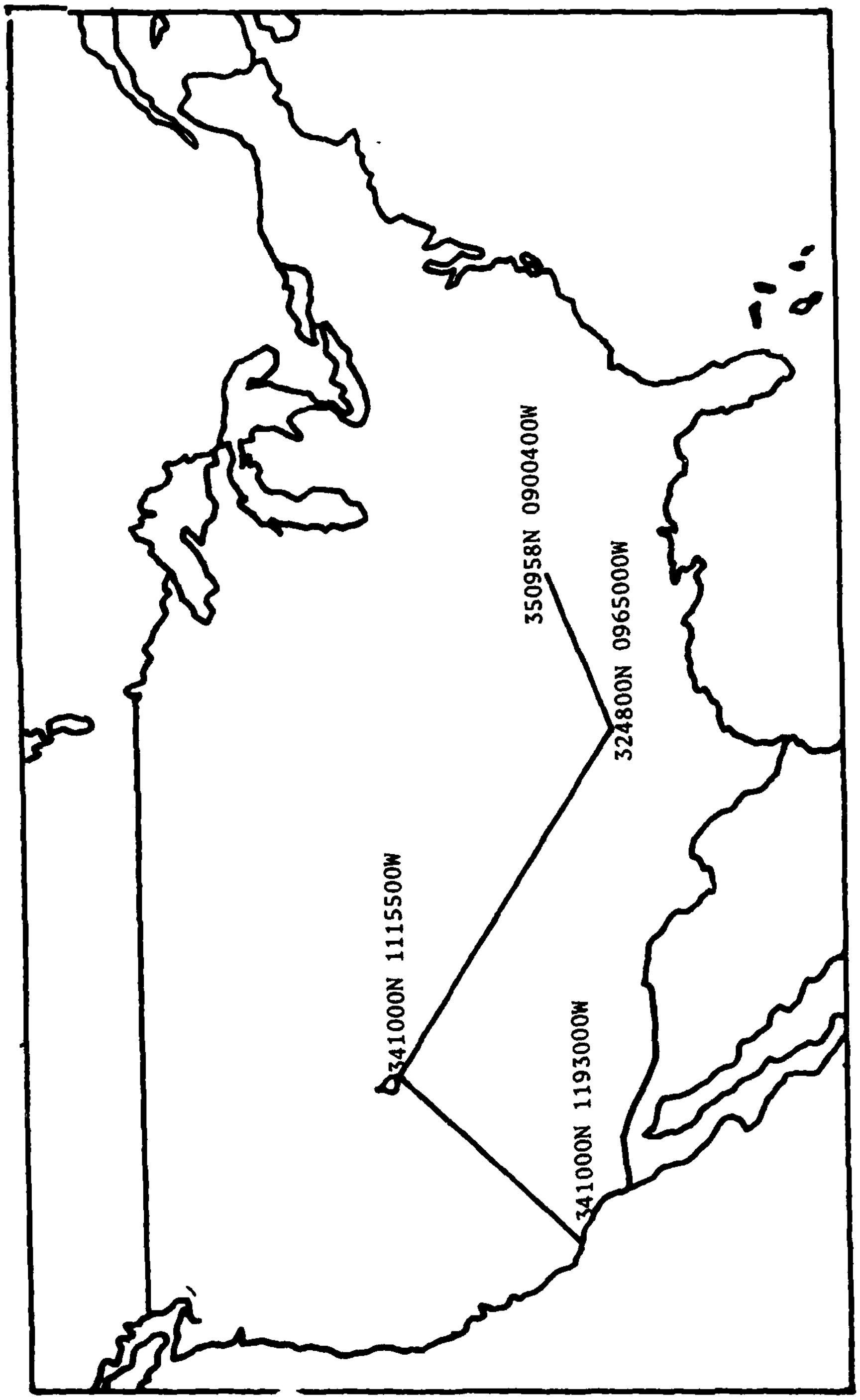


EXHIBIT GIP110-3

GIPSY SYMBOL PLOT

LIST COMMAND SYNTAX: LIST ([OVERLAYED] [SIZE <size option>]).

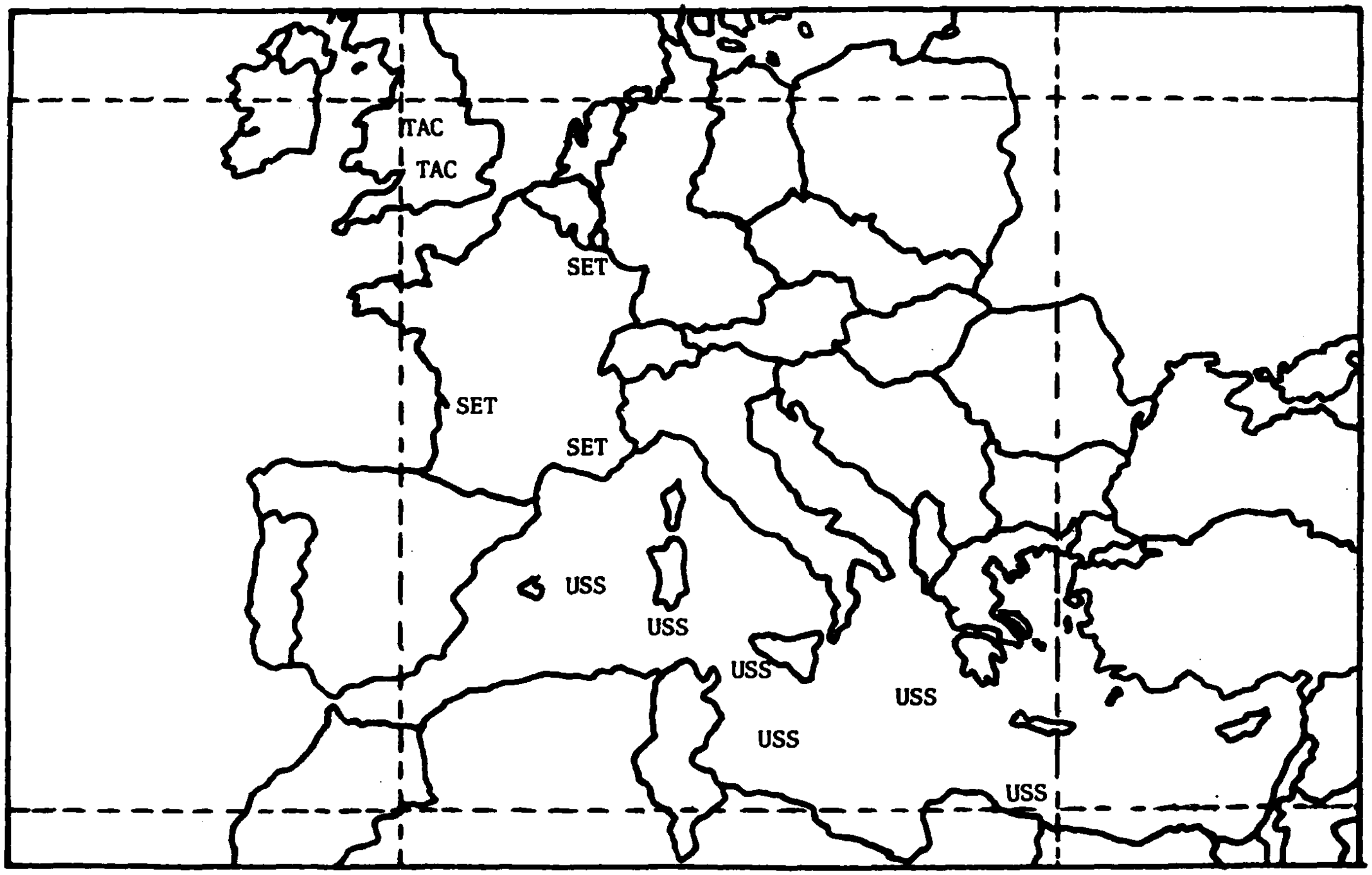


EXHIBIT GIP110-4

GEOGRAPHIC LIST

GIPSY SYMBOL PLOT

```
TAC510005N00200002+012479ARMYHQTACEUR    03502545USCINCEUR
TAC530230N0013000W+012479ARMYHQTACEUR    03500545USCINCEUR
SET450300N0000000W=010479NAVYEUR RECON   01400620CINCUSNAVEUR
SET490010N0050000E=010479NAVYEUR RECON   01400620CINCUSNAVEUR
SET440000N0044500E=010479NAVYEUR RECON   01400620CINCUSNAVEUR
USS400000N0050000E#013179NAVYENTERPRISE  04504510COMSIXTHFLT
USS383000N0083000E#013179NAVYENTERPRISE  04054510COMSIXTHFLT
USS373000N0120000E#013179NAVYENTERPRISE  04504510CINCUSNAVEUR
USS360000N0193000E#013179NAVYENTERPRISE  04504480CINCUSNAVEUR
USS323612N0240001E#020179NAVYENTERPRISE  04504480COMSIXTHFLT
```

EXHIBIT GIP110-5

GIP110 Project

Retrieve the PCS file you saved in Lesson GIP100, then:

1. Build a track table which defines a track for which the field UNITTYPE contains "AIR FORCE". Use at least one track option in your track definition.

2. Change the title of your map.

3. Display the map.

4. Resave your PCS file for the GIP120 project.

5. List the data used for an area of your map.

EXHIBIT GIP110-6

COMMANDS FOR MODIFYING GEOGRAPHIC DISPLAYS

```
                              [    {<location>}                      ] 3
                              [ AT {+         }                      ]
{DISPLAY }                    [ <symbol options>                     ]
{GENERATE} SYMBOL "Symbol"    [ NAMED <list of symbol entry names>   ] .
                                                                       0

                  {{  OF  }                   [AT <Location>    ]}
{DISPLAY }        {{RADIUS} <radius> <units>  [<symbol options> ]}
{GENERATE} CIRCLE { NAMED <list of circle entry names>            }  .

                [{FROM {<location>}}  3                 ]
                [{     {+         }}                    ]
{DISPLAY }      [{TO   {<location>}}                    ]
{GENERATE} TRACK[{     {+         }}                    ]
                [{<track options>  }  0                 ]
                [ NAMED <List of track entry names>     ]  .

{DISPLAY }  {LOCATION            } [SIZE <size option>         ]
{GENERATE}  {AZIMUTH             } [COLOR <color name>         ]
            {DISTANCE            } [LINE TYPE <line type>      ]
            {CUMULATIVE DISTANCE } [PROTECT                    ]
            {TOTAL DISTANCE      } [GROUPED                    ]
                                   [GROUPED SEPARATELY         ]
                                   [NAME <name to be assigned> ]

        {SYMBOLS}      {NONE                       }
  LIMIT {CIRCLES}  TO  {ALL                        }
        {TRACKS }      {*                          }
                       {<names of subsets of data> }
```

EXHIBIT GIP120-1

GIP PROJECT

Retrieve the PCS file you saved in lesson GIP110 and display the map on your screen. Then, make the following changes.

1. Add a track, a symbol, and a circle using DISPLAY commands. Use the name option of the DISPLAY command to give each a name.

2. Enter the GENERATE commands which will make the track, symbol, and circle permanent.

3. Display your map again. Then, add one of the distance options to the track.

4. Finally, enter the commands which will delete the symbols and tracks from your map, then display your map again.

EXHIBIT GIP120-2

BASIC LANGUAGE PROGRAMMING (H6000-CDT) COURSE EXHIBITS

SAMPLE PROGRAM USING THE INPUT STATEMENT

```
10    PRINT "THIS PROGRAM GIVES YOU AGE IN THE YEAR 2000";
20    PRINT "DO YOU WANT TO KNOW";
30    INPUT A$
40    IF A$="NO" GOTO 100
50    PRINT "WHAT IS YOUR PRESENT AGE";
60    INPUT A
70    PRINT "WHAT YEAR IS THIS";
80    INPUT B
90    PRINT "YOU'LL BE";A+2000-B; "YEARS OLD IN THE YEAR 2000"
100   END

BRN
THIS PROGRAM GIVES YOUR AGE IN THE YEAR 2000
DO YOU WANT TO KNOW?  YES
WHAT IS YOUR PRESENT AGE?  29
WHAT YEAR IS THIS?  1983
YOU'LL BE 46 YEARS OLD IN THE YEAR 2000.
```

EXHIBIT BSC020-1

EXAMPLE OF READ AND DATA STATEMENTS

```
10    READ A,B,C,D,E
20    PRINT A;B;C;D;E
30    PRINT "THE AVERAGE OF THESE NUMBERS IS";
40    PRINT (A+B+C+D+E)/5
50    DATA 98,76,83,92,87
60    END

BRN
 98    76    83    92    87
THE AVERAGE OF THESE NUMBERS IS 87.2
```

EXHIBIT BSC020-2

EXAMPLE OF USING BRANCHES VS. NOT USING BRANCHES

```
10   LET A=1
20   LET X=A**2
30   PRINT X
40   PRINT "VALUE OF A IS", A
50   LET A=2
60   LET X=A**2
70   PRINT X
80   PRINT "VALUE OF A IS", A
90   LET A=3
100  LET X=A**2
110  PRINT X
120  PRINT "VALUE OF A IS", A
130  LET A=4
140  LET X=A**2
150  PRINT X
160  PRINT "VALUE OF A IS", A
170  END

BRN
 1
VALUE OF A IS 1
 4
VALUE OF A IS 2
 9
VALUE OF A IS 3
 16
VALUE OF A IS 4
```

```
10   IF A>=4 THEN 70
20   LET A=A+1
30   LET X=A**2
40   PRINT X
50   PRINT "VALUE OF A IS", A
60   GOTO 10
70   END

BRN
 1
VALUE OF A IS 1
 4
VALUE OF A IS 2
 9
VALUE OF A IS 3
 16
VALUE OF A IS 4
```

EXHIBIT BSC020-3

EXAMPLE OF THE IF STATEMENT

```
     10   LET A=99
     20   IF A =100 THEN 50
     30   PRINT A; "IS NOT EQUAL TO 100"
     40   STOP
     50   PRINT A; "IS EQUAL TO 100"
     60   END

     BRN
     99 IS NOT EQUAL TO 100

IF WE CHANGE LINE 10 TO:
     10 LET A=100

THEN:
     BRN
     100 IS EQUAL TO 100
```

EXHIBIT BSC020-4

EXAMPLE OF THE GOTO STATEMENT

```
10   LET A=5
20   LET B=1
30   LET C=A*B
40   PRINT C
50   LET B=B+1
60   IF B=5 THEN 80
70   GOTO 30
80   END

BRN
5
10
15
20
```

EXHIBIT BSC020-5

EXAMPLE OF THE ON GOTO STATEMENT

```
 10   INPUT A
 20   LET X= SQR(A)
 30   ON X GOTO 50, 60, 80
 40   PRINT "BRANCH 1"
 50   GOTO 10
 60   PRINT "BRANCH 2"
 70   GOTO 10
 80   PRINT "BRANCH 3"
 90   GOTO 10
100   END

BRN
?1
BRANCH 1
?4
BRANCH 2
?9
BRANCH 3
?16
INVALID COMPUTED GOTO IN 30
```

EXHIBIT BSC020-6

EXAMPLE FOR BRANCHING REVIEW

```
10   PRINT "I AM THINKING OF A NUMBER FROM 1 TO 10"
20   PRINT "GUESS MY NUMBER";
30   INPUT G
40   IF G=7 THEN 70
50   PRINT "WRONG! GUESS AGAIN";
60   GOTO 30
70   PRINT "THAT'S RIGHT"
80   END

BRN
I AM THINKING OF A NUMBER FROM 1 TO 10
GUESS MY NUMBER? 2
WRONG! GUESS AGAIN? 4
WRONG! GUESS AGAIN? 5
WRONG! GUESS AGAIN? 7
THAT'S RIGHT
```

EXHIBIT BSC020-7

BASIC PROGRAMS PROJECTS

Project 1

Write a simple BASIC program with remark comments, that will:

a. Use the input statement to read in a value.

b. Use the print statement to prompt for the value.

c. Calculate and then print the square roots of 25, 25.1, 37, 93, 100 and any other values you care to input.

Hint: Review Exhibits BSC020-1 and BSC020-6.

Project 2

Write another simple BASIC program that will:

a. Calculate the square and cube of the integers from 1 to 10.

b. Print out the integer, its square, and its cube.

Hint: Review Exhibits BSC020-6 and BSC020-7.

EXHIBIT BSC020-8

USING A "FOR NEXT" LOOP

```
010  FOR A=1 TO 10
020  PRINT A, A*2, A*3
030  NEXT A
040  END

BRN
              1              1              1
              2              4              8
              3              9             27
              4             16             64
              5             25            125
              6             36            216
              7             49            343
              8             64            512
              9             81            729
             10            100           1000
```

EXHIBIT BSC030-1

USING THE STEP STATEMENT

```
010  FOR A=1 TO 10 STEP .5
020  PRINT A, A*2, A*3
030  NEXT A
040  END

BRN
              1              1              1
            1.5           2.25          3.375
              2              4              8
            2.5           6.25         15.625
              3              9             27
            3.5          12.25         42.875
              4             16             64
            4.5          20.25         91.125
              5             25            125
            5.5          30.25        166.375
              6             36            216
            6.5          42.25        274.625
              7             49            343
            7.5          56.25        421.875
              8             64            512
            8.5          72.25        614.125
              9             81            729
            9.5          90.25        857.375
             10            100           1000
```

EXHIBIT BSC030-2

USING CHARACTER DATA WITH THE "FOR NEXT" LOOP

```
010  FOR K=1 TO 5
020  PRINT "XOXOXOXOXOX"
030  NEXT K
040  END

BRN

XOXOXOXOXOX
XOXOXOXOXOX
XOXOXOXOXOX
XOXOXOXOXOX
XOXOXOXOXOX
```

EXHIBIT BSC030-3

NESTED LOOPS

```
          LEGAL                                   ILLEGAL

     100 FOR A=1 TO 5                        100 FOR A=1 TO 5
             -                                       -
             -                                       -
             -                                       -
     200 FOR B=3 TO 9                        200 FOR B=3 TO 9
             -                                       -
             -                                       -
             -                                       -
     300 NEXT B                              300 NEXT A
             -                                       -
             -                                       -
             -                                       -
     400 FOR C=9*3 TO 9*4                    400 FOR C=9*3 TO 9*4
             -                                       -
             -                                       -
             -                                       -
     500 NEXT C                              500 NEXT B
             -                                       -
             -                                       -
             -                                       -
     600 NEXT A                              600 NEXT C
```

EXHIBIT BSC030-4

SUBSCRIPTS

A	A(5)			A(5,5)		
A	A(1)	A(1,1)	A(1,2)	A(1,3)	A(1,4)	A(1,5)
	A(2)	A(2,1)	A(2,2)	A(2,3)	A(2,4)	A(2,5)
	A(3)	A(3,1)	A(3,2)	A(3,3)	A(3,4)	A(3,5)
	A(4)	A(4,1)	A(4,2)	A(4,3)	A(4,4)	A(4,5)
	A(5)	A(5,1)	A(5,2)	A(5,3)	A(5,4)	A(5,5)

EXHIBIT BSC030-5

USING ARRAYS

```
010  FOR 1=1 TO 3
020  FOR J=1 TO 5
030  READ B2(I,J)
040  NEXT J
050  NEXT I
060  FOR I=1 TO 3
070  FOR J=1 TO 5
080  PRINT B2(I,J);
090  NEXT J
100  PRINT
110  NEXT 1
120  DATA 2,3,-5,-9,4
130  DATA 1,-7,8,-1,9
140  DATA 5,7,-2,-3,-8
150  END

BRN

     2       3       -5       -9        4
     1      -7        8       -1        9
     5       7       -2       -3       -8
```

EXHIBIT BSC030-6

USING THE "DIM" STATEMENT

```
010  DIM A(6,10)
020  FOR I=1 TO 6
030  LET A(I,1)=I
040  FOR J=1 TO 10
050  LET A(1,J)=J
060  PRINT A(I,J);
070  NEXT J
080  PRINT
090  NEXT I
100  END

BRN

          1     2     3     4     5     6     7     8     9    10
          2     0     0     0     0     0     0     0     0     0
          3     0     0     0     0     0     0     0     0     0
          4     0     0     0     0     0     0     0     0     0
          5     0     0     0     0     0     0     0     0     0
          6     0     0     0     0     0     0     0     0     0
```

EXHIBIT BSC030-7

SORT EXAMPLE USING AN ARRAY

```
010  DIM A(15)
020  PRINT "ENTER FIRST NUMBER";
030  INPUT A(1)
040  FOR I=2 TO 15
050  PRINT "ENTER NEXT NUMBER";
060  INPUT A(I)
070  NEXT I
080  FOR I=1 TO 14
090  FOR J=I+1 TO 15
100  IF A (I)<=A(J) THEN 140
110  T=A(I)
120  A(I)=A(J)
130  A(J)=T
140  NEXT J
150  NEXT I
160  FOR I=1 TO 15
170  PRINT A(I);
180  NEXT I
190  END
```

```
BRN

ENTER FIRST NUMBER?12
ENTER NEXT NUMBER?34
ENTER NEXT NUMBER?56
ENTER NEXT NUMBER?78
ENTER NEXT NUMBER?90
ENTER NEXT NUMBER?9
ENTER NEXT NUMBER?5
ENTER NEXT NUMBER?.14
ENTER NEXT NUMBER?.8
ENTER NEXT NUMBER?.007
ENTER NEXT NUMBER?-54
ENTER NEXT NUMBER?-32
ENTER NEXT NUMBER?-96
ENTER NEXT NUMBER?-.8
ENTER NEXT NUMBER?5
-96 -54    -32    -.8   .007    .14     .8      5      5      9     12     34
 56  78     90
```

EXHIBIT BSC030-8

USING MATRIX OPERATIONS

```
010  DIM A(2,3),B(2,3),C(2,3)
020  DIM D(2,3),M(2,3),T(3,2)
030  MAT READ A,B
040  REM * FIND THE SUM OF MATRIX A AND B
050  MAT S+A+B
060  MAT PRINT S
070  REM * FIND THE DIFFERENCE OF MATRIX A AND B
080  MAT D=A-B
090  MAT PRINT D
100  REM * MULTIPLY MATRIX A BY 2
110  MAT M=(2)*A
120  MAT PRINT M
130  REM * TRANSPOSE MATRIX B
140  MAT T=TRN(B)
150  MAT PRINT T
160  DATA 1,2,3,4,5,6
170  DATA 6,5,4,3,2,1
180  END

BRN

                 7              7              7
                 7              7              7

                -5             -3             -1
                 1              3              5

                 2              4              6
                 8             10             12

                 6              3
                 5              2
                 4              1
```

EXHIBIT BSC040-1

GOSUB AND RETURN STATEMENTS

```
010  PRINT "            A","            B","            C","            GCD"
020  READ A,B,C
030  LET X=A
040  LET Y=B
050  GOSUB 200
060  LET G=X
070  LET Y=C
080  GOSUB 200
090  PRINT A,B,C,G
100  GOTO 20
110  DATA 60,90,120
120  DATA 38456,64872,98765
130  DATA 32,384,72
200     LET Q=INT(X/Y)
210     LET R=X-Q*Y
220     IF R=0 THEN 300
230     LET X=Y
240     LET Y=R
250     GOSUB 200
300     LET G=Y
310     RETURN
320  END
```

```
BRN

                A              B              C             GCD
               60             90            120              60
            38456          64872          98765               1
               32            384             72              24

OUT OF DATA IN 20
```

EXHIBIT BSC040-2

NESTED SUBROUTINES

```
       -
       -
       -
100 GOSUB 200
       -
       -
       -
200 *** THIS IS SUBROUTINE 1 ***
       -
       -
       -
250 GOSUB 400
       -
       -
       -
300 RETURN
       -
       -
       -
400 *** THIS IS SUBROUTINE 2 ***
       -
       -
       -
450 RETURN
       -
       -
       -
480 END
```

EXHIBIT BSC040-3

SUBROUTINE REVIEW

```
010  REM  THIS PROGRAM CALCULATES THE MEAN AVERAGE OF A SET
015  REM  OF NUMBERS.
020  REM  MAIN PROGRAM
030  GOSUB 200
040  FOR J=1 TO 4
050  READ N
060  GOSUB 100
070  LET M=S/N
080  PRINT N,S,M
090  NEXT J
095  STOP
099  REM  SUBROUTINE TO MANIPULATE DATA
100     FOR K=1 TO N
110     READ X(K)
120     NEXT K
130     LET S=0
140     FOR K=1 TO N
150     LET S=S+X(K)
160     NEXT K
170     RETURN
199  REM  SUBROUTINE TO PRINT COLUMN HEADINGS
200  PRINT "       NUMBER        SUM               MEAN"
210     RETURN
220  DATA 3,87,72,93
230  DATA 3,81,86,90
240  DATA 4,95,97,83,89
250  DATA 5,88,92,85,84,91
260  END

BRN

             NUMBER          SUM            MEAN
               3             252              84
               3             257        85.66667
               4             364              91
               5             440              88
```

EXHIBIT BSC040-4

MATH AND PRINT FUNCTIONS

FUNCTION	DESCRIPTION
ABS(X)	RETURNS THE ABSOLUTE VALUE OF X
ATN(X)	RETURNS THE RADIAN ARCTANGENT OF THE VALUE X
CLG(X)	RETURNS THE COMMON LOGARITHM (BASE 10) OF X
COS(X)	RETURNS THE COSINE OF THE RADIAN VALUE X
COT(X)	RETURNS TO COTANGENT OF THE RADIAN VALUE X
EXP(X)	RETURNS THE NUMERIC CONSTANT 2.71828 RAISED TO THE POWER OF X
INT(X)	RETURNS THE LARGEST INTEGER VALUE OF X
LOG(X)	RETURNS THE NATURAL LOGARITHM OF THE VALUE X
RND(X)	RETURNS RANDOM NUMBERS, (X) IS OPTIONAL
SGN(X)	RETURNS AN INDICATION OF THE SIGN (+, -, OR 0) OF X
SIN(X)	RETURNS THE SINE OF THE RADIAN VALUE X
SPC(X)	CAUSES X SPACES TO BE PRINTED
SQR(X)	RETURNS THE SQUARE ROOT OF X
TAB(X)	CAUSES OUTPUT TO BE PRINTED AT THE X COLUMN ON THE OUTPUT LINE
TAN(X)	RETURNS THE TANGENT OF THE RADIAN VALUE X

EXHIBIT BSC050-1

EXAMPLE OF A USER-DEFINED FUNCTION

```
010  REM CONVERT CELSIUS TO FAHRENHEIT
020  DEF FNT(X)=(9/5)*X+32
030  PRINT "DO YOU WANT TO CONVERT";
035  INPUT A$
040  IF A$="NO" THEN 80
050  PRINT "CELSIUS TEMPERATURE";
055  INPUT C
060  PRINT "EQUALS";FNT(C);"FAHRENHEIT"
070  GOTO 30
080  END
```

```
BRN

DO YOU WANT TO CONVERT?YES
CELSIUS TEMPERATURE?0
EQUALS    32 FAHRENHEIT
DO YOU WANT TO CONVERT?YES
CELSIUS TEMPERATURE?200
EQUALS   392 FAHRENHEIT
DO YOU WANT TO CONVERT?YES
CELSIUS TEMPERATURE?112
EQUALS   233.6 FAHRENHEIT
DO YOU WANT TO CONVERT?NO
```

EXHIBIT BSC050-2

USING STRING ARRAYS

```
010  DIM X$(3)
020  FOR I=1 TO 3
030  INPUT X$(I)
040  NEXT I
050  FOR I-1 TO 3
060  PRINT X$(I)
070  NEXT I
080  END

BRN

?ABCD
?EFGH
?IJKL
ABCD
EFGH
IJKL
```

EXHIBIT BSC060-1

RELATIONAL OPERATORS USED WITH STRING VARIABLES

OPERATOR	EXAMPLE	MEANING
=	A$ = B$	THE STRING A$ AND B$ ARE EQUIVALENT IN CHARACTERS, NOT INCLUDING TRAINING SPACES
<	A$ < B$	THE STRING A$ OCCURS BEFORE B$ IN ALPHABETICAL SEQUENCE
<=	A$ <= B$	THE STRING A$ IS EQUIVALENT TO OR OCCURS BEFORE B$ IN ALPHABETIC SEQUENCE
>	A$ > B$	THE STRING A$ OCCURS AFTER B$ IN ALPHABETICAL SEQUENCE
>=	A$ >= B$	THE STRING A$ OCCURS AFTER B$ IN ALPHABETICAL SEQUENCE
<>	A$ <> B$	THE STRINGS A$ AND B$ ARE NOT EQUIVALENT

EXHIBIT BSC060-2

STRING FUNCTIONS

FUNCTION	USE
ASC(X$)	PROVIDES THE NUMERIC VALUE FOR THE SPECIFIED CHARACTER
LEN(X$)	INDICATES THE NUMBER OF CHARACTERS IN THE SPECIFIED STRING, X$
SST(X$,N1,N2)	INDICATES A SUBSTRING IN THE SPECIFIED STRING, X$, THAT BEGINS AT POSITION N1 AND IS N2 CHARACTERS LONG
STR$(N)	CONVERTS THE SPECIFIED NUMERIC VALUE TO A CHARACTER
VAL(X$)	COMPUTES THE NUMERIC VALUE OF THE NUMERIC CHARACTERS REPRESENTED BY THE SPECIFIED STRING, X$

EXHIBIT BSC060-3

BASIC Programming Project 3

Write a BASIC program that will:

a. Read in data that consists of three numbers: a country code, the number of Fighter Squadrons in that country, and the number of Bomber Squadrons in that country.

b. Calculate the total threat to you, by country, and place that figure into the array. Threat is equal to ten times the numbcr of Bomber Squadrons plus three times the number of Fighter Squadrons.
(i.e. T=(3*F)+(10*B))

c. Print out the data reading and the threat results in a format similar to the following:

```
                THREAT ASSESSMENT SUMMARY
COUNTRY CODE          X    X    X
FIGHTER SQUADRONS     X    X    X
BOMBER SQUADRONS      X    X    X
THREAT                X    X    X
```

d. Input data:

```
1,15,4
14,24,3
18,10,5
```

The first numbers in each line are the country codes while the second and third numbers in each row are the Fighter and Bomber squadron totals for that country.

Hint: Your array should be 2-dimensional and large enough to add the threat totals into.

EXHIBIT BSC060-4

BASIC Programming Project 4

Write a BASIC program that will:

a. Read in data that consists of three numbers: a ward code, the number of beds in that ward, and the number of nurses in that ward.

b. Calculate the emergency capacity of each ward and place that figure into the array. Emergency Capacity is equal to three times the number of beds plus ten times the number of nurses. (i.e. E=(3*B)+(10*N))

c. Print out the data read in and the emergency capacity results in a format similar to the following:

```
                 EMERGENCY CAPACITY SUMMARY

      WARD CODE           X    X    X
      BEDS PER WARD       X    X    X
      NURSES PER WARD     X    X    X
      EMERGENCY CAPACITY  X    X    X
```

d. Input data:

1,15,4
14,24,3
18,10,5

The first numbers in each line are the ward codes while the second and third numbers in each row are the total beds and nurses in the wards.

Hint: Your array should be 2-dimensional and large enough to add the emergency capacity into.

EXHIBIT BSC060-5

BASIC Programming Project 5

1. Write a BASIC computer program to produce a Procurement Planning report for an item called X.

2. Your program should be structured to accept the following inputs by the terminal.

   a. The usage forecasts for item X for months 1 to 12.
   b. The beginning inventory.
   c. The desired ending inventory.

3. Then it should calculate the average number of item X to be procured monthly.

$$\frac{\text{Avg procurement}}{\text{month}} = \frac{\text{Total Usage + Ending Inventory - Beginning Inventory}}{\text{12 months}}$$

Total Usage = the total of the twelve usage forecasts.

4. Usage forecasts by month.

1: 180	5: 1116	9: 240
2: 192	6: 540	10: 972
3: 480	7: 516	11: 168
4: 300	8: 288	12: 192

5. The beginning inventory is 972 while the desired ending inventory is 480.

6. Your output should be close to the following format.

```
                           PROCUREMENT PLANNING FOR ITEM X
MO          BEGINNING INVENTORY PROCUREMENT    USAGE FORECAST END INVENTORY

 1                   972             X                X          X
 2                     X             X                X          X
 .                     .             .                .          .
 .                     .             .                .          .
 .                     .             .                .          .
12                     X             X                X        480
```

EXHIBIT BSC060-6