1103 CENTRAL EXCHANGE

NEWSLETTER NUMBER 8

February 1956

Remington Rand
DIVISION OF SPERRY RAND CORPORATION
ENGINEERING RESEARCH ASSOCIATES DIVISION
1902 WEST MINNEHAHA AVE. ST. PAUL W4, MINNESOTA

NEWSLETTER NUMBER 8
February 1956

EDITOR'S PAGE

Several months ago a cooperative organization of 1103A users was formed under the name of USE - Univac Scientific Exchange. The early 1103A purchasers desired to go beyond the Central Exchange with its voluntary exchange of completed routines and actively cooperate in the early stages of program planning and assignment of programming manpower. A description of this organization is enclosed.

Abstract cards are now available for all Central Exchange routines. Sets of these are mailed automatically to all 1103 installations. Additional sets will be supplied upon request.

The "Notes on the Timing of the Controlled Reproducer" which were omitted from the last Newsletter are enclosed with this one.

Peggy Johnson
Systems Analysis Department

NEWSLETTER NUMBER 8
February 1956

REPORTS

CONVAIR The revised version of FLIP is now available. This program, FLIP III, is compatable with the old but is better than 20% faster. A great deal of the description of FLIP has been rewritten to render it more comprehensible on first reading. The subroutines which are in FLIP to date are listed as enclosures. A plastic encased card with all FLIP commands and the constant pool is being prepared for use at Convair. A reasonable number of these will be supplied to others upon request.

A recent addition to FLIP is a package consisting of an alarm routine, an octal print routine, and a Flexprint routine. The octal and flexprint subroutines afford a number of choices as to format and what portion of the accumulator is printed on the typewriter. Another addition is the Double Entry Table Interpolation and Lookup which uses parts of FLIP.

A magnetic tape corrector routine MOCK VIII has been written to facilitate changes in programs stored on magnetic tape. The changes are read from punched cards.

Charles Swift writes that Convair is considering a new use of the interrupt feature with their 1103. They propose achieving a simplified general trace by using the External Function instruction to initiate a "delayed interrupt". If the time delay is just long enough to permit one jump to be executed before the interrupt signal, a trace routine can set up the interruption of the next main routine command.

Editorial Comment: To illustrate, assume the trace routine is at addresses w through w+n and the program which is to be traced is stored beginning at address v.

```
F3        RJ      w+n     w
.
.
.                                  ┐
w                                  │
.                                  │
.                                  ├ Trace routine
.                                  │
w+n-1     EF      α       β        │
w+n       MJ      0       v        ┘
.
.
.                                  ┐
v                                  │
.                                  ├ Routine to be traced
.                                  │
.                                  │
```

Tracing would operate as follows: Start the computer at address w+n-1. The "delayed interrupt" is set up and control is transferred to address v by the jump in w+n. During execution of the instruction at the v the interrupt occurs and control is returned to the trace routine by means of the Return Jump in F_3.

This method of achieving a "hot trace" provides an exceedingly simple means of tracing jump instructions. However, since neither the Repeat instruction nor a repeated instruction can be interrupted, they will not be traced. The jump in F_1 which terminates the repeat sequence is traced so that the end result of the repeat is recorded. In many cases this will have the advantage of eliminating copious and unnecessary output (such as during block transfers and table look-up).

NEWSLETTER NUMBER 8
February 1956

[illegible] A Magnetic Tape Block Counter routine has been written. This routine automatically keeps track of the position of the reading head of each of the magnetic tape units provided a Return Jump to the block counter routine is written after each magnetic tape order.

A Card Title Subroutine has also been submitted for distribution. This routine was designed to allow the programmer wishing tabular output to make use of punched card equipment to produce quickly a neat printout of flexible format. The routine converts alpha-numeric information in Flexowriter code into IBM code on cards for use as page headings, column headings, and line titles. These cards, together with data cards, can be listed to produce tabular formats, for example, on the IBM 407. Input is the Flexowriter codes packed 6 to an 1103 word. Output is cards punched on the Controlled Reproducer.

RAMO-WOOLDRIDGE A change has been instituted with regard to most service routines. The 40000 entries to service routines were provided for flexibility in moving routines about on the drum. However, the program entry depended on the location of the routine and hence invalidated this facility. Therefore cell 40020 is now reserved as a common program exit. Since the basic code is unchanged except for allowing for this procedure, only the revised descriptions are published herein and not the codings. The coding may be obtained upon request. The following service routines have been ammended: RW-73, The Flexowriter Memory Dump; RW-90, The Bioctal Memory Dump; RW-100, The Octal Card Dump; RW-102, Changed Word Post-Mortem; RW-92, Storage to Magnetic Tape Transfer; RW-93, Magnetic Tape to Storage Transfer; RW-103, Binary Card Read-In; RW-63, Ferranti Input Routine.

In addition to maintaining the library for current operation thought is being directed to the anticipated installation of the model 1103A. This includes the preparation of a two pass assembly program and further mechanization of the program checkout procedure. Some changes may also be made in the Routine Library structure.

A number of examples have been run to test the new Linear Matrix Equation Solver. Matrices of order up to 16 have been inverted. Accuracy for the largest matrix (size 16) was approximately 7 to 8 places. This routine is now being prepared for distribution.

An integral root routine is checked out as is a "Hot Trace" for SNAP. The former is enclosed with this Newsletter.

Another fixed point Definite Integral Evaluation Routine is now available. The difference between RW-89 and this routine is that the error terms are of fourth and fifth order respectively. There is also a floating point version of this routine. Fixed point logarithm and exponential routines are also enclosed.

The revised write-up of the Ramo-Wooldridge One Pass Assembly Routine (RW-72) which is in Newsletter 7 should be corrected to read "... the routine prints GMP-0" in the last sentence of paragraph 2 page 9.

A routine has been coded which reads fixed point decimal numbers from cards, converts the numbers to their octal equivalents and stores them in the computer at locations specified by a base address and the location number on the card.

Copies of "An Integrated Computation System for the ERA-1103" are available upon direct request to Dr. Walter Bauer. This is the text of a talk presented to the Association for Computing Machinery National Meeting at Philadelphia in September 1955.

A complex arithmetic version of SNAP (the Floating point arithmetic package) has been checked out. Also, a Fixed Point Card Output routine has been prepared. Work is now progressing on an algebric equation solver which will find all real and complex roots.

REMINGTON RAND UNIVAC For several months now we have been operating the Serial 9 1103 computer here in St. Paul. One of the principle uses of the computer is for mechanized computer design. The design process consists of formulating expressions for the logic of a computer system, simulating this logic with an 1103 program, and assigning physical specifications of the designed system, also by means of an 1103 program. A number of programs have been prepared to facilitate the automatic processing of equations expressing the logic of a computer system. Such tasks are performed as verifying the equation structure with regard to rules regulating the formulation (such as timing restrictions, number of inputs to a core, etc.) and sorting equations with respect to various criteria. The processes of simulation and preparation of manufacturing and maintenance aids are almost entirely mechanized. 1103 programs simulate whichever logical operations have been selected to be checked by choosing the equations involved in the operation and effecting the actual function of the appropriate components in the equations. Manufacturing aids which are produced by the 1103 programs are parts lists, wiring tabulation forms, assignment of unit assembly positions to type cards, and assignment of relative positions in the completed system to the unit assemblies. The programs are coded to effect the most efficient but realistic solution to a manufacturing problem which allows several solutions.

A description of this mechanized design process is available upon request.

Enclosed with the last Newsletter was a description of the use of the "dead Space" on the magnetic drum for storage of loading routines for the Model 1103A computer. A routine has been coded for storage in the dead space which simulates the MT START of the Model 1103. That is, one block of 120 words is read from tape unit I to core storage starting with address 00001. The computer stops following the transfer to permit switching from ABNORMAL DRUM to NORMAL DRUM.

The following is extracted from a report from Leon Dominick of the Convair maintenance crew:

"A table of failures has been compiled concerning the reliability of the non-diagnostic test at the Convair installation. The period of time covered by this table is 15 December 55 thru 15 February 56."

"The non-diagnostic test was run with reduced heaters and margins after every preventative maintenance period and before Convair went on the machine. At no time did failures occur during these tests when run with margins and reduced heaters as appropriate to each part of the test."

"Convair has been running the non-diagnostic test, on their time, on an average of twice per 24 hour period, whenever they suspected machine trouble without definite evidence to support their suspicions."

"The table lists 18 times when the non-diagnostic test failed for Convair. With two exceptions, a malfunction of the computer was located which caused the test failure. In those two instances where failures of the test occured with no trouble found, it is believed the failures were of a very random nature, and that they were caused by arcing of the 300 volt generator. On only one occasion did the non-diagnostic test fail to show an error when the error was obvious through examination of the indicators on the maintenance console. This error was a failure to write in one digit on drum."

"Convair has gained enough confidence in the reliability of the non-diagnostic test to usually agree to its use as a check on the reliable operation of the machine."

NEWSLETTER NUMBER 8
February 1956

The following question was raised here concerning the conversion of floating binary numbers to floating decimal for output on the High Speed Printer attached to the 1103A: is there actually sufficient time available within the print cycle for the 1103A to convert these numbers when the printer is operating at the rate of 600 lines per minute? A conversion routine has been written which prints eight numbers per line. Each number consists of an eight digit mantissa with sign and a two digit characteristic with sign. The average time to convert and set up a line or blockette is 35.3 ms. (For decimal numbers between 10^{-10} and 10^{10}). The maximum time to convert and setup a line or blockette is 54.4 ms. (For decimal numbers 10^{29}). Since 90 milliseconds are available for computation during the print cycle, the output rate is definitely printer-limited. In fact the rate of output is at least doubled by recording on magnetic tape for off-line printing.

The method of conversion is as follows: the mantissa is scaled 2^{36} while the binary characteristic is reduced to zero. A count of the decimal characteristic is retained while the binary characteristic is reduced. By using a split positive entry with a shift of two followed by a split. add with a shift of one, the mantissa is multiplied by ten. A repeat of these instructions seven more times yields an eight digit decimal mantissa. By use of the split commands, six decimal digits are placed in each output word and the excess threes added to produce the proper Univac codes.

<u>WHITE SANDS</u> An Octal Card Reed routine has been coded which reads from cards containing any number of octal words from 1 to 6. An <u>Octal Card Dump</u> is also completed.

A short routine has been devised for checking the accuracy of a paper tape prepared on the High Speed Punch. After the bioctal tape has been punched, this routine will read the tape and compare it with the contents of storage from which the tape was punched. Any discrepancy results in a printout of both the correct and incorrect words.

NEWSLETTER NUMBER 8
February 1956

ENCLOSURES

CV-109	MOCK VIII-a magnetic tape corrector routine
OR-110	Magnetic Tape Block Counter Routine
RW-111	Fixed Point Definite Integral Evaluation
RW-112	Floating Point Definite Integral Evaluation
RW-113	Fixed Point Natural Logarithm
RW-114	Fixed Point Exponential (e^x)
CV-115	Alarm, Octal, and Flexprint Package
RW-116	N^{th} Root Routine
RW-117	Fixed Point Decimal Card Read-In Routine

REVISIONS

RW-73	Flexowriter Memory Dump
RW-90	Bioctal Memory Dump
RW-100	Octal Card Dump
RW-102	Changed Word Post-Mortem
RW-92	Storage to Magnetic Tape
RW- 93	Magnetic Tape to Storage Transfer
RW-103	Binary Card Read-In
RW-63	Ferranti Input Routine
CV-11	FLIP III- a floating point subroutine system

M. T. Routine for Convair Service Routines
M. T. Routine for FLIP (revised)
Activator (revised)
Complex Arithmetic
Inspect and Change (CV-12)
Bioctal Paper Tape (CV-81)
Charactron Trace, Processer (CV-77)
Magnetic Tape Storage
Differential Equations (CV-97)
Card to Paper Tape (CV-94)
Bessel Functions
Charactron Output (CV-98)
Card Output
Card Input
Alarm Routine (FLIP I)
Arctangent and Arccotangent (FLIP I)
Exponential (FLIP I)
Input Sum Check (revised)
Logarithm (FLIP I)
Square Root (FLIP I)
Loader (FLIP I)
Loader Parameters (FLIP I)
Basic FLIP (revised)
Print and Punch (FLIP I)
Flexowriter Input (FLIP I)
Sine Cosine (FLIP I)
Charactron Trace, Activator
Charactron Trace, Concurrent
Permanent Constants (FLIP I)
Assembly Routine (FLIP)
Trace (FLIP I)
Alarm, Octal, and Flexprint (CV-115)

NEWSLETTER NUMBER 8
February 1956

CV-52 Flexprint Subroutine

ENCLOSURES

OR-118	Card Title Subroutine- a routine For Formating card output
WS-119	Octal Card Read
WS-120	Octal Card Dump
WS-121	Check on High Speed Punch When Punching Bioctal Tapes
CV-122	4 Point Lagrange Interpolation Subroutine
CV-123	Determinant Evaluation Package-Real

3:21	Notes on the timing of the Controlled Reproducer
8:22	Formation of USE- a Cooperative Organization of 1103 A Users

PX 71900-8

1103 Central Exchange

Distribution List

Byron D. Smith
Remington Rand Univac
Division of Sperry Rand Corporation
1902 West Minnehaha Avenue
St. Paul 4, Minnesota
(Midway 6-9601, Extension 312)

Commanding General
White Sands Proving Ground
New Mexico
Attention: Dr. John Titus, Chief
Computing Branch
Flight Determination
Laboratory

National Security Agency
Mrs. Dorothy T. Blum
4314 North Pershing Drive
Arlington 3, Virginia

Director, National Security Agency
Washington 25, D. C.
Attention: 3512

The Ramo-Wooldridge Corporation
Mr. Werner Frank
Computer Systems
8820 Bellanca Avenue
Los Angeles 45, California

Mr. L. R. Turner
2693 Westwood Road
Westlake, Ohio

Mr. W. Bruce Taylor
Johns Hopkins University
Operations Research Office
7100 Connecticut Avenue
Chevy Chase, Maryland

Commander
Air Force Armament Center
Attention: ACVMC
Eglin Air Force Base, Florida

Commander
Wright Air Development Center
Attention: WCRRU, Leonard B. Hall
Wright-Patterson Air Force Base, Ohio

Mr. Ben Ferber
Digital Computing Laboratory
Convair
San Diego, California

Mr. Donald B. Houghton, Chief
Analysis Section
Laboratories for Research and Development
The Franklin Institute
20th and Parkway
Philadelphia 3, Pennsylvania

Librarian, Numerical Analysis Research
University of California
405 Hilgard Avenue
Los Angeles 24, California

Mr. Albert E. Smith
Code 223
Bureau of Ships
Navy Department
Washington 25, D.C.

Dr. Harry Polachek
Head of Applied Mathematics Laboratories
Code 800
David Taylor Model Basin
Washington 7, D.C.

1103 Central Exchange

Distribution List

Mr. Douglas T. Ross
Massachusetts Institute of Technology
Servomechanisms Laboratory
Department of Electrical Engineering
Cambridge 39, Massachusetts

Mr. John M. Frankovich
Digital Computer Laboratory
Massachusetts Institute of Technology
211 Massachusetts Avenue
Cambridge 39, Massachusetts

Dr. George F. Cramer
Electronic Computer Department
19th Floor
Remington Rand
Division of Sperry Rand Corporation
315 Fourth Avenue
New York City 10, New York

Mr. A.E. Roberts, Vice President
General Kinetics, Incorporated
555 23rd Street South
Arlington 2, Virginia

Dr. Grace M. Hopper
Remington Rand
Division of Sperry Rand Corporation
1624 Locust Street
Philadelphia 3, Pennsylvania

Mr. Erwin Tomash
Electronic Computer Department
19th Floor
Remington Rand
Division of Sperry Rand Corporation
315 Fourth Avenue
New York City 10, New York

Mr. Randell E. Porter
Boeing Airplane Company
Seattle, Washington

Mr. Robert Simon
Electronic Computer Department
19th Floor
Remington Rand
Division of Sperry Rand Corporation
315 Fourth Avenue
New York City 10, New York

Mr. W.R. Hydeman
Suite 7
4218 Mayfield Road
Cleveland 21, Ohio

Mr. William A. Gerhart
120 Pear St.
Reading, Pennsylvania

Mr. Herbert Glantz
5225 South Eighth St.
Arlington 4, Virginia

Mr. Thomas T. Wilder
Remington Rand
Division of Sperry Rand Corporation
41 1st Street
San Fransisco, California

Mr. A.S. Baron
Westinghouse Electric Corporation
1405 Parker Road
Baltimore, Maryland

Mr. Karl E. Ball
Sandia Corporation
Sandia Base
Albuquerque, New Mexico

Mr. Theodore W. Helweg, ERA Engineer
Johns Hopkins University - ORO
7300 Pearl Street
Bethesda, Maryland

1103 Central Exchange

Distribution List

Mr. R. A. Cole
Data Processing Group
Pilotless Aircraft Division
Boeing Airplane Company
Seattle, Washington

Mr. Roy E. Hegler, ERA Engineer
ERA Rem-Rand 1103 Computer
Ramo-Wooldridge Corporation
5740 Arbor Vitae
Los Angeles 45, California

Mr. J. Allen Ovenstone
c/o Captain J.M. Armstrong, R.A.N.
Defense Research and Development Representative
Australian Joint Service Staff
P.O. Box 4837
Washington 8, D.C.

Mr. A.C. Hendrickson
Remington Rand
Division of Sperry Rand Corporation
Electronic Computer Department - 19th Floor
315 Fourth Avenue
New York City 10, New York

Mr. A. Christofferson, Jr.
P.O. Box 2228
San Diego, California

Mr. Kenneth R. Webster
203 Fourth Avenue
White Sands Proving Ground
New Mexico

Miss Ruth M. Davis
David Taylor Model Basin
Washington 7, D.C.

Mr. S.C. Marcus
Emerson Electric Manufacturing Company
111 Eighth Avenue
New York City, New York

Mr. John R. Pearson
Holloman Air Force Base
Telecomputing Corporation
133 East Santa Anita Avenue
Burbank, California

Mr. Ralph Stidham
Holloman Air Force Base
Telecomputing Corporation
133 East Santa Anita Avenue
Burbank, California

Mr. James R. Vine
Holloman Air Force Base
Telecomputing Corporation
133 East Santa Anita Avenue
Burbank, California

Mr. Allan R. Schoenfeld
Holloman Air Force Base
Telecomputing Corporation
133 East Santa Anita Avenue
Burbank, California

Mr. Allan Keane
Remington Rand
Divsion of Sperry Rand Corporation
Electronic Computer Department - 19th Floor
315 Fourth Avenue
New York City 10, New York

Mr. Stanley Rintel
Remington Rand
Division of Sperry Rand Corporation
Electronic Computer Departmnt - 19th Floor
315 Fourth Avenue
New York City 10, New York

1103 Cenral Exchange

Distribution List

Miss Jean E. Sammet
Sperry Gyroscope Company
Division of Sperry Rand Corporation
Mail Station R-210
Great Neck, New York

National Security Agency
Washington 25, D.C.
Attention: Dr. R.A. Leibler (NSA - 34)

Mr. Harold Dahlbeck
Electronic Computer Department
19th Floor
Remington Rand
Division of Sperry Rand Corporation
315 Fourth Avenue
New York City 10, New York

Mr. Edward L. Braun
Coordinator
MSD Computing Center
Research Laboratories
Lockheed Missile Systems Division
Van Nuys, California

Mr. John W. Young, Jr.
1929 South Quebec Street
Arlington 4, Virginia

Mr. Clem Grabner
7100th Support Wing
APO 633, New York, New York

Mr. John Wooten
Plant Number Two
Boeing Airplane Company
Wichita, Kansas

Mr. Robert H. Wallace
Remington Rand
Division of Sperry Rand Corporation
415 S. Main Ave.
San Antonio 5, Texas

Director, Evans Signal Laboratory
Belmar, New Jersey
Attention: Director, Physical Sciences Divisio
Mr. H. Stein
Computer Techniques Section

Mr. Robert G. Tantzen
Technical Analysis Division
HADC
Holloman, New Mexico

Mr. Robert L. Young
Director of Aero-Ballistics ACB
AFAC, Eglin Air Force Base, Florida

Mr. Theodore Dwight Buetell
Douglas Aircraft Company
3000 Ocean Park Blvd.
Santa Monica, California

Mr. Francis Newton Kurriss
Douglas Aircraft Company
3000 Ocean Park Blvd.
Santa Monica, California

Commanding General
Holloman Air Development Center
Holloman Air Force Base, New Mexico
Attention: M.C. Green HDTT

1103 Central Exchange

Distribution List

Mr. Thomas Duke
Telecomputing Corporation
128385 Saficoy Room 371
North Hollywood, California

Dr. Werner Leutert
Systems Division
Lockheed Missiles
Van Nuys, California

Dr. Clair Miller
California Research Corporation
576 Standard Ave.
Richmond, California

Flight Simulation Branch
Electromechanical Laboratories
White Sands Proving Grounds
New Mexico
Attention: Mr. Haryy Davis

Mr. T. B. Andrews
National Advisory Committee for Aeronautics
Langley Field, Virginia

Mr. Thomas R. Dines
22803 65th Place W.
Mount Lake Terrace
Washington

Dr. Ben Mount
AEC Westinghouse
Pittsburg, Pennsylvania

Mr. Jack Burke
Remington Rand Division of
Sperry Rand Corporation
Ferry at 4th Ave.
Pittsburg 22, Pennsylvania

Mr. A. Downing
AVRO Aircraft
Toronto, Canada

Attention: R. P. Blixt
Remington Rand Univac
Division of Sperry Rand Corporation
Box 1206
Melborne, Florida

Jack Rose
Remington Rand
Division of Sperry Rand Corporation
2601 Wilshire Blvd.
Los Angeles, California

Mr. Jim Murray
Remington Rand
Division of Sperry Rand Corporation
1615 "L" Street
Wahington 6, D.C.

Theodore S. Garfinkle
The Johns Hopkins University
Operations Research Office
7100 Connecticut Ave.
Chevy Chase, Maryland

Miss Arla E. Weinert
905 Fairvies Ave. Apt.#3
Washington 12,D.C.

Mr. Morgan Walker
DuPont Engineering Department
Wilmington 98, Delaware

Cornell Aeronautical Laboratory INC.
Buffalo, New York
Attention: Dr. F. Rosenblatt

Cornell Aeronautical Laboratory Inc.
Buffalo New York
Attention: Mr. G. Richmond

FX 71900-8

ANALYSIS
PREPARED BY R. M. Price
CHECKED BY
REVISED BY

PAGE ST 001-1
REPORT NO. ZM 491
MODEL All
DATE 12/12/55

MOCK VIII

The MOCK VIII is a magnetic tape corrector routine designed to facilitate changes in the increasing number of programs stored on magnetic tape.

Changes are read from punched cards and have the following formats:

	OLD WORD			BLOCK WORD		NEW WORD		
FORMAT I								
Col. 1	4	7	13	20	25	29	32	38
	CC	UUUUU	VVVVV	BBBB	WW	CC	UUUUU	VVVVV

	OLD WORD			BLOCK WORD		NEW WORD		
FORMAT II								
Col. 1	4	9	14	20	25	29	34	39
F	CCCC	XXXX	YYYY	BBBB	WW	CCCC	XXXX	YYYY

Format II is simply a reading and key-punching convenience (it is noted by an F in column one), any command including flip commands may be punched under FORMAT I (column one blank).

The first card must have the MT number punched in column one (it does not have any thing else punched in it) and it is followed by any number of change cards whose block numbers are in ascending sequence. A blank card signals the end of the changes.

The blocks of the magnetic tape are considered as being numbered from 0000 thru 3777 and the words of each block from 00 thru 37. The tape should be rewound when starting. BBBB and WW are octal numbers.

If the old word is not known simply leave that part of the card blank.

OPERATING INSTRUCTIONS

1. LOAD SROK LOAD CHANGE CARDS WITH TAPE INDICATOR CARD IN FRONT. (no prime)

2. MD START PAK 70377.

3. ERROR STOPS.
   1. "WRD" will print and the machine will stop with Q containing the old word from the tape, the block number in u of R, and the word number in v of R, when the old word on the tape is not the indicated one.
   2. "BLK" will print if the block number is out of sequence and the block and word numbers will appear in R as above.

IN EITHER CASE PRESS START TO IGNORE THIS CHANGE AND CONTINUE TO THE NEXT ONE.

Drum Allocation 70114 - 70377. This routine is not in standard form and is not to be modified. All constants and temporaries are included in the routine. It does not save or restore ES.

By: R. M. Price

07-109

MOCK VIII ST 001

70114	00000	45	00000	00001	
70115	00001	17	00000	00106	PICK 1ST CARD
70116	00002	37	00205	00121	READ 1ST CARD
70117	00003	55	00334	00036	MT# → Q, 00334
70120	00004	11	00117	10000	MASK → Q
70121	00005	53	00334	00043	SET WRITE
70122	00006	53	00334	00050	SET ADVANCE
70123	00007	53	00334	00051	SET READ
70124	00010	53	00334	00052	SET BACK
70125	00011	53	00334	00102	SET WRITE FINAL
70126	00012	53	00334	00106	SET REWIND
70127	00013	23	00320	00320	ZERO TO CURRENT BLK.NO.
70130	00014	16	00244	00042	SET 5A
70131	00015	16	00052	00111	SET 9A
70132	00016	23	00321	00321	CLEAR TALLY
70133	00017	37	00017	00020	
70134	00020	16	00245	00023	SET PARAM. STORE
70135	00021	37	00205	00121	READ ONE CARD
70136	00022	75	30003	00024	STORE
70137	00023	11	00330	30400	DATA
70140	00024	11	00333	20000	
70141	00025	47	00026	00251	BLANK CARD ?
70142	00026	21	00023	00246	↓ NO, UP STORE
70143	00027	21	00321	00236	UP TALLY
70144	00030	42	00245	00021	TALLY FULL?
70145	00031	23	00321	00236	YES, TALLY-1 → TALLY
70146	00032	15	00244	00034	SET STORE P_i
70147	00033	75	30003	00035	} STORE
70150	00034	11	30400	00330	} P_i
70151	00035	11	00226	10000	MASK → Q
70152	00036	51	00330	00322	00 0BBBB 00000 → TEMP, A
70153	00037	43	00226	00102	

PX 71900-8-109

By: R. M. Price

MOCK VIII ST 001

```
70154  00040  42 00320 00115   C.B.N. > BBBB?  YES→ALARM
70155  00041  43 00320 00110   C.B.N. = BBBB→YES
70156  00042  37 00042 00044   NO
70157  00043  65 00001 00340   WRITE PREVIOUS BLOCK
70160  00044  23 20000 00320   SET
70161  00045  23 20000 00236      ADVANCE
70162  00046  11 00226 10000          TAPE
70163  00047  53 20000 00050              ORDER
70164  00050  66 00000 00043   ADVANCE MT
70165  00051  64 00001 00340   READ MT
70166  00052  67 00001 00113   BACK MT
70167  00053  11 00322 00320   BLOCK NO.→C.B.N.
70170  00054  11 00254 20000   SET
70171  00055  11 00230 10000      TRANSFER OF
70172  00056  52 00330 00075         NEW WORD TO OLD
70173  00057  55 20000 00017   SET "OLD WORD
70174  00060  15 10000 00063     TO A" TRANSFER
70175  00061  11 00330 10000   SKIP
70176  00062  44 00075 00063   ┌COMPARISON?
70177  00063  11 30060 20000   NO, OLD WORD TO A
70200  00064  43 00331 00075   OLD WORD CORRECT?
70201  00065  11 20000 10000   NO.
70202  00066  31 00255 00052     ERROR
70203  00067  37 00074 00072        PRINT
70204  00070  11 00330 20000           AND
70205  00071  56 00000 00076            HALT
70206  00072  61 00000 20000   FLEX
70207  00073  34 20000 00006      PRINT
70210  00074  47 00072 30074         ROUTINE
70211  00075  11 00332 30056   NEW WORD TO OLD WORD
70212  00076  37 00076 00077
70213  00077  21 00034 00247
```

FX 7190C-8-109

By: R. M. Price

CV-109

MOCK VIII ST 001

```
70214  00100  41 00321 00033   USED ALL PARAMS.? NO
70215  00101  45 00000 00016   YES READ NEXT SET
70216  00102  65 00001 00340   WRITE FINAL BLOCK
70217  00103  11 00226 10000      SET
70220  00104  53 00320 00106         CURRENT BLK. NO. + 1
70221  00105  21 00106 00237         TO REWIND ORDER
70222  00106  67 00000 00114   REWIND MT
70223  00107  57 00000 00000   FINAL STOP
70224  00110  47 00053 00111   (BLOCK# EQUALS C.B.N.) DOES BLK# = (
70225  00111  37 00111 70154   YES        (9)   SWITCH
70226  00112  45 00000 00053                  (9B)
70227  00113  16 00050 00042   (9A)  SET WRITE. SET 9B
70230  00114  45 00000 00051   JUMP TO READ
70231  00115  31 00256 00052   ALARM. BLK. NO.
70232  00116  45 00000 00067      OUT OF SEQUENCE
70233  00117  00 70000 00000
70234  00120  00 00000 00 *
70235  00121  17 00000 00231   PICK READ CARD AND READ
70236  00122  75 30005 00124   CLEAR
70237  00123  23 00333 00333      TEMP. STG.
70240  00124  11 00232 00325
70241  00125  37 00135 00127   READ ROW 9
70242  00126  37 00135 00127   READ ROW 8
70243  00127  76 00000 00327         READ
70244  00130  76 10000 10000            ONE
70245  00131  76 10000 00326               ROW
70246  00132  55 10000 00003   TEST FOR
70247  00133  44 00240 00241     BIT IN
70250  00134  21 00333 10000        COLUMN FOUR
70251  00135  37 00135 00136
70252  00136  11 00233 00146
70253  00137  11 00234 00327   SET INDEX FOR  "ONE FIELD"
```

PX 719CC-8-109

By: R. M. Price

CV-109

MOCK VIII ST 001

70254	00140	31	00227	00014	SET SENTINEL
70255	00141	32	20000	00002	SHIFT LEFT: $A_R(3)$ $A_L(2)$
70256	00142	44	00143	00144	BIT IN THIS COL.? NO
70257	00143	32	00325	00000	YES: A_R + ROW NO.
70260	00144	46	00145	00141	SENTINEL?
70261	00145	31	20000	00000	YES REMOVE SENTINEL
70262	00146	35	30136	30320	$T_i + A_R \longrightarrow T$
70263	00147	21	00146	00235	
70264	00150	41	00327	00140	FINISHED ONE FIELD?
70265	00151	37	00151	00152	YES
70266	00152	55	00326	00000	$F_2 \longrightarrow Q$
70267	00153	37	00151	00140	SET JUMP
70270	00154	41	00325	00127	READ ALL ROWS? (THRU ROW 0)
70271	00155	37	00135	00127	YES: READ ROW 11
70272	00156	12	00334	20000	
70273	00157	42	00121	00206	FLIP OR NORMAL?
70274	00160	55	00334	10011	FLIP. EXTRACT
70275	00161	51	00223	20000	OLD
70276	00162	55	10000	00003	WORD
70277	00163	52	00224	20000	FROM
70300	00164	55	00335	10017	CARD
70301	00165	52	00225	00331	IMAGE
70302	00166	55	00336	10014	EXTRACT
70303	00167	51	00223	20000	NEW
70304	00170	55	10000	00003	WORD
70305	00171	52	00250	20000	FROM
70306	00172	55	00337	10017	CARD
70307	00173	52	00243	20000	IMAGE
70310	00174	55	10000	00003	
70311	00175	52	00225	00332	
70312	00176	55	00335	10014	
70313	00177	51	00226	20000	SET "BBBB"

PX 71900-8-109

By: R. M. Price

MOCK VIII ST 001

70314	00200	55	00336	10006	
70315	00201	52	00230	00330	SET "WW"
70316	00202	37	00135	00127	READ ROW 12
70317	00203	37	00203	00204	
70320	00204	21	00330	00227	SET NO "OLD WORD" INDICATOR
70321	00205	45	00000	30000	EXIT
70322	00206	55	00334	10011	NORMAL. EXTRACT
70323	00207	51	00227	00331	OLD
70324	00210	55	10000	00003	WORD
70325	00211	15	10000	00331	FROM
70326	00212	55	00335	10017	CARD
70327	00213	16	10000	00331	IMAGE
70330	00214	55	00336	10014	EXTRACT
70331	00215	51	00227	00332	NEW
70332	00216	55	10000	00003	WORD
70333	00217	15	10000	00332	FROM
70334	00220	55	00337	10022	CARD
70335	00221	16	10000	00332	IMAGE
70336	00222	45	00000	00176	
70337	00223	77	77000	00000	CONSTANTS
70340	00224	00	00777	70000	
70341	00225	00	00000	07777	
70342	00226	00	07777	00000	
70343	00227	77	00000	00000	
70344	00230	00	00000	00077	
70345	00231	00	00000	00105	
70346	00232	00	00000	00*07	
70347	00233	35	00334	00334	
70350	00234	00	00000	00002	
70351	00235	00	00001	00001	
70352	00236	00	00000	00001	
70353	00237	00	00001	00000	

FX 71900-8-109

MOCK VIII ST 001

70354	00240	16	00205	00203	
70355	00241	55	10000	00040	
70356	00242	45	00000	00134	
70357	00243	00	00000	70000	
70360	00244	00	00400	00044	
70361	00245	00	00000	00400	
70362	00246	00	00000	00003	
70363	00247	00	00003	00000	
70364	00250	00	00777	00000	
70365	00251	16	00023	00252	BLANK CARD, SET
70366	00252	11	00226	30251	SENTINEL
70367	00253	45	00000	00032	
70370	00254	11	00332	00340	
70371	00255	45	47311	22257	FLEX CODE FOR *WRD*
70372	00256	45	47231	13657	FLEX CODE FOR *BLK*
70373		00	00*		BLANK
70374		00	00*		CELLS
70375		75	30257	00001	PROGRAM
70376		11	70114	00000	TRANSFER TO ES
70377		45	00000	70375	ENTRY TO ROUTINE

15 December 1955

OPERATIONS RESEARCH OFFICE
Chevy Chase, Md.

Title:	Magnetic Tape Block Counter Routine
Format:	Standard Form
Storage:	
a) Total:	01000 - 01077, 100 octal
b) Instructions:	01000 - 01063, 64 octal
c) Constants:	01064 - 01077, 14 octal
d) Constant Pool:	40, 41, 73, 74
e) Temporary Pool:	11, 12
Alarm Conditions:	
a) Alarm 1:	The operation part of the MT order does not equal 64, 65, 66, or 67.
b) Alarm 2:	The j part of the MT order does not equal 0, 1, 2, or 3.
Coded & Machine Checked By:	W. Bruce Taylor

Description:

The block counter routine automatically keeps track of the position of the reading head of each of the magnetic tape units, provided a return jump to the block counter routine, - namely 37 01001 01002, - is written after each magnetic tape order.

The cumulative number of blocks which the reading head has advanced is stored in the registers whose addresses are a, a+1, a+2, and a+3, for the 0th, 1st, 2nd, and 3rd tape unit, respectively. Thus (a) is the block count of the 0th tape unit, where the concept of "block count" is further discussed in the following.

The addresses a thru a+3 are selected, and (a) thru (a+3) are set equal to zero by the following sequence of orders:

Order	Function of Order
MT j n v	One of the four magnetic tape orders.
37 01001 01002	Return jump to block counter routine.
00 a 00000	Parameter word for selection and clearing of counters.
- - - - - - - - -	Next order executed after block counter routine.

(Fig. 1. See next page)

PX 71900-8-110

Fig. 1. Sequence of orders which first selects a thru a+3 as the addresses of the registers in which the block counts will be stored, then clears these registers, then sets $(a+j) = n.2^{15}$.

The block counter routine selects and clears the counters by first testing whether the two left-most octal digits of the word immediately following the return jump are both equal to zero. If so, the selection and clearing takes place. However, selection and clearing does not take place otherwise, and in this latter case the order executed after the block counter routine is the order following (in address sequence) the return jump order. This use of the block counter routine is as follows:

Order	Function of Order
MT j n v	One of the four magnetic tape orders.
37 01001 01002	Return jump to block counter routine.
- - - - - - - - -	Next order executed after block counter routine.

Fig. 2. Sequence of orders which increases or decreases (a+j) by $n.2^{15}$, depending upon whether the order MTjnv advances or backs tape unit j.

The block counter routine is meant to be used first as in Fig. 1, and then for subsequent uses as in Fig. 2. Thus the counter addresses are chosen at their first usage, and retained thereafter.

Usually, a zero setting of the counters will correspond to a manual setting of the magnetic tape units each to the beginning of their first block. But this is not necessary, since the block counter routine can accumulate negative values of n if it is called for by the program which uses the routine. Thus a MT unit could be positioned in the middle of its reel so that forward positions from its starting position would be recorded as positive values, while backward positions would be recorded as negative values.

This routine has been found useful in a program in which the n value of the MT orders is a function of the prior course of the program.

Block Counter Routine

Storage Address	Order			Function of Order
01000	37	76000	76002	Alarm exit
01001	45	00000	30000	Normal exit
01002	23	00011	00011	Clear (11)
01003	16	01001	00011	y+2→(11), where (y) = MTjnv.
01004	55	00011	00017	$[y+2]\cdot 2^{15}$→(11).
01005	15	00011	01007	y+2→u (1007)
01006	55	00011	00025	y+2→(11)
01007	11	[30000]	00012	(y+2)→(12); written by (1005)
01010	11	01064	10000	op extractor to (Q)
01011	51	00012	20000	0→(L); op(y+2)$\cdot 2^{30}$→(R)
01012	43	00040	01055	If (A) = 0, jump to extract a, where a = address of block count of 0th tape unit.
01013	23	00011	00041	y→(11)
01014	55	00011	00017	y$\cdot 2^{15}$→(11)
01015	15	00011	01016	y→u(1016)
01016	11	[30000]	00011	MTjnv→(11); written by (1015)
01017	11	01064	10000	op extractor to (Q).
01020	51	00011	20000	0→(L); op(y)$\cdot 2^{30}$→(R)
01021	11	20000	20000	op(y)$\cdot 2^{30}$→(A)
01022	43	01065	01030	op(y) = BM ? Yes, to 1030
01023	43	01066	01032	op(y) = AM ? Yes, to 1032
01024	43	01067	01032	op(y) = WM ? Yes, to 1032
01025	43	01070	01032	op(y) = RM ? Yes, to 1032
01026	11	00074	20000	1→(A)
01027	45	00000	01000	1-alarm, no MT order detected.

PX 71900-8-110

Storage Address	Order			Function of Order
01030	11	01071	01053	Prepare to decrease block-count.
01031	45	00000	01033	Jump over increase-case.
01032	11	01072	01053	Prepare to increase block-count
01033	11	01073	10000	Put j extractor in (Q).
01034	51	00011	20000	$j(y)\cdot 2^{27} \rightarrow (A)$.
01035	43	00040	01051	j = 0 ? Yes, to 1051. No, continue.
01036	21	01053	00073	$a+1 \rightarrow u(1053)$.
01037	51	00011	20000	$j(y)\cdot 2^{27} \rightarrow (A)$.
01040	43	01074	01051	j = 1 ? Yes, to 1051. No, continue.
01041	21	01053	00073	$a+2 \rightarrow u(1053)$.
01042	51	00011	20000	$j(y)\cdot 2^{27} \rightarrow (A)$.
01043	43	01075	01051	j = 2 ? Yes, to 1051. No, continue.
01044	21	01053	00073	$a+3 \rightarrow u(1053)$.
01045	51	00011	20000	$j(y)\cdot 2^{27} \rightarrow (A)$.
01046	43	01076	01051	j = 3 ? Yes, to 1051. No, continue.
01047	11	00041	20000	$2 \rightarrow (A)$.
01050	45	00000	01000	2-alarm, no proper j-value detected.
01051	11	01077	10000	Put n-extractor in (Q).
01052	51	00011	00012	$n(y)\cdot 2^{15} \rightarrow (12)$.
01053	[00	30000	30000]	Modify block-count; written by 1030, 1032, 1036, 1041, 1044.
01054	45	00000	01001	Jump to normal exit.
01055	15	00012	01071	Set u(1071) = a
01056	15	00012	01072	Set u(1072) = a
01057	55	00012	00025	$a \rightarrow (12)$.

PX 71900-8-110

Storage Address	Order			Function of Order
01060	16	00012	01063	a→ v(1063).
01061	21	01001	00074	Reset exit to skip (y+2).
01062	75	10004	01013	Clear (a), (a+1), (a+2), & (a+3) & return
01063	11	00040	[30000]	to continue MT block-count routine.
01064	77	00000	00000	op extractor.
01065	67	00000	00000	op (BMjn-).
01066	66	00000	00000	op (AMjn-).
01067	65	00000	00000	op (WMjnv).
01070	64	00000	00000	op (RMjnv).
01071	23	[30000]	00012	Order which decreases block-count.
01072	21	[30000]	00012	Order which increases block-count.
01073	00	70000	00000	j extractor.
01074	00	10000	00000	$1 \cdot 2^{27}$.
01075	00	20000	00000	$2 \cdot 2^{27}$.
01076	00	30000	00000	$3 \cdot 2^{27}$.
01077	00	07777	00000	n extractor.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Fixed Point Definite Integral Evaluation

Specifications

Identification Tag:	DIE-1
Type:	Subroutine
Assembly Routine Spec:	SUB 51236 03506
Storage:	29 instructions, addresses L00 thru L28
	6 constants in program, addresses C01 thru C06
	35 words total program storage, addresses L00 thru L28, C01 thru C05
	5 words temporary storage pool used, addresses 00027b thru 00033b
	The constant pool is used by this routine.
Program Entrances:	Addresses L02, L03, L04
Program Exit:	Address L01
Alarm Exit:	The alarm exit is not used by this routine.
Drum Assignment:	Addresses 65744b thru 66006b
Machine Time:	(4.432 + .032n) milliseconds for Entrance #1 (4.184 + .032n) " for Entrance #2 (4.416 + .032n) " for Entrance #3
Mode of Operation:	Fixed point

Coded by:	C. Miller	November, 1955
Code Checked by:	C. Miller	November, 1955
Machine Checked by:	C. Miller	November, 1955
Approved by:	W. F. Bauer	November 25, 1955

Descr_ption

This routine computes an approximation to the integral mean

$$\frac{1}{(x_n - x_o)} \int_{x_o}^{x_n} y dt$$

for given tabular data $y_i = y(x_i)$, $(i = 0,\ldots,n)$ stored at consecutive cells and scaled at 2^s. The arguments are equally spaced and given by

$$x_i = x_o + ih \qquad (i = 0,\ldots,n)$$

The routine requires the address where $y_o \cdot 2^s$ is stored and either the address of $y_n \cdot 2^s$ or the value of n. The result is left in both A and Q at the end of the routine and scaled by the same factor as were the y's. The routine does not need to have cognizance of this scaling.

The difference between the routine DIE-0 and this routine is that the error terms are of fourth and fifth order respectively.

Programming Instructions

Three entrances are available depending on which combinations of the parameters are given. We define a_i as the address where $y_i \cdot 2^s$ is stored. See below for limitations on the parameters.

1. Entrance #1
   a. Place a_o in Q and n (scaled at 2^0) in A.
   b. Execute RJ 00L01 00L02 if the subroutine begins at cell 00L00.
   c. Control is returned to the word following the RJ instruction.
2. Entrance #2
   a. Place in A the double extension of the parameter word

      oo uuuuu vvvvv

      where uuuuu = a_o and vvvvv = a_n
   b. Execute RJ 00L01 00L03 if the subroutine begins at cell 00L00.
   c. Control is returned to the word following the RJ instruction.

PX 71900-8-111

3. Entrance #3

   a. Enter with

      RJ 00L01 00L04

      00 uuuuu vvvvv

   assuming that the subroutine begins at cell 00L00 and where uuuuu = a_o and vvvvv = a_n.

   b. Control is returned to the cell following the parameter word.

For all three cases the calculation is identical, with the result (scaled 2^8) left in A and Q upon exit from the subroutine.

Range of Parameters

The routine requires that $8 \leq n \leq 4095$ and that a_o and a_n both be ES addresses or both be MD addresses. A check is not made which would enforce these requirements. For n too small, an incorrect computation will be produced. For inadmissible addresses (a_o and a_n), n too large or operation code non zero in the parameter word, the routine will either produce an incorrect answer or halt on an overflow fault or an SCC fault.

Execution Time

Assuming the data to be in ES, the execution times are, in milliseconds:

4.432 + .032n	(Entrance #1)
4.184 + .032n	(Entrance #2)
4.416 + .032n	(Entrance #3)

If, however, the data are in MD one must add to these figures the time for five and a fraction drum revolutions assuming a 4 interlace. This is an amount of time varying from 170 to 204 milliseconds.

Mathematical Analysis

According to techniques of Milne*, quadrature formulas were derived as follows:

$$\int_{x_o}^{x_1} y\,dx = \frac{h}{24}\left[9\,y_o + 19\,y_1 - 5\,y_2 + y_3\right] - \frac{19}{720}\,y^{(4)}\,h^5 \qquad (1)$$

and the dual

$$\int_{x_{n-1}}^{x_n} y\,dx = \frac{h}{24}\left[9\,y_n + 19\,y_{n-1} - 5\,y_{n-2} + y_{n-3}\right] - \frac{19}{720}\,y^{(4)}\,h^5 \qquad (1')$$

These expressions were substituted in the following identity for the integration over the intervals $[x_o, x_1]$ and $[x_{n-1}, x_n]$.

$$\int_{x_o}^{x_n} = \frac{1}{4}\left[2\int_{x_o}^{x_1} + \int_{x_o}^{x_2} + \int_{x_1}^{x_3} + \sum_{i=2}^{n-2}\int_{x_{i-2}}^{x_{i+2}} + \int_{x_{n-3}}^{x_{n-1}} + \int_{x_{n-2}}^{x_n} + 2\int_{x_{n-1}}^{x_n}\right] \qquad (2)$$

In the remaining integrals use was made of Milne's formula (4), page 123 for the integration over the interval $[x_{i-2}, x_{i+2}]$ while Simpson's rule was employed for integration over the remaining integrals. This gave rise to the quadrature formula

$$\int_{x_o}^{x_n} y\,dx = h \sum_{i=0}^{n} \delta_i \; y_i \doteq E \qquad (3)$$

with

$\delta_o = 13/48 + 7/90 = .3486\ 111111$

$\delta_1 = 39/48 + 39/90 = 1.2\ 458\ 333333$

$\delta_2 = 15/48 + 51/90 = .8791666667$

$\delta_3 = 5/48 + 83/90 = 1.0263\ 888889$

$\delta_i = 1 \qquad\qquad 4 \leq i \leq n-4$

$\delta_{n-i} = \delta_i$

*Milne, W. E., Numerical Calculus, Princeton University Press, Princeton, New Jersey, 1949, Chapter IV.

and E is the error term

$$E = \frac{3}{80} y^{(4)} h^5 + (n-4) \left(\frac{2}{945}\right) y^{(6)} h^7$$

$$E = \frac{3}{80n^5} y^{(4)} L^5 + \left(\frac{n-4}{n^7}\right) \left(\frac{2}{945}\right) y^{(6)} L^7$$

where $L = nh = x_n - x_o$ is the length of the interval of integration. Dividing formula (3) by $L = nh = x_n - x_o$ gives the approximation to the integral mean:

$$M = \frac{1}{n} \sum_{i=0}^{n} \delta_i y_i$$

The routine calculates

$$M \cdot 2^s = \frac{1}{n} \sum_{i=0}^{n} \delta_i y_i 2^s$$

Error Analysis

The truncation error in the quadrature formula used is as indicated in the analysis above. The total round-off error in the routine's calculation of $M \cdot 2^s$ does not exceed $1 + 4/n$.

That is,

$$\left| (A)_f - M \cdot 2^s \right| \angle 1 + 4/n.$$

PX 71900-8-111

```
D                   RWC00  00000                   RW CONSTANTS             00  00000  00000
D                   RWT00  00023                   RW TEMPS            27   00  00000  00000
D                   00L00  01024                                     2000   00  00000  00000
D                   00C00  01052                                     2034   00  00000  00000
D                   0SL00  51236                                    65744   00  00000  00000
D                   0SC00  51264                                    66000   00  00000  00000
  SL00     DV       RWT00  A0000                   ANS TO Q ACC     65744   73  00027  20000
  SL01     MJ       00000  A0000                   EXIT             65745   45  00000  20000
  SL02     MA       Q0000  RWC17                   ENT NO 1         65746   72  10000  00021
  SL03     MJ       00000  00L07                   ENT NO 2         65747   45  00000  02007
  SL04     TV       00L01  00L06                   ENT NO 3         65750   16  02001  02006
  SL05     RA       00L01  RWC16                    SET UP EXIT     65751   21  02001  00020
  SL06     MP       RWC16  00000                   PAR TO ACC       65752   71  00020  00000
  SL07     TU       A0000  00L28                                    65753   15  20000  02034
  SL08     TP       A0000  RWT00                   PAR TO RWT00     65754   11  20000  00027
  SL09     LQ       RWT00  10015                                    65755   55  00027  10017
  SL10     TU       Q0000  00L19                   STASH ADD YN     65756   15  10000  02023
  SL11     LQ       Q0000  00006                                    65757   55  10000  00006
  SL12     TV       Q0000  RWT00                                    65760   16  10000  00027
  SL13     TV       Q0000  00L25                   STASH ADD Y0     65761   16  10000  02031
  SL14     ST       RWT00  RWT00                   N TO RWT00       65762   36  00027  00027
  SL15     SA       00L28  00015                   SET UP           65763   32  02034  00017
  SL16     TU       A0000  00L27                       RP COUNT     65764   15  20000  02033
  SL17     RS       00L19  00C01                   BLOCK TRF        65765   23  02023  02035
  SL18     RP       30004  00L20                    YN MINUS 3      65766   75  30004  02024
  SL19     TP       00000  RWT01                      TO RWT01      65767   11  00000  00030
  SL20     MP       00C02  RWT04                                    65770   71  02036  00033
  SL21     MA       00C04  RWT03                                    65771   72  02040  00032
  SL22     MA       00C05  RWT02                                    65772   72  02041  00031
  SL23     MA       00C06  RWT01                                    65773   72  02042  00030
  SL24     RP       30004  00L26                                    65774   75  30004  02032
  SL25     MA       00C03  00000                                    65775   72  02037  00000
  SL26     TP       B0000  A0000                                    65776   11  30000  20000
  SL27     RP       00000  00L00                                    65777   75  00000  02000
  SL28     AT       00000  A0000                                    66000   35  00000  20000
  SC01     00       00003  00000                                    66001   00  00003  00000
  SC02     03       48611  11111-0  1    35  D0  PLUS 1             66002   13  11751  17512
  SC03-    06       51388  88889-0  1    35  D0                     66003   53  11751  17511
  SC04     02       45833  33333-0  1    35  D1                     66004   07  67356  73567
  SC05-    01       20833  33333-0  1    35  D2                     66005   74  10421  04210
  SC06     02       63888  88889-0  2    35  D3                     66006   00  66026  60266
```

PX 71900-8-111

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Floating Point Definite Integral Evaluation

Specifications

Identification Tag:	DIE-2
Type:	Subroutine
Assembly Routine Spec:	SUB 51271 04504
Storage:	41 instructions, addresses L00 thru L01 H00 thru H38 4 constants in program, addresses L00 thru L03 45 words total program storage, addresses L00 thru L01 H00 thru H38 C00 thru C03 1 word temporary storage pool used, address 00027b The constant pool is used by this routine.
Program Entrances:	Addresses L02, L03, L04
Program Exit:	Address L01
Alarm Exit:	The alarm exit is not used by this routine.
Drum Assignment:	Address 66007b thru 66063b
Machine Time:	(8.74 + 1.55n) milliseconds for Entrance #1 (8.49 + 1.55n) " for Entrance #2 (8.72 + 1.55n) " for Entrance #3
Mode of Operation:	Floating point, requiring SNAP to be in memory.

Coded by:	C. Miller	November, 1955
Code Checked by:	C. Miller	November, 1955
Machine Checked by:	C. Miller	November, 1955
Approved by:	W. F. Bauer	November 25, 1955

PX 71900-8-112

Description

This routine is a floating point version of DIE-1 evaluating the integral mean

$$\frac{1}{x_n - x_o} \int_{x_o}^{x_n} y \, dt$$

The data must be presented in floating point and the computation will place a floating point result in both the A and Q registers at the end of the routine. This routine requires that SNAP be in the memory.

Programming Instructions

There are three possible entries to this routine. These are identical to those used in DIE-1.

Range of Parameters

See DIE-1

Execution Time

Assuming the data to be in ES, the average execution times are, in milliseconds:

8.74 + 1.55n	(Entrance #1)
8.49 + 1.55n	(Entrance #2)
8.72 + 1.55n	(Entrance #3)

If, however, the data are in MD the time is n plus a fraction drum revolution, or an amount of time varying from 34n to 34 (n + 1) milliseconds.

Mathematical Analysis

See DIE-1.

For coding convenience, this routine computes M as follows:

$$M = \frac{\alpha_3 \left\{ \alpha_2 \left[\alpha_1 \ (\alpha_o z_o + z_1) \right] + z_2 \right\} + z_3 + \sum_{i=4}^{n-4} y_i}{n}$$

PX 71900-8-112

with $\alpha_i = \delta_i \; \delta_{i+1}^{-1}$

and $z_i = y_i + y_{n-i}$

This formulation is equivalent to the M defined in DIE-1.

Error Analysis

The truncation error in the quadrature formula is described in DIE-1. The round-off error in this routine is approximately equal to $- \lambda/h \; Y(\xi)$ where λ is a pseudo random variable in the interval $0 \leq \lambda < 2^{-26}$, h is the step size ($h = x_{i+1} - x_i$), and

$$Y(x) = \int_{x_o}^{x} y(t)\, dt$$

while ξ is some point in the interval $[x_o, x_n]$

If M^* is the machine approximation to M then

$$M^* = M - \lambda/h \; Y(\xi)$$

Since M is the discrete approximation to the integral mean

$$\frac{1}{x_n - x_o} \int_{x_o}^{x_n} y \, dx ,$$

it is necessary to multiply M by the interval $L = nh = x_n - x_o$ in order to obtain the integral of y. This gives the relation

$$L M^* = L M - n \lambda Y(\xi)$$

and hence the error in the integral due to round-off within the routine is

$$-n \lambda Y(\xi)$$

PX 71900-8-112

```
D              RWC00  00000               RW  CONSTANTS          00  00000  00000
D              RWT00  00023               RW  TEMPS          27  00  00000  00000
D              00L00  01024                                2000  00  00000  00000
D              00H00  01026                                2002  00  00000  00000
D              00C00  01065                                2051  00  00000  00000
D              0SL00  51271                               66007  00  00000  00000
D              0SH00  51273                               66011  00  00000  00000
D              0SC00  51312                               66060  00  00000  00000
 SL00    TP    00C04  Q0000               DUMMY           66007  11  02055  10000
 SL01    MJ    00000  B0000               EXIT            66010  45  00000  30000
 SH00    MA    Q0000  RWC17               ENT NO 1        66011  72  10000  00021
 SH01    MJ    00000  00H05               ENT NO 2        66012  45  00000  02007
 SH02    TV    00L01  00H04               ENT NO 3        66013  16  02001  02006
 SH03    RA    00L01  RWC16                SET UP EXIT    66014  21  02001  00020
 SH04    MP    RWC16  00000               PAR TO ACC      66015  71  00020  00000
 SH05    TP    00002  RWT00               SAVE  F         66016  11  00002  00027
 SH06    TU    A0000  00H16                SET IN Y0      66017  15  20000  02022
 SH07    LA    A0000  00015                               66020  54  20000  00017
 SH08    TU    A0000  00H18                SET IN YN      66021  15  20000  02024
 SH09    TP    00H18  A0000                               66022  11  02024  20000
 SH10    ST    00H16  00002                N GOES TO F    66023  36  02022  00002
 SH11  FLN0    00015  00000                   FLOATING    66024  14  44017  00000
 SH12    11    00H01  01362  BRB          BLOCK SNAP      66025  11  02003  01362
 SH13    23    01325  20000  B            CLEAR           66026  23  01325  20000
 SH14    23    01326  20000  B                E           66027  23  01326  20000
 SH15    TU    00H38  00H20               SET IN ALPHA    66030  15  02050  02026
 SH16    TP    00000  Q0000               E PLUS Y0       66031  11  00000  10000
 SH17    37    01362  01416  B               GOES TO E    66032  37  01362  01416
 SH18    TP    00000  Q0000               E PLUS YN       66033  11  00000  10000
 SH19    37    01362  01416  B                    TO E    66034  37  01362  01416
 SH20    TP    00000  Q0000               E TIMES ALPH    66035  11  00000  10000
 SH21    37    01362  01457  B                    TO E    66036  37  01362  01457
 SH22    RA    00H16  RWC15                               66037  21  02022  00017
 SH23    RS    00H18  RWC15                               66040  23  02024  00017
 SH24    RA    00H20  RWC15                               66041  21  02026  00017
 SH25    TJ    00L00  00H16                               66042  42  02000  02022
 SH26    RA    00H18  RWC15               USE H18DUMMY    66043  21  02024  00017
 SH27    TP    00H16  00H28                               66044  11  02022  02036
 SH28    00    0                          E PLUS YI       66045  00  00000  00000
 SH29    37    01362  01416  B                  TO E      66046  37  01362  01416
 SH30    RA    00H28  RWC15                               66047  21  02036  00017
 SH31    TJ    00H18  00H28                               66050  42  02024  02036
 SH32    TP    00002  Q0000                DIVIDE E       66051  11  00002  10000
 SH33    37    01362  01507  B                 BY N       66052  37  01362  01507
 SH34    27    01325  01326  B            PACK            66053  27  01325  01326
 SH35    LQ    A0000  000                 ANS TO Q ACC    66054  55  20000  00000
 SH36    11    01557  01362  B            REVIVE SNAP     66055  11  01557  01362
 SH37    TP    RWT00  00002               RESTORE F       66056  11  00027  00002
 SH38    45    00C00  00L01  BRR                          66057  45  02051  02001
 SC00    02    79821  62760-0  1  F         A      0      66060  17  74364  23076
 SC01    01    41706  16114    0  F          L     1      66061  20  15526  11062
 SC02    08    56562  92290-0  1  F           P    2      66062  20  06664  36651
 SC03    01    02638  88889    0  F            H   3      66063  20  14066  02660
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Fixed Point Natural Logarithm

Specifications

Identification Tag:	LOG-1
Type:	Subroutine
Assembly Routine Spec:	SUB 51201 03517
Storage:	18 instructions, addresses M00 thru M17
	17 constants in program, addresses C00 thru C16
	35 words total program storage, addresses M00 thru M17, C00 thru C16
	2 words temporary storage pool used, addresses 00027b thru 00030b
	The constant pool is used by this routine.
Program Entrance:	Address M02
Program Exit:	Address M01
Alarm Exit:	The alarm exit is used by this routine.
Drum Assignment:	Address 65701b thru 65743b
Machine Time:	2.63 milliseconds (minimum) 4.51 " (average) 6.39 " (maximum)
Mode of Operation:	Fixed point

Coded by:	C. Miller	November, 1955
Code Checked by:	C. Miller	November, 1955
Machine Checked by:	C. Miller	November, 1955
Approved by:	W. F. Bauer	November 22, 1955

Description

This subroutine computes a single precision approximation to ln x for any positive x which can be accommodated in the double length accumulator scaled at 2^{35}. $(0 < x < 2^{35})$

The result is left in the accumulator, at the end of the routine, scaled at 2^{35}. Since

$$|\ln x| < \ln 2^{35} = 24.2$$

we see that the result may well extend into A_L, but will never overflow A.

Programming Instructions

1. Place $x \cdot 2^{35}$ in A $\quad (0 < x < 2^{35})$
2. Enter the subroutine with RJ 00M01 00M02 (assuming the first word of this subroutine is stored at 00M00).
3. Control is returned to cell p + 1 if the RJ order is contained in cell p. The accumulator will contain $(\ln x) \cdot 2^{35}$.

Alarm Conditions

1. If $x \leq 0$ then the alarm routine AIR-1 is entered. This initiates the printing of the word "alarm" on the flexowriter, followed by the octal address of the RJ instruction used to enter LOG-1 and the contents of A and Q.
2. If $x \geq 2^{35}$ then a multiply-add overflow may occur in line 00M05 of the subroutine. If the overflow does not occur (and therefore does not halt the machine) then the answer produced is correct to within the error described below.

Mathematical Method

Assuming that x is strictly positive, i.e. $x \cdot 2^{35} \geq 1$, then the routine scale factors x in order to obtain

$$x = 2^s u$$

where s is integral and $\frac{1}{2} \leq u < 1$.

Setting $v = 4/3\, u - 1$

one obtains $x = 2^s u = 2^s (3/4)\ (v + 1)$

with $-1/3 \leq v < 1/3$

Hence $\ln x = s(\ln 2) + \ln 3/4 + \ln (v + 1)$

The routine evaluates an eleventh order polynomial approximation to ln (v + 1) over the interval $-1/3 \leq v < 1/3$. This polynomial was obtained with the aid of CVF-0 and has a maximum error of 3.85×2^{-35}. The twelve coefficients of this polynomial are included in the accompanying listing. It is to be noted that the first and second coefficients are modified by the addition of ln 3/4 and -1 respectively.

Error Analysis

The error in the routine's approximation to ln x will not exceed, in absolute value

$$(.72 \cdot \left| \ln x \right| + 8.60) \cdot 2^{-35}.$$

That is

$$\left| (A)_f - (\ln x) \cdot 2^{35} \right| < .72 \cdot \left| \ln x \right| + 8.60$$

Most of the error is due to round-off within the routine. The actual error is usually less than the upper bound stated here.

```
D               RWC00  00000                  RW  CONSTANTS              00  00000  00000
D               RWT00  00023                  RW  TEMPS           27     00  00000  00000
D               00M00  01024                                    2000     00  00000  00000
D               00C00  01042                                    2022     00  00000  00000
D               0SM00  51201                                   65701     00  00000  00000
D               0SC00  51219                                   65723     00  00000  000 0
  SM00     37   75701  75703  B                  ALARM         65701     37  75701  757 3
  SM01     MJ   00000  A0000                     EXIT          65702     45  00000  20000
  SM02     TJ   RWC16  00M00                  OK  IF  X  POS   65703     42  00020  02000
  SM03     SF   A0000  00C00                    K  TO  C00     65704     74  20000  02022
  SM04     SS   00C01  00035                                   65705     34  02023  00043
  SM05     MA   B0000  00C02                  V  TIMES  2  TO  65706     72  30000  02034
  SM06     TP   B0000  RWT00                    35  TO  RWT0   65707     11  30000  00027
  SM07     LA   00C05  20036                  H  OF  V  TIMES  65710     54  02027  20044
  SM08     RP   20011  00M10                    2  TO  35      65711     75  20013  02012
  SM09     PM   00C06  RWT00                       TO  RWT01   65712     24  02030  00027
  SM10     TP   B0000  RWT01                                   65713     11  30000  00030
  SM11     TP   00C00  A0000                  S  IS  K  OR  72 65714     11  02022  20000
  SM12     TJ   00C03  00M14                     MINUS  K      65715     42  02025  02016
  SM13     ST   RWC18  A0000                                   65716     36  00022  20000
  SM14     MP   A0000  00C04                                   65717     71  20000  02026
  SM15     AT   RWT00  A0000                                   65720     35  00027  20000
  SM16     AT   RWT01  A0000                  ANS  IN  A    35 65721     35  00030  20000
  SM17     MJ   00000  00M01                                   65722     45  00000  02001
  SC00                                        PRIVATE  TEMP    65723     00  00000  00000
  SC01     30   00000  00000  B               3/4THS        35 65724     30  00000  00000
  SC02     12   52525  25253  B               1/3RD         35 65725     12  52525  25253
  SC03     00   00000  00  37                                  65726     00  00000  00045
  SC04     06   93147  18056-0  1     35        LN  2          65727     26  13441  37676
  SC05     01   28251  84968-0  1     34      COEF  OF      11 65730     02  03250  71760
  SC06-    C1   33649  76676-0  1     34        11TH  ORD10    65731     75  67111  01772
  SC07     01   05077  09477-0  1     34        APPROX       9 65732     01  53462  52166
  SC08-    01   20530  66482-0  1     34        TO  LN       8 65733     76  04447  16001
  SC09     01   43313  41362-0  1     34        OF  1  PLUS7   65734     02  22601  40142
  SC10-    01   66932  02409-0  1     34        V  OVER      6 65735     75  25037  42576
  SC11     01   99983  25744-0  1     34        NEG  1/3RD5    65736     03  14620  64472
  SC12-    02   49992  96484-0  1     34              TO     4 65737     74  00003  54036
  SC13     03   33333  59896-0  1     34        POS  1/3RD3    65740     05  25252  63451
  SC14-    05   00000  06781-0  1     34       MINUS  V      2 65741     67  77777  75562
  SC15-    01   24000  00000-0  9     34                     U 65742     77  77777  77752
  SC16-    02   87682  07245-0  1     34      LN  3/4THS     0 65743     73  31323  57354
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Fixed Point Exponential Routine (e^x)

Specifications

Identification Tag:	EXP-2
Type:	Subroutine
Assembly Routine Spec:	SUB 50916 03713
Storage:	24 instructions, addresses LL00 thru LL04 L04 thru L22
	13 constants in program, addresses C00 thru C12
	37 words total program storage, addresses LL00 thru LL04 L04 thru L22 C00 thru C12
	2 words temporary storage pool used, addresses 00027b thru 00030b
	The constant pool is used by this routine.
Program Entrance:	Address LL02
Program Exit:	Address LL01
Alarm Exit:	The alarm exit is used by this routine.
Drum Assignment:	Address 50916b thru 50952b
Machine Time:	2.3 milliseconds (minimum) 3.3 " (average) 4.3 " (maximum)
Mode of Operation:	Fixed point

Coded by:	C. Miller	October, 1955
Code Checked by:	C. Miller	October, 1955
Machine Checked by:	C. Miller	October, 1955
Approved by:	W. F. Bauer	November 15, 1955

Description

This subroutine computes e^x when entered with $x \cdot 2^{35}$ in the accumulator where x has the domain

$$-(\ln 2)\ (2^{35} + 2^{-1}) < x < 34.5\ (\ln 2)$$

The result is left in the accumulator with scaling 2^{35}, assuming that the capacity of the double length accumulator is not exceeded.

The routine employs the Polynomial Multiply (PM) instruction.

Programming Instructions

1. Place $x \cdot 2^{35}$ in A.
2. Enter the routine with RJ 00K01 00K02, where 00K00 is the location of the first word of the subroutine.
3. The subroutine returns control to the cell following the RJ instruction with $e^x \cdot 2^{35}$ in A.

Alarm Conditions

1. If x falls in the interval

   $$34.5\ (\ln 2) < x < (\ln 2)\ (2^{35} - 2^{-1})$$

   then the alarm routine ALR-1 is entered. This initiates the printing of the word "alarm" on the flexowriter, followed by the octal address of the RJ instruction used to enter EXP-2.

   This alarm condition is equivalent to

   $$e^x \geq 2^{34.5}$$

   and, hence, in terms of the scaled result

   $$e^x \cdot 2^{35} \geq 2^{69.5}$$

   This value will nearly overflow in A and therefore becomes an upper limit.
2. For $x \leq -(\ln 2)\ (2^{35} + 2^{-1})$

   or $x \geq (\ln 2)\ (2^{35} - 2^{-1})$

   a divide overflow will occur at cell 00K03 of the subroutine.

3. If ALR-1 was entered (or at the end of the routine) one can obtain the quantities

$$z \cdot 2^{35} = (00030b)$$

$$s = (00027b)$$

where $\frac{1}{2}\sqrt{\frac{1}{2}} \leq z < \sqrt{\frac{1}{2}}$

and $e^x \cong 2^s z$

This is not true in the event of a divide fault.

Mathematical Method

The routine finds q, an integer such that

$$x = q\,(\ln 2) + r$$

where $|r| \leq \frac{\ln 2}{2}$

This gives $e^x = (e^{\ln 2})^q \cdot e^r = 2^q \cdot e^r = (2^{q+1}) \cdot (\frac{e^r}{2})$

Since the factor 2^{q+1} is easily applied by shifting, it is only necessary to calculate the quantity $e^r/2$. This is accomplished by a 7th order approximating polynomial where the domain of r is

$$-\frac{\ln 2}{2} \leq r \leq \frac{\ln 2}{2}$$

This polynomial was obtained with the aid of routine CVF-0. The coefficients of the polynomial are listed in the accompanying code listing. The maximum discrepancy between the function $e^r/2$ and the polynomial, in the interval stated above, is $.75 \times 2^{-35}$.

Error Analysis

The error in the machine's approximation to e^x is bounded in all cases by

$$\left[(11.3 + .7\;|x|\;)\,e^x + 1\right] \cdot 2^{-35}$$

That is

$$\left|(A)_f - e^x \cdot 2^{35}\right| < (11.3 + .7\;|x|\;)\,e^x + 1$$

Most of the error is due to round-off within the routine. The actual error is usually less than the bound stated here.

```
D              RWC00  00000                 RW  CONSTANTS               00  00000  00000
D              RWT00  00023                 RW  TEMPS            27     00  00000  00000
D              0LL00  01024                                    2000     00  00000  00000
D              00L00  01025                                    2001     00  00000  00000
D              00C00  01048                                    2030     00  00000  00000
D              SLL00  50916                                   65244     00  00000  00000
D              0SL00  50917                                   65245     00  00000  00000
D              0SC00  50940                                   65274     00  00000  00000
SLL00     37   75701  75702  B                 ALARM          65244     37  75701  75702
SLL01     MJ   00000  A0000                    EXIT           65245     45  00000  20000
SLL02     AT   00C12  A0000                                   65246     35  02044  20000
SLL03     DV   00C00  RWT00                 Q  TO  RWT00      65247     73  02030  00027
SLL04     ST   00C12  RWT01                 R  TO  RWT01      65250     36  02044  00030
  SL04    LA   00C04  20036                                   65251     54  02034  20044
  SL05    RP   20007  00L07                                   65252     75  20007  02010
  SL06    PM   00C05  RWT01                                   65253     24  02035  00030
  SL07    TP   B0000  RWT01                 EXP  R  TO  T01   65254     11  30000  00030
  SL08    RA   RWT00  RWC16                 S  IS  Q  PLUS1   65255     21  00027  00020
  SL09    SJ   00L10  00L17                                   65256     46  02013  02022
  SL10    TV   00C02  00L14                                   65257     16  02032  02017
  SL11    AT   00C01  A0000                   S  NEG          65260     35  02031  20000
  SL12    TJ   00C02  00L14                 TEST  FORZERO     65261     42  02032  02017
  SL13    TV   A0000  00L14                                   65262     16  20000  02017
  SL14    LA   RWT01  00000                 SHIFT  RIGHT      65263     54  00030  00000
  SL15    TP   A0000  A0000                                   65264     11  20000  20000
  SL16    MJ   00000  0LL01                                   65265     45  00000  02001
  SL17    AT   00C03  A0000                   S  POS          65266     35  02033  20000
  SL18    TJ   00C02  00L20                 EMMINENT          65267     42  02032  02025
  SL19    MJ   00000  0LL00                                   65270     45  00000  02000
  SL20    TV   A0000  00L21                                   65271     16  20000  02026
  SL21    LA   RWT01  00000                 SHIFT  LEFT.      65272     54  00030  00000
  SL22    MJ   00000  0LL01                                   65273     45  00000  02001
  SC00    06   93147  18056-0  1      35    NATLOG  OF  2     65274     26  13441  37676
  SC01    00   00000  20110  B              CON               65275     00  00000  20110
  SC02    00   00000  20044  B                 STA            65276     00  00000  20044
  SC03    00   00000  20000  B                     NTS        65277     00  00000  20000
  SC04    01   99243  65600-0  4      33                      65300     00  00064  16602
  SC05    01   39485  76760-0  3      33    C   S   B     34  65301     00  00555  51650
  SC06    08   33324  84740-0  3      33     0   C   Y        65302     00  04210  40553
  SC07    04   16662  18354-0  2      33      E   A           65303     00  25252  43112
  SC08    01   66666  66994-0  1      33       F   L          65304     01  25252  52561
  SC09    05   00000  01077-0  1      33        F   E         65305     04  00000  00135
  SC10    09   99999  99997-0  1      33               D      65306     10  00000  00000
  SC11    09   99999  99996-0  1      33                      65307     10  00000  00000
  SC12    06   93147  18056-0  1      34    HALF  LN  OF  2   65310     13  05620  57737
```

PX 71900-8-114

ALARM, OCTAL AND FLEXPRINT PACKAGE IE 002

(revised 12- 27- 55)

The Alarm, Octal and Flexprint Package consists of an alarm routine, an octal print subroutine, and a flexprint subroutine.

The alarm routine is used to wave a flag whenever a test shows that an undesired event has occurred. There are two entries to the alarm routine for this purpose...

(1) 37 76000 76001 (main routine)

(2) 37 76000 76002 (subroutine)

Entry (1) is used to indicate an alarm condition in the main routine, entry (2) is used for an alarm condition in a standard coded subroutine. In either case the main routine address is printed out.

Entry to the alarm routine causes the following information to be printed out...

LLLLLLLLLLLL RRRRRRRRRRRR alarm xxxxx

LLLLLLLLLLLL is (L) in octal

RRRRRRRRRRRR is (R) in octal

xxxxx is the main routine address.

Registers A and Q and location 00000 are used for temporary storage. After an alarm, location 00000 contains 45 00000 xxxxx+1 . After the printout, the computer stops with the instruction 56 00000 00000 set up. To continue from line xxxxx+1, push the start button.

The location of the alarm routine is such that references to it are not modified by the assembly modification routine (SC 001). Thus all subroutines may refer directly to the entry and exit of the alarm routine.

This revised alarm routine does not use the constant pool or the temporary pool.

Flexcodes for the octal digits 0 - 7 are stored in locations 76037 - 76046 respectively.

The alarm routine resets itself completely-- its sum remains constant with use.

The alarm routine has built into it, octal and flexprint subroutines which may be used as such by the programmer. These portions of the alarm routine are also self resetting-- the use of these subroutines does not alter the sum of the Alarm, Octal and Flexprint Package.

These octal and flexprint subroutines are described on the following page.

1. To print octal information...

entry	result
37 76022 76002	C.R., (L), space, (R), space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = (R)initial (R) = 00 00000 00000 (00000) = 45 00000 76000
37 76016 76003	C.R., (L), space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = 00 00000 00000 (R) = a print order (00000)is not altered.
37 76016 76004	(L), space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = 00 00000 00000 (R) = a print order (00000) is not altered.
37 76016 76005	leftmost five octal digits of (L) and space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = (L)initial $\cdot 2^{15}$ (R) = a print order (00000) not altered.
37 76016 76006	leftmost two octal digits of (L) and space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = (L)initial $\cdot 2^{6}$ (R) = a print order (00000) not altered.
37 76022 76017	(00000), space ⟶ typewriter.
	final state... (Q) = 00 00000 00001 (L) = 00 00000 00000 (R) = a print order (00000) = 45 00000 76000

2. To print flex information...

entry	result
37 76016 76047	six flexcode characters in (L) and space ⟶ typewriter.
	final state... (Q) not altered (L) = 00 00000 00000 (R) = 00 00000 00000 (00000) not altered.

PX 71900-8-115

ALARM PRINT ROUTINE IE 002

76000	45 00000 76000	
76001	16 76026 76022	
76002	11 20000 00000	STORE (R)
76003	61 00000 76047	CARRIAGE RETURN
76004	55 76012 00005	SET FOR 12 DIGITS
76005	55 76012 00011	SET FOR 5 DIGITS
76006	55 76012 00012	SET FOR 2 DIGITS
76007	34 20000 00003	OCTAL PRINT LOOP (76007–76013)
76010	32 76037 00000	
76011	11 20000 76012	
76012	00 01000 10001	
76013	44 76014 76007	
76014	11 10000 76012	RESTORE FLAG
76015	61 00000 76021	SPACE
76016	37 76016 76017	SWITCH
76017	31 00000 00044	(00000) → (L)
76020	11 76000 00000	STORE JUMP IN CELL 00000
76021	37 76016 76004	PRINT (00000)
76022	37 76022 76023	SWITCH
76023	31 76000 00017	SET TO OBTAIN
76024	15 20000 76025	MAIN ROUTINE ADDRESS
76025	16 76000 00000	JUMP TO MAIN ROUTINE → (00000)
76026	15 76023 76025	RESTORE
76027	16 76027 76000	RESTORE
76030	31 76042 00047	
76031	37 76016 76047	PRINT "ALARM"
76032	41 00000 76033	ADJUST ADDRESS FOR PRINTING
76033	31 20000 00071	
76034	37 76016 76005	PRINT 5 DIGIT ADDRESS
76035	56 00000 00000	STOP
76036	00 00000 00000	AVAILABLE

PX 71900-8-115

76037	61 00000 76037	PRINT ORDER, FLEX 0
76040	00 00000 00052	FLEX 1
76041	00 00000 00074	2
76042	43 01130 12070	"ALARM" , 3
76043	00 00000 00064	4
76044	00 00000 00062	5
76045	00 00000 00066	6
76046	00 00000 00072	7
76047	61 00000 20045	} FLEX PRINT LOOP
76050	34 20000 00006	} FLEX PRINT LOOP
76051	47 76047 76015	} FLEX PRINT LOOP

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Nth Root Routine

Specifications

Identification Tag:	NRT-0
Type:	Subroutine
Assembly Routine Spec:	SUB 51316 03701
Storage:	36 instructions, addresses 10F00 thru 10F35 1 constant in program, address 10F36 37 words total program storage, addresses 10F00 thru 10F36 4 words temporary storage pool used, addresses 00027b thru 00033b The constant pool is used by this routine.
Program Entrance:	Address 10F02
Program Exit:	Address 10F01
Alarm Exit:	The alarm exit is used by this routine.
Drum Assignment:	Address 66064b thru 66130b
Machine Time:	Average execution time 2(n-2) + 5 milliseconds for $n \leq 50$
Mode of Operation:	Fixed point

Coded by:	W. Frank	November 25, 1955
Code Checked by:	W. Frank	November 28, 1955
Machine Checked by:	W. Frank	November 30, 1955
Approved by:	W. F. Bauer	December 1, 1955

Description

This subroutine extracts the n^{th} root of any number M, scaled at 2^{35}, and such that

$$\left| M \cdot 2^{35} \right| \leq 2^{35} - 1.$$

n must be an integer in the range

$$2 \leq n < 2^{16}$$

The routine must be entered with $M \cdot 2^{35}$ in A and $n \cdot 2^{0}$ in Q. The result will be left in A, scaled at 2^{35}, at the conclusion of the routine.

Programming Instructions

1. Place $M \cdot 2^{35}$ in A_R. (A_L) is ignored by this routine.
2. Place $n \cdot 2^{0}$ in Q.
3. Enter the subroutine with RJ 00F01 00F02, where 00F00 is the location of the first word of NRT-0.
4. The subroutine returns control to the cell following the RJ instruction with $(\sqrt[n]{M}) \cdot 2^{35}$ in A.

Alarm Conditions

The subroutine enters the alarm routine ALR-1 if n is negative or M is negative for n even. In either case, the word "alarm" is printed on the flexowriter, followed by the octal address of the RJ instruction used to enter NRT-0.

Execution Time

The time taken to find the n^{th} root of a number is inversely proportional to the magnitude of the number and directly proportional to the size of n. An average estimate, for $n \leq 50$, is approximately $3(n-2) + 5$ milliseconds.

Mathematical Method

An iterative procedure, employing the Newton-Raphson method*, is used to solve the equation

$$x^n = M$$

The process is of second order and is defined by

$$x_{i+1} = x_i + \frac{1}{n}\left[\frac{|M|}{(x_i)^{n-1}} - x_i\right]$$

where $x_0 \cdot 2^{35} = 2^{35} - 1$

The iteration is terminated when

$$\frac{|M|}{(x_i)^{n-1}} - x_i \geqslant 0$$

A secondary test is made to insure

$$|M| < (x_i)^{n-1}$$

This test is necessary, eventhough the process is monotonic; for, it is possible that truncation of the result of multiplication and division can violate this property. In that event, x_{i-1} is taken as the solution. A special case is $M = 0$, where the solution is $x = 0$ for all p.

Accuracy

The error in the result of this routine was found to be less than 10^{-10}; that is, for an input argument, which is correct to 35 bits, one can expect an answer which may be incorrect at most in the right octal digit.

*Scarborough, J.B., Numerical Mathematical Analysis, second edition, The John Hopkins Press, Baltimore, Md., 1950, p. 192.

NRT-O
Pg. 4 of 4

```
D                00F00 01024                       2000  00  00000  00000
D                10F00 51316                      66064  00  00000  00000
D                0CP00 00013                         15  00  00000  00000
10F00     37     75701 75702  B   ALARM           66064  37  75701  75702
10F01     MJ     00000 0          EXIT            66065  45  00000  00000
10F02     ZJ     00F03 00F01      ARG ZERO        66066  47  02003  02001
10F03     TP     A0000 0CP11                      66067  11  20000  00030
10F04     TP     Q0000 0CP10                      66070  11  10000  00027
10F05     QT     00016 A0000      N EVEN OR       66071  51  00020  20000
10F06     ZJ     00F09 00F07        OD            66072  47  02011  02007
10F07     TP     0CP11 A0000                      66073  11  00030  20000
10F08     SJ     00F00 00F09      ALARM           66074  46  02000  02011
10F09     TM     0CP11 0CP14                      66075  12  00030  00033
10F10     TP     00F36 0CP12      SET XO VALUE    66076  11  02044  00031
10F11     54     0CP10 20017  BRB                 66077  54  00027  20017
10F12     SJ     00F00 00F13      N NEGATIVE      66100  46  02000  02015
10F13     TU     A0000 00F17                      66101  15  20000  02021
10F14     RS     00F17 00015                      66102  23  02021  00017
10F15     RS     00F17 00015                      66103  23  02021  00017
10F16     SP     0CP12 00035      SET UP B        66104  31  00031  00043
10F17     RP     00000 00F19                      66105  75  00000  02023
10F18     MP     B0000 0CP12      XITH TO N-1     66106  71  30000  00031
10F19     TP     B0000 0CP13                      66107  11  30000  00032
10F20     TP     0CP14 A0000                      66110  11  00033  20000
10F21     TJ     0CP13 00F23                      66111  42  00032  02027
10F22     MJ     00000 00F30                      66112  45  00000  02036
10F23     LA     A0000 00035                      66113  54  20000  00043
10F24     DV     0CP13 A0000                      66114  73  00032  20000
10F25     ST     0CP12 A0000                      66115  36  00031  20000
10F26     SJ     00F27 00F30      CONVFRGENCE     66116  46  02033  02036
10F27     DV     0CP10 A0000                      66117  73  00027  20000
10F28     AT     0CP12 0CP12      XITH PLUS 1     66120  35  00031  00031
10F29     MJ     00000 00F16                      66121  45  00000  02020
10F30     TP     0CP11 A0000                      66122  11  00030  20000
10F31     SJ     00F32 00F34                      66123  46  02040  02042
10F32     TN     0CP12 A0000                      66124  13  00031  20000
10F33     MJ     00000 00F01                      66125  45  00000  02001
10F34     TP     0CP12 A0000                      66126  11  00031  20000
10F35     MJ     00000 00F01                      66127  45  00000  02001
10F36     37     77777 77777  B                   66130  37  77777  77777
```

PX 71900-8-116

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Fixed Point Decimal Card Read-In Routine (revised)

Specifications

Identification Tag:	CRI-2
Type:	Subroutine
Assembly Routine Spec:	SUB 51353 25039 (See special assembly instructions)
Storage:	201 instructions, addresses 00E00 thru 04E35
	49 constants in program, addresses 00N00 thru 00H38
	24 words temporary storage required, but not stored with the program
	250 words total program storage, addresses 00E00 thru 00H38
	10 words temporary storage pool used, addresses 00027b thru 00040b
	The constant pool is used by this routine
Program Entrance:	Address 00E02
Program Exit:	Address 00E01
Alarm Exit:	The alarm exit is used by this routine.
Drum Assignment:	Address 66131b thru 66522b
Machine Time:	.5 seconds per card
Mode of Operation:	Fixed point

Coded by:	C. Koos	November 9, 1955
Code Checked by:	C. Koos	November 11, 1955
Machine Checked by:	C. Koos	November 14, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine reads fixed point decimal numbers from cards, converts these numbers to their octal equivalents and stores them in the computer at locations specified by a base address of the parameter word and a location number on the card.

The input cards are divided into four fields, occupying columns 5-23, 24-42, 43-61 and 62-80. Each field contains in the order indicated:

a. a five place decimal location number which is added to the base address of the parameter word and which positions the number in the proper cell.

b. a ten digit fractional mantissa with decimal point at extreme left of field.

c. a two digit exponent of 10 (decimal scaling factor).

d. a two digit exponent of 2 (binary scaling factor) which scales the number in the computer accordingly.

A punch in the eleven row above the least significant digit of the mantissa or the scaling factors indicates a negative value. Any number of the fields, in any combination, may be used. Blank portions are ignored. The routine will also ignore completely blank cards, and will stop only upon finding a 12 punch in column 80.

Accuracy

The routine will give a conversion accurate to 35 bits rounded.

Assembly Instructions

Although only 250 instructions are assembled, an additional 24 cells (bringing the total to 274) must be allowed for temporary storage.

Programming Instructions

1. The first time the routine is entered, include the instruction EF 00000 00A15 (assuming that the routine was assigned to region 00A00 for assembly). This instruction picks the first card, and is not used except when the read station is empty. Do not include this step if you have previously used the routine and have not emptied the read hopper.

2. Enter the routine with an RJ instruction. Assuming that the routine was assigned to the region indicated in step 1, use the instruction RJ 00A01 00A02.

3. Enter the parameter word. The parameter word should follow the RJ instruction and should contain a base address in its rightmost 15 bits; the remainder of the word should consist of zeros. This base address is added to the relative location numbers from the cards in order to find the correct addresses for storing.

4. Cards will be read until the routine encounters a card with a 12 punch in column 80.

5. At the conclusion of the routine, control will be returned to the cell immediately following the parameter word.

Alarm Conditions

An alarm will occur:

1. If the routine encounters a power of ten which has an absolute value greater than 40.

2. 2. If the final scaled form of the number requires more than 35 bits.

3. If an attempt is made to load a number into an illegal address.

In all cases the tag word "CRI-2", along with the address of the cell from which the subroutine was entered, will be printed on the flexowriter. Starting at this time will cause a fault, due to an attempt to execute the parameter word. Increasing PAK by one, and then starting, will result in re-entering this routine.

CRI-2
Pg. 4 of 7
revised 12-9-55

D			00A00	00013			15	00	00000	00000
D			00B00	51353			66131	00	00000	00000
D			01B00	51378			66162	00	00000	00000
D			02B00	51423			66237	00	00000	00000
D			03B00	51467			66313	00	00000	00000
D			04B00	51518			66376	00	00000	00000
D			00C00	51564			66454	00	00000	00000
D			00E00	01024			2000	00	00000	00000
D			01E00	01049			2031	00	00000	00000
D			02E00	01094			2106	00	00000	00000
D			03E00	01138			2162	00	00000	00000
D			04E00	01189			2245	00	00000	00000
D			00F00	01274			2372	00	00000	00000
D			00H00	01235			2323	00	00000	00000
D.			00M00	51554			66442	00	00000	00000
D			00N00	01225			2311	00	00000	00000
D			00T00	00023			27	00	00000	00000
	B00	37	75701	75702	B	ALARM EXIT	66131	37	75701	75702
	B01	MJ	00000			NORMAL EXIT	66132	45	00000	00000
	B02	EF	00F20	00E19		ENTRANCE	66133	17	02416	02023
	B03	SP	00E01	00015			66134	31	02001	00017
	B04	TU	A0000	00E05			66135	15	20000	02005
	B05	TP	00000	00T00		STORE ADDR	66136	11	00000	00027
	B06	TP	00A00	00T01			66137	11	00015	00030
	B07	TP	00H05	00T04		SET COUNTER	66140	11	02330	00033
	B08	TV	00N00	00E24			66141	16	02311	02030
	B09	RP	10016	00E11		CLEAR	66142	75	10020	02013
	B10	TP	00A00	00F00		STORAGE	66143	11	00015	02372
	B11	RJ	00N09	00N09			66144	37	02322	02322
	B12	LQ	00T03	00006			66145	55	00032	00006
	B13	QT	00A01	00T05			66146	51	00016	00034
	B14	LQ	Q0000	00002			66147	55	10000	00002
	B15	11	00T01	20104	BRB		66150	11	00030	20104
	B16	QA	00H01	00T09		STORE WORD 4	66151	52	02324	00040
	B17	LQ	Q0000	00017			66152	55	10000	00021
	B18	QT	00H02	00T08		STORE WORD 3	66153	51	02325	00037
	B19	11	00T05	20105	BRB		66154	11	00034	20105
	B20	LQ	00T02	00006			66155	55	00031	00006
	B21	QA	00H03	00T07		STORE WORD 2	66156	52	02326	00036
	B22	LQ	Q0000	00017			66157	55	10000	00021
	B23	QT	00H02	00T06		STORE WORD 1	66160	51	02325	00035
	B24	MJ	00000	01E00			66161	45	00000	02031
	1B00	RS	00T04	00A03			66162	23	00033	00020
	1B01	ZJ	01E02	01E44			66163	47	02033	02105
	1B02	TV	00N01	01E15			66164	16	02312	02050
	1B03	TP	00N02	01E11			66165	11	02313	02044
	1B04	TP	00N03	01E07			66166	11	02314	02040
	1B05	TP	00H18	00T02		SET COUNTER	66167	11	02345	00031
	1B06	TP	00N04	A0000			66170	11	02315	20000
	1B07	11	00T06	10002	BRB		66171	11	00035	10002
	1B08	QA	00H18	01E10		EXTRACT EXP	66172	52	02345	02043
	1B09	EJ	00N04	01E12			66173	43	02315	02045
	1B10	MP	00T04	00H15			66174	71	00033	02342
	1B11	AT	00F00	00F00		STORE	66175	35	02372	02372
	1B12	RA	01E07	00A02			66176	21	02040	00017
	1B13	RA	01E11	00A04			66177	21	02044	00021
	1B14	IJ	00T02	01E06		IJ ON 3	66200	41	00031	02037
	1B15	MJ	00000	01E16			66201	45	00000	02051
	1B16	TP	00N05	01E07			66202	11	02316	02040
	1B17	RJ	01E15	01E05			66203	37	02050	02036
	1B18	TP	01E11	01E36			66204	11	02044	02075
	1B19	TV	00N06	01E39			66205	16	02317	02100
	1B20	TU	00N00	01E24			66206	15	02311	02061
	1B21	TV	00N03	01E27			66207	16	02314	02064
	1B22	TP	00H18	00T02		SET COUNTER	66210	11	02345	00031
	1B23	TU	00N03	01E28			66211	15	02314	02065
	1B24	TP	00H19	00T03		SET COUNTER	66212	11	02346	00032

CRI-2
Pg. 5 of 7
revised 12-9-55

1B25	TV	04E19	01E30		66213	16	02270	02067
1B26	TP	00A00	00T05		66214	11	00015	00034
1B27	LQ	00A03	10002	EXTR TO Q	66215	55	00020	10002
1B28	QT	00T06	A0000		66216	51	00035	20000
1B29	ZJ	01E30	01E31		66217	47	02067	02070
1B30	RA	00T05	00H04		66220	21	00034	02327
1B31	RA	01E30	00A03		66221	21	02067	00020
1B32	LQ	Q0000	00001	SHIFT EXTR	66222	55	10000	00001
1B33	IJ	00T03	01E28		66223	41	00032	02065
1B34	RA	01E28	00A02		66224	21	02065	00017
1B35	MP	00T04	00T05		66225	71	00033	00034
1B36	AT	00F08	00F08	STORE	66226	35	02402	02402
1B37	RA	01E36	00A04		66227	21	02075	00021
1B38	IJ	00T02	01E24	IJ ON 3	66230	41	00031	02061
1B39	MJ	00000	01E40		66231	45	00000	02101
·1B40	TU	00N01	01E24		66232	15	02312	02061
1B41	RA	01E27	00H05		66233	21	02064	02330
1B42	RJ	01E39	01E22	GET REL ADDR	66234	37	02100	02057
1B43	MJ	00000	00N09		66235	45	00000	02322
1B44	RP	30004	02E01		66236	75	30004	02107
2B00	TP	00T06	00F20		66237	11	00035	02416
2B01	TP	00H18	00T02	SET COUNTER	66240	11	02345	00031
2B02	TU	04E15	02E05		66241	15	02264	02113
2B03	TV	04E15	02E30		66242	16	02264	02144
2B04	TV	03E07	02E34		66243	16	02171	02150
2B05	TP	00F04	00T03	GET DEC EXP	66244	11	02376	00032
2B06	TV	00N07	02E21		66245	16	02320	02133
2B07	TU	00N06	02E22		66246	15	02317	02134
2B08	TU	00N07	02E23		66247	15	02320	02135
2B09	TU	02E29	02E26		66250	15	02143	02140
2B10	TP	00T03	A0000		66251	11	00032	20000
2B11	RP	20004	02E13	TEST EXP	66252	75	20004	02123
2B12	TJ	00H25	02E16		66253	42	02354	02126
2B13	RJ	00E24	00N09		66254	37	02030	02322
2B14	RJ	00E24	00N09		66255	37	02030	02322
2B15	MJ	00000	04E05	TO ALARM	66256	45	00000	02252
2B16	QT	00A01	00T04		66257	51	00016	00033
2B17	RS	02E21	00T04		66260	23	02133	00033
2B18	LQ	00T04	00015		66261	55	00033	00017
2B19	RS	02E22	00T04		66262	23	02134	00033
2B20	RA	02E23	00T04		66263	21	02135	00033
2B21	RS	00T03	00H27		66264	23	00032	02356
2B22	TP	00H32	00T04		66265	11	02363	00033
2B23	TP	00H21	00T05		66266	11	02350	00034
2B24	LQ	00T03	00015		66267	55	00032	00017
2B25	RA	02E26	00T03		66270	21	02140	00032
2B26	MP	00H04	00T04		66271	71	02327	00033
2B27	SF	A0000	00T01		66272	74	20000	00030
2B28	SJ	02E29	02E30		66273	46	02143	02144
2B29	AT	00H04	A0000	ROUND	66274	35	02327	20000
2B30	TP	A0000	00F04	STORE	66275	11	20000	02376
2B31	TP	00T01	A0000		66276	11	00030	20000
2B32	TJ	00H33	02E34	TJ ON 37	66277	42	02364	02150
2B33	ST	00A05	A0000		66300	36	00022	20000
2B34	AT	00T05	00F16	STORE	66301	35	00034	02412
2B35	RA	02E05	00A02		66302	21	02113	00017
2B36	RA	02E30	00A03		66303	21	02144	00020
2B37	RA	02E34	00A03		66304	21	02150	00020
2B38	IJ	00T02	02E05	IJ ON 3	66305	41	00031	02113
2B39	TU	03E21	03E04		66306	15	02207	02166
2B40	TU	00N02	03E05		66307	15	02313	02167
2B41	TU	03E13	03E06		66310	15	02177	02170
2B42	TU	04E15	03E07		66311	15	02264	02171
2B43	TU	00N03	03E08		66312	15	02314	02172
3B00	TV	04E15	03E33		66313	16	02264	02223
3B01	TV	00N02	03E35		66314	16	02313	02225
3B02	RJ	00E24	00N09	READ 11 ROW	66315	37	02030	02322

CRI-2
Pg. 6 of 7
revised 12-9-55

3B03	TP	00H18	00T02		SET COUNTER	66316	11	02345	00031
3B04	TP	00F08	00F08			66317	11	02402	02402
3B05	TP	00F00	00T04			66320	11	02372	00033
3B06	TP	00F16	00T05			66321	11	02412	00034
3B07	TP	00F04	00F16			66322	11	02376	02412
3B08	LQ	00T06	10035			66323	55	00035	10043
3B09	QJ	03E10	03E11		TEST B	66324	44	02174	02175
3B10	TN	00T04	00T04			66325	13	00033	00033
3B11	LQ	Q0000	00033			66326	55	10000	00041
3B12	QJ	03E15	03E13		TEST D	66327	44	02201	02177
3B13	MP	00F16	00F08			66330	71	02412	02402
3B14	MJ	00000	03E23			66331	45	00000	02211
3B15	TP	00H28	A0000			66332	11	02357	20000
3B16	TJ	00F08	03E19			66333	42	02402	02205
3B17	RS	00T04	00H19			66334	23	00033	02346
3B18	LA	00F08	00009			66335	54	02402	00011
3B19	TN	00T05	00T05			66336	13	00034	00034
3B20	RS	00T04	00H38			66337	23	00033	02371
3B21	LA	00F08	00033			56340	54	02402	00041
3B22	RJ	03E30	03E25			66341	37	02220	02213
3B23	TV	00N08	03E30			66342	16	02321	02220
3B24	TP	00H14	00F16			66343	11	02341	02412
3B25	DV	00F16	00T03			66344	73	02412	00032
3B26	LA	A0000	00035			66345	54	20000	00043
3B27	DV	00F16	00F16			66346	73	02412	02412
3B28	SP	00T03	00035			66347	31	00032	00043
3B29	SA	00F16	00000			66350	32	02412	00000
3B30	MJ	00000	03E31			66351	45	00000	02221
3B31	SF	A0000	00T01			66352	74	20000	00030
3B32	LA	A0000	00001			66353	54	20000	00001
3B33	TP	A0000	00F04		STORE	66354	11	20000	02376
3B34	RA	00T04	00T01			66355	21	00033	00030
3B35	AT	00T05	00F00		STORE	66356	35	00034	02372
3B36	RP	20005	03E38			66357	75	20005	02230
3B37	RA	03E04	00A02			66360	21	02166	00017
3B38	RA	03E33	00A03			66361	21	02223	00020
3B39	RA	03E35	00A03			66362	21	02225	00020
3B40	IJ	00T02	03E04		1J ON 3	66363	41	00031	02166
3B41	RJ	00N09	00N09		READ 12 ROW	66364	37	02322	02322
3B42	TU	00N08	03E49			66365	15	02321	02243
3B43	TU	00E02	04E00			66366	15	02002	02245
3B44	TU	00N02	04E10			66367	15	02313	02257
3B45	TU	00N03	04E22			66370	15	02314	02273
3B46	TU	04E15	03E48			66371	15	02264	02242
3B47	TP	00H18	00T02		SET COUNTER	66372	11	02345	00031
3B48	TP	00F04	00F04			66373	11	02376	02376
3B49	TP	00F12	A0000			66374	11	02406	20000
3B50	ZJ	04E01	04E00			66375	47	02246	02245
4B00	EJ	00F20	04E26			66376	43	02416	02277
4B01	AT	00T00	A0000			66377	35	00027	20000
4B02	TJ	00H35	04E07			66400	42	02366	02254
4B03	TJ	00H36	04E05		TO ALARM	66401	42	02367	02252
4B04	TJ	00A02	04E07			66402	42	00017	02254
4B05	11	00H37	75756	BRB	TAG TO ALARM	66403	11	02370	75756
4B06	MJ	00000	00E00		TO ALARM	66404	45	00000	02000
4B07	TV	A0000	04E25			66405	16	20000	02276
4B08	TP	00F04	A0000			66406	11	02376	20000
4B09	ZJ	04E10	04E25			66407	47	02257	02276
4B10	RS	00F00	00H34			66410	23	02372	02365
4B11	SJ	04E12	04E05			66411	46	02261	02252
4B12	RA	A0000	00A05			66412	21	20000	00022
4B13	TJ	00H34	04E15			66413	42	02365	02264
4B14	MJ	00000	04E16			66414	45	00000	02265
4B15	RS	00F04	00F04		CLEAR	66415	23	02376	02376
4B16	TV	A0000	04E17			66416	16	20000	02266
4B17	SP	00F04	00000			66417	31	02376	00000
4B18	SJ	04E19	04E22			66420	46	02270	02273

```
4B19  RA  A0000  00H04         ROUND          66421  21  20000  02327
4B20  TP  A0000  A0000                        66422  11  20000  20000
4B21  SJ  04E05  04E22                        66423  46  02252  02273
4B22  LQ  00T06  00031                        66424  55  00035  00037
4B23  QJ  04E24  04E25                        66425  44  02275  02276
4B24  TN  A0000  A0000                        66426  13  20000  20000
4B25  TP  A0000  00                           66427  11  20000  00000
4B26  RA  03E49  00A02                        66430  21  02243  00017
4B27  RA  04E00  00A02                        66431  21  02245  00017
4B28  RA  04E10  00A02                        66432  21  02257  00017
4B29  RA  03E48  00A02                        66433  21  02242  00017
4B30  RA  04E22  00A02                        66434  21  02273  00017
4B31  IJ  00T02  03E48         1J ON 3        66435  41  00031  02242
4B32  TP  00T01  A0000                        66436  11  00030  20000
4B33  ZJ  04E34  00E02                        66437  47  02307  02002
4B34  RA  00E01  00A03                        66440  21  02001  00020
4B35  MJ  00000  00E01         TO EXIT        66441  45  00000  02001
 M00  00  00H19  01E00                        66442  00  02346  02031
 M01  00  00H20  01E16                        66443  00  02347  02051
 M02  AT  00F00  00F00                        66444  35  02372  02372
 M03  11  00T06  10002  BRB                   66445  11  00035  10002
 M04  MP  00T04  00H14                        66446  71  00033  02341
 M05  LQ  00T06  00034                        66447  55  00035  00042
 M06  00  00H32  01E40                        66450  00  02363  02101
 M07  00  00H21  00H27                        66451  00  02350  02356
 M08  00  00F12  03E31                        66452  00  02406  02221
 M09  RP  30003  00E12            READ        66453  75  30003  02014
 C00  76  47777  00030  B       ONE ROW       66454  76  47777  00030
 C01  00  00017  77400  B                     66455  00  00017  77400
 C02  00  00017  77777  B                     66456  00  00017  77777
 C03  00  00017  77700  B                     66457  00  00017  77700
 C04  00  00000  00  1  B      10 TO  0       66460  00  00000  00001
 C05  00  00000  00012  B             1       66461  00  00000  00012
 C06  00  00000  00144  B             2       66462  00  00000  00144
 C07  00  00000  01750  B             3       66463  00  00000  01750
 C08  00  00000  23420  B             4       66464  00  00000  23420
 C09  00  00003  03240  B             5       66465  00  00003  03240
 C10  00  00036  41100  B             6       66466  00  00036  41100
 C11  00  00461  13200  B             7       66467  00  00461  13200
 C12  00  05753  60400  B             8       66470  00  05753  60400
 C13  00  73465  45000  B             9       66471  00  73465  45000
 C14  11  24027  62000  B      10 TO 10       66472  11  24027  62000
 C15  00  00000  00  1  B                     66473  00  00000  00001
 C16  00  00000  00 12  B                     66474  00  00000  00012
 C17  00  00000  00 13  B                     66475  00  00000  00013
 C18  00  00000      3  B                     66476  00  00000  00003
 C19  00  00000  00011  B                     66477  00  00000  00011
 C20  00  00000      4  B                     66500  00  00000  00004
 C21  00  00000  00101  B      65 DECIMAL     66501  00  00000  00101
 C22  00  00000  00 40  B      32 DECIMAL     66502  00  00000  00040
 C23  00  00000         B                     66503  00  00000  00000
 C24  00  00            B                     66504  00  00000  00000
 C25  00  00000  00 12  B                     66505  00  00000  00012
 C26  00  00000  00024  B                     66506  00  00000  00024
 C27  00  00000  00 36  B                     66507  00  00000  00036
 C28  00  00000  00 51  B                     66510  00  00000  00051
 C29  00  00000  00  1  B      10 TO  0       66511  00  00000  00001
 C30  11  24027  62000  B            10       66512  11  24027  62000
 C31  25  53616  57055  B            20 032   66513  25  53616  57055
 C32  31  17454  47150  B      10 TO 30 065   66514  31  17454  47150
 C33  00  00000  00 45  B      37 DECIMAL     66515  00  00000  00045
 C34  00  00000  00044  B      36 DECIMAL     66516  00  00000  00044
 C35  00  00000  02000  B                     66517  00  00000  02000
 C36  00  00000  40000  B                     66520  00  00000  40000
 C37  16  12145  67404  B      TAG CRI02      66521  16  12145  67404
 C38  00  00000  00104  B      68 DECIMAL     66522  00  00000  00104
```

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Flexowriter Memory Dump (Revised)

Specifications

Identification Tag:	MDP-0
Type:	Service Routine (with subroutine entrance)
Special Storage:	The constant pool and temporary storage pool are not used by this routine.
Service Entrance:	Address 40004b
Program Entrance:	40004b
Program Exit:	40020b
Alarm Exit:	The alarm exit is not used by this routine
Machine Time:	Approximately 27 sec/100 words on paper tape Approximately 2.7 min/100 words on typewriter

Coded by:	R. Beach	October 27, 1955
Code Checked by:	R. Beach	October 27, 1955
Machine Checked by:	R. Beach	October 28, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine dumps, in octal, the contents of consecutive storage cells onto paper tape, in flexowriter code, or onto the directly connected flexowriter. It stores ES, A and Q and bootstraps itself into ES for execution. ES, A and Q are restored at the conclusion of a dump.

The address is printed for the first word of the dump, and for each subsequent word whose address is zero, modulo eight. An extra carriage return is included in order to provide double spacing at the end of each block of eight words. If a cell contains zero, only the first digit is printed. However, if the address is to be printed then all twelve zeros are printed.

The flexowriter should be in a shift down state, as the numbers are then larger, and hence more legible.

This routine makes use of MDP-1 for part of its operation, and both routines must therefore be present on the drum.

Operating Instructions

1. When routine is used as a service routine start at 40004b. Machine halts on an MS stop with Q clear.

   Insert the parameter word in Q (see below) and start. Machine executes dump and halts on 56 00000 40004b.

2. When routine is used as a subroutine enter with 37 40020 40004b followed by the parameter word (see below). At the conclusion of the dump, control will be transferred to the cell following that one in which the parameter word is located.

3. The parameter word has the form

   AB uuuuu vvvvv

   where uuuuu and vvvvv are the first and last words to be dumped respectively. For a typewriter dump the octal digits A and B must both be zero while for a paper tape dump B is to be different from zero.

4. Restore
   To restore ES, A and Q at any time after the parameter word has been given, set PAK to 40040b and start.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Bioctal Memory Dump (revised)

Specifications

Identification Tag:	MDP-1
Type:	Service Routine (with subroutine entrance)
Special Storage:	The constant pool and temporary storage pool are not used by this routine.
Service Entrances:	Bioctal Tape: 40005b Binary Cards: 40016b
Program Entrance:	Bioctal Tape: 40005b Binary Cards: 40016b
Program Exit:	40020b
Alarm Exit:	The alarm exit is not used by this routine.

Coded by:	R. Beach C. Koos	October 25, 1955
Code checked by:	R. Beach	October 28, 1955
Machine Checked by:	R. Beach	October 28, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

PX 71900-8-90

Description

1. General

This routine dumps, in binary, the contents of consecutive storage cells onto paper tape or cards for later read in by FRI-0 or CRI-1. The routine stores ES, A and Q and bootstraps itself into ES for execution. At its conclusion, it restores ES, A and Q. The type of dump is determined by the entrance used, and the range of the dump is given by a parameter word.

2. Binary Tape

Binary (or bi-octal) tape is punched as specified by the parameter word. The routine punches leader, insert and check addresses, check sums, stop code for FRI-0 and a trailer. The leader and initial seventh level punch may be suppressed if desired, as may, also, the stop code and trailer. A check sum will be punched for each dump. If a dump of more than 808 words is specified, the routine treats this problem as more than one dump and executes as many separate dumps as there are multiples of 808.

3. Binary Cards

Because the Bull Reproducer is limited to 160 punches per card, with no more than 60 punches in any one row, binary cards are dumped in groups of three cards for later merging on off-line equipment into one card. Each card contains up to eight words.

Card No. 1:

Row 9	Address of first word in group	(Col. 1 - 15)
	Number of words in the group of three cards	(Col. 17 - 21)
	Sum of all words in the group of three cards	(Col. 32 - 72)
Row 8	First word (field one)	(Col. 1 - 36)
	Second word (field two)	(Col. 37 - 72)
Row 6, 7	Contains 4 more words as in Row 8. In addition, Row 7 has a punch in column 80 to identify it as the first card in the group.	

Card No. 2: Contains up to 8 words in rows five, four, three and two with a punch in row eleven, columns 80 and 75, to identify it as the second card.

Card No. 3: Contains up to 8 words in the remaining four rows and an identification punch in row eleven, column 80.

If a stop code is specified, a punch is placed in row twelve, column 80 of the second card of the last group of three cards dumped. The dump is accomplished at full card punch speed.

To keep the number of punches at a minimum, the number of binary ones in each word is counted, and the word is complemented if this number exceeds eighteen. In this event a punch is placed in column 78 of that row, for field one, and in column 79, for field two, indicating this complementation.

The output of the dump may now be processed by the IBM equipment on an off-line basis. Each group of three cards is merged into one binary card containing up to twenty-two words of information. This merging may be accomplished by two passes on the sorter followed by two passes on the reproducer, or by one pass on the reproducer (gang punch operation), followed by one pass on the sorter.

Operating Instructions

I. Binary Tape

a. Service Routine
1. Start at 40005b. Computer halts on an MS order.
2. Insert parameter word (see below) in Q and start
3. At conclusion of dump, computer will halt on 56 00000 40005b

b. Subroutine
1. Enter with 37 40020 40005b followed by parameter word (see below)
2. At and end of dump, control is transferred to the cell following the one containing the parameter word.

II. Binary Cards

a. Service Routine
1. Start at 40016b. Computer halts on an MS order.
2. Insert parameter word (see below) in Q and start.
3. At conclusion of dump, computer halts on 56 00000 40016b.

b. Subroutine
1. Enter with 37 40020 40016b followed by parameter word (see below)
2. At conclusion of dump, control is transferred to the cell following the one containing the parameter word.

III. Parameter Word

The parameter word is

AB uuuuu vvvvv

where uuuuu and vvvvv are the first and last words to be dumped respectively. The second octal digit, B, specifies a stop code if non-zero; no stop code if zero.

For binary tape dumps, setting the first octal digit, A, of the parameter word different from zero suppresses a leader and initial seventh level punch.

IV. Restore

At any time during a dump, ES, A and Q may be restored by starting at

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

OCTAL CARD DUMP (revised)

Specifications

Identification Tag:	MDP-2
Type:	Service routine (with subroutine entrance)
Special Storage:	The constant and temporary storage pools are not used by this routine.
Service Entrance:	Address 40015b
Program Entrance:	Address 40015b
Program Exit:	Address 40020b
Alarm Exit:	The alarm exit is not used by this routine.
Machine Time:	2.7 seconds plus 0.5 seconds per card maximum machine time.

Coded by:	C. Koos	July 21, 1955
Machine Checked by:	C. Koos	August 5, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine will dump the contents of a group of consecutive (either ES or MD) storage cells onto cards. Each card will contain (in octal) four consecutive words and the address of the cell containing the first word on the card. The following card columns are used:

Columns 1 thru 5	Address of the first word
Columns 13 thru 24	First word
Columns 25 thru 36	Second word
Columns 37 thru 48	Third word
Columns 49 thru 60	Fourth word

Any card for which all four words are equal to zero is omitted and the next card produced carries a punch in the 12 row of column 9. The first and last cards of every dump will be produced even if they contain all zeros and the last card will carry a punch in the 12 row of column 10. In addition, each card contains an identifying 12 punch in column 8.

This routine bootstraps itself into ES to operate and at its conclusion restores the machine to its original state.

Operating Instructions (to be followed when the routine is used as a service routine)

1. Put the computer in test mode, high speed (this step is unnecessary for a dump of all ES only).

2. Set PAK to 40015b and start.

3. Computation will halt with an MSO instruction and Q will contain all zeros.

4. Manually insert the parameter word into Q.

   a. a parameter word of all zeros will dump ES

   b. in all other cases, place the first address in Q_u and the last address in Q_v

   c. the range of the dump may not extend from ES to drum addresses

5. The machine will halt with an MSO instruction when the dump is completed and the machine has been restored to its original state.

6. If another dump is required, it is necessary only to press the start button again to return to stop 3 above.

7. If the operator wishes to stop a dump at any time after step 3 above, he needs only to make a forced stop, master clear, and MD start with PAK set to 40040b. The machine will then be restored to its original state and computation will halt with the same MSO instruction mentioned in step 5. This same procedure is applicable if an SCC fault occurs after entering an illegal parameter word in Q.

Programming Instructions (to be followed when the routine is used as a subroutine)

1. Enter the routine with an RJ instruction

Use the instruction 37 40020 40015b. The word in your program immediately following the RJ instruction must contain the parameter word (as described in step 4 of "Operating Instructions" above). If the RJ instruction is given at address n the parameter word will be at address n + 1 and at the conclusion of the dump control will be returned to the instruction address n + 2.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

CHANGED WORD POST-MORTEM ROUTINE (revised)

Specifications

Identification Tag:	MDP-3
Type:	Service Toutine (with subroutine entrance)
Special Storage:	The constant pool and temporary storage pool are not used by this routine.
Service Entrance:	Address 40037b
Program Entrance:	40037b
Program Exit:	40020b
Alarm Exit:	The alarm exit is not used by this routine.
Machine Time:	(14.1 + .5n) seconds where n=number of cards punched.

Coded by:	R. Beach	October 26, 1955
Code Checked by:	R. Beach	October 26, 1955
Machine Checked by:	R. Beach	October 26, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine compares ES with the MD image of ES and prints out those words of ES which are not the same as their correspondent in the image. ES is not altered by the routine, and the MD image is up-dated to be identical with ES when exit is made from the routine.

The routine stores ES at addresses 66000b to 67777b and reads portions of this image and the regular image (76000b - 77777b) into ES and compares words.

If the corresponding words are the same, they are replaced by zero, unless the new value is zero. In the latter case the word is replaced by 45 40037 40020b. The changed words and zeros are then read into ES. ES is then dumped on the line printer. (Note: Until the line printer is in use, this dump will be made onto cards by employing MDP-2). ES and the 76000b image are then restored from the 66000b image.

Each card contains four words. If any one word is zero, it should be ignored as it is not a changed word. A word which has been changed to zero has been given the arbitrary tag 45 40037 40020b and will be punched as such. Also, a word that was changed to this tag will be identified in the same manner. The programmer must therefore distinguish between these two cases.

Operating Instructions

1. When routine is used as a service routine set PAK to 40037b. Routine will find changed words, print them out, and stop on 56 00000 40037b.

2. When routine is used as a subroutine enter routine with 37 40020 40037b. Operation of routine is the same except that routine exits to address y+1 if y is the address of the RJ instruction used to enter the routine.

3. Most service routines use all or parts of ES and their activation will destroy the old 76000b image. Hence, if a changed word comparison is desired, the execution of MDP-3 must precede the use of other post-mortem routines.

Alarm Conditions

There are no alarm conditions in this routine. However, if the routine hangs up during punching, or if the machine is halted during punching, a start at 40040b will clear the punch, restore ES, and up-date the 76000b image.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

STORAGE TO MAGNETIC TAPE TRANSFER ROUTINE (revised)

Specifications

Identification Tag:	STT-0
Type:	Service routine (with a program entry available)
Special Storage:	The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40006b
Program Entrance:	Address 40006b
Program Exit:	Address 40020b
Alarm Exit:	The alarm exit is not used by this routine
Machine Time:	5.6 seconds for transfer of (ES)

Coded by:	R. Beach	May 11, 1955
Code Checked by:	C. Koos	August 14, 1955
Machine Checked by:	C. Koos	August 20, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine transfers information from the internal computer memory to magnetic tape where it will be stored until read back in again by TST-0.

A parameter word is used to specify

1. The location of data to be stored
2. The MT unti to be used for storage
3. Whether or not MT is to be rewound to its original position after storage
4. The address to which control is to be transferred when the data is read back by TST-0.

When using STT-0 as a subroutine the parameter word follows the RJ instruction used to enter the routine. When using STT-0 as a service routine the parameter word is manually entered in Q when the computer halts (after being started at the service entrance).

At the time of entry the routine stores (ES) on the drum, bootstraps itself into ES, stores (A) and (Q) and obtains the parameter word. At the conclusion the routine restores (ES), (A) and (Q) and transfers control to the exit instruction.

The routine stores one block of information in addition to the number of blocks necessary for storing the data, as follows:

1. The first half of the first block contains (Q), (A), the parameter word and twelve zero words.
2. The second half of the first block thru the first half of the last block inclusive contain the information to be stored.
3. The last half of the last block contains the sum of the data (that is, the double precision sum of the split extension of each word), the number of blocks transferred to tape, the starting and stopping addresses for the transfer, and eleven zero words.

Parameter Word

This parameter word is of the form BC DEEFF GGGGG, where B, C, D, E, F, and G are all octal digits.

B. The octal digit B determines whether (ES) is to be stored on MT. If $B = 0$, (ES) will be stored; if $B \neq 0$, (ES) will not be stored.

C. The octal digit C determines whether MT is to be rewound to its original position after the data has been transferred. If $C = 0$ the rewind will be executed, if $C \neq 0$ it will not be.

D. The octal digit D determines the MT unit on which the data is to be stored. MT units are specified by the same digits used in the standard 1103 MT commands.

E. The two octal digit number EE specifies the address of the first word to be transferred from internal storage to tape. This number is the integer part of the first address divided by 8^3. That is, (EE)(512) is the address of the first cell to be transferred.

F. The two octal digit number FF specifies the address of the last word to be transferred. As in E above this number must also be a multiple of 512. (FF)(512) is the address of the last word to be transferred.

G. The V-address portion of the parameter word (GGGGG) specifies the address to which PAK is to be set when the transferred information is read back to internal memory by TST-0.

As an example consider the parameter word 01 24246 00017b. This specifies a transfer of (ES) and the contents of cells 42000b thru 45777b with no rewinding after the transfer. PAK will be set to 00017b by TST-0 when the routine is read back to internal memory.

Operating Instructions (to be followed when the routine is used as a service routine)

1. Set PAK to 40006b and start.

2. Computation halts with an MS instruction.

3. Enter the parameter word in Q and start.

4. Computation halts when the transfer is completed, setting PAK to the address specified in the parameter word.

Programming Instructions (to be followed when the routine is to be used as a subroutine)

1. Enter the routine with the RJ instruction 37 40020 40006b. If the RJ instruction is stored at address n the parameter word should be in address n + 1 and at its conclusion the routine will transfer control to the instruction in address n + 2.

Restore

To restore (ES), (A), and (Q) at any time before normal completion set PAK to 40040b and start.

The magnetic tape will be rewound at this time if the parameter word specifies a rewind.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Magnetic Tape to Storage Transfer Routine (revised)

Specifications

Identification Tag:	TST-0
Type:	Service routine (with a program entry available)
Special Storage:	The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40007b
Program Entrance:	Address 40007b
Program Exit:	Address 40020b
Alarm Exit:	The alarm exit is used by this routine
Machine Time:	5.6 seconds for transfer of (ES)

Coded by:	R. Beach	May 11, 1955
Code Checked by:	C. Koos	August 13, 1955
Machine Checked by:	C. Koos	August 21, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

Description

This routine has been designed to read back into the internal computer memory the information transferred to MT by STT-0. A parameter word is used to tell the routine which MT contains the information to be transferred and the address to which PAK is to be set after the routine has finished operating.

When used as a subroutine, the parameter word follows the RJ instruction transferring control to TST-0. When used as a service routine the parameter word is entered in the Q register before activating the routine.

The routine stores (ES) on the drum, bootstraps itself into ES and reads in one block from magnetic tape. It examines the parameter word used by STT-0 and loads the specified portions of memory while computing the sum as the data is transferred. The sum is checked against the sum placed on MT by STT-0.

If the sum is correct, the parameter word is consulted to determine the address to which PAK is to be set and the proper address is placed in the exit instruction. The parameter word from STT-0 is checked to determine whether or not the MT is to be rewound after the transfer and a rewind command given if rewind was specified when STT-0 was used to store the data. (A) and (Q) are then set from values stored on MT, (ES) is restored, and control is transferred to the exit instruction.

Parameter Word

The form of the parameter word is OX Y0000 ZZZZZ, where X, Y, and Z are octal digits.

X. The octal digit X determines the cell to which control will be transferred to at the conclusion of the routine.

If X = 0 control will be transferred to the address specified in the parameter word used for STT-0 when the data was transferred to magnetic tape. If X ≠ 0 control will be transferred to ZZZZZ.

Y. The octal digit Y determines which MT unit will be selected. MT units are specified by the same digits used in the standard 1103 MT commands.

Z. The V-address of the parameter word (ZZZZZ) specifies the address to which control will be transferred at the conclusion of the routine (see X above).

Operating Instructions (to be followed when TST-0 is used as a service routine)

1. Manually enter the parameter word in Q.

2. Set PAK to 40007b and start.

3. Computation will halt after a successful transfer with PAK set as specified (see "Parameter Word" above).

Programming Instructions

1. Use the RJ instruction 37 40020 40007b to enter TST-0. The cell immediately following the RJ instruction must contain the parameter word.

2. After successful transfer control will be transferred to the cell specified by the parameter word.

Alarm Conditions

If the sum test fails ALR-1 is entered and "TST-0 XXXXX" is printed on the flexowriter. The address is insignificant.

Starting after the alarm halt causes a rewind of the tape and another transfer of the same data from MT.

Restore

If, at any time during its operation, TST-0 is interrupted (or after an alarm print), PAK set to 40040b and the machine started, the routine will

1. Rewind MT (if this had been specified)

2. Restore (ES), (A), and (Q)

3. Transfer control to the TST-0 exit instruction.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

Binary Card Read-In Routine (revised)

Specifications

Identification Tag:	CRI-1
Type:	Service Routine (with subroutine entrance)
Special Storage:	The constant pool and temporary storage pool are not used by this routine
Service Entrance:	40017b
Program Entrance:	40017b
Program Exit:	40020b
Alarm Exit:	The alarm exit is used by this routine

Coded by:	R. Beach	October 26, 1955
Code Checked by:	R. Beach	October 26, 1955
Machine Checked by:	R. Beach	October 26, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. F. Bauer	December 9, 1955

PX 71900-8-103

Description

This input routine reads binary punched cards produced by the binary card dump (MDP-1 revised) at full card reader speed. Once activated it continues to read cards, ignoring blank cards, until it has read a card containing a 12 punch in column 80. The routine loads the memory as directed by the address appearing on the card, and checks the sum of the data read in against the sum stored on the card, for each card.

The routine stores ES and bootstraps itself into ES for execution. After execution, it restores ES.

Operating Instructions

1. When routine is used as a service routine set PAK to 40017b and start.

2. When routine is used as a subroutine, enter routine with 37 40020 40017b

3. Card positioning

   A card must be positioned on the read side of the Bull or a "no information" fault will occur on the first cycle. The fault, however, positions the card so one may simply start at 00103b.

4. To restore ES at any time, start at 40040b. This start may also be used to end a read in for which the 12 punch in column 80 of the last card was omitted.

Alarm Conditions

If the sum of data read in fails to check against the sum appearing on the card, ALR-1 is activated to print.

alarm 00102 000000000000 1100123vvvvv xxxxxxxxxxxx

where vvvvv is the address on the card. Q is not significant.

Starting ignores the alarm.

THE RAMO-WOOLDRIDGE CORPORATION
Los Angeles 45, California

The Ferranti Input Routine (revised)

Specifications

Identification Tag:	FRI-0
Type:	Service routine (with subroutine entrance)
Special Storage:	The constant and temporary storage pools are not used by this routine
Service Entrance:	Address 40001b
Program Entrance:	Address 40001b
Program Exit:	Address 40020b
Alarm Exit:	The alarm exit is used by this routine

Coded by:	R. Beach	May 18, 1955
Code Checked by:	R. Summers	May 19, 1955
Machine Checked by:	R. Beach	August 4, 1955
Revised by:	C. Koos	December 1, 1955
Approved by:	W. Bauer	December 9, 1955

Description

I. General

This routine is designed to read, by means of the Ferranti reader, seven-level bioctal tape prepared as described below. The routine reads in paper tape at the full speed of the Ferranti with only short hesitation when a check or insert address is encountered.

If desired, the tape may contain a check sum to be tested for agreement with the computed sum of the data read-in. The routine will read data into any ES or MD cell although the reading of information into certain drum cells (as described in detail below) will result in abnormal operation.

The routine stores the contents of ES on MD at addresses 76000b through 77777b and then transfers itself to ES. It sums itself (in ES) and checks the sum against the correct sum (stored on MD).

The Ferranti reader is started in the free running mode and the routine proceeds to read tape and process the information contained on the tape in the same manner as does the ERA photoelectric reader (for exceptions, see II. 3 and 4).

Each word to be transferred to memory is summed as it is read in from tape. Words which are to be read into ES are first stored in the MD image of ES (76000b thru 77777b).

During operation all words are read into ES from the tape and a block transfer to MD is made when (1) ES has been filled with data (that is, when 924 words have been read in); (2) an insert address appears on the tape; or (3) the "end of tape" seven-level combination has been read in (see II. 4).

The reader is stopped before making the transfer and is started again after the transfer has been completed in the first two cases; in the last case, the reader is stopped, ES is restored from the MD image and control is transferred to the exit.

The reader is also halted when a check address appears on the tape. If no check sum test (see II. 3) is to be made after a successful check address test the reader is started immediately; if the check sum test is specified the reader is started after the test is made and the sum determined to be correct.

The routine does not prevent read in to addresses 76000b thru 77777b nor to those cells used by the routine for its own operations.

II. Requirements for Tape Preparation

1. The first word on a tape must be an insert address.

2. Check addresses should be used, although FRI-0 will operate without them. A check address immediately following an insert address must be the same as the insert address.

3. For a check sum test the following four words must appear on the tape at the point where the sum is to be tested:

   a. Insert address 75202b

   b. High order 36 bits of check sum

   c. Low order 36 bits of check sum

   d. Check address 75204b

Operating Instructions (to be followed when the routine is used as a service routine)

1. Set PAK to 40001b and start.

2. Computation will halt with the MS instruction 56 00000 40001b at the completion of the read in.

Programming Instructions (to be followed when the routine is used as a subroutine)

1. Enter the routine with the RJ instruction 37 40020 40001b

2. Control is returned to the cell immediately following the RJ instruction as soon as an "end of block" punch is reached on the tape.

Alarm Conditions

1. No "end of tape" punch. This condition is indicated by the tape running completely out of the Ferranti reader. When such a condition occurs the operator should.

   a. Master clear

   b. Set PAK to 00074b and start

   c. When computation halts (when a service entry was used) with the MS instruction 56 00000 40001b the machine will be returned to its original state and the data read from the tape will be properly stored.

      If a program entry was used control will be transferred to the proper cell in the main program.

2. FRI-0 not transferred to ES correctly. If ALR-1 prints "FRI-0 xxxxx and (A) and (Q), the sum of the program transferred to ES has failed to check. Starting at this point transfers FRI-0 to ES again.

   A second failure indicates that FRI-0 is not on the drum correctly and should be restored.

3. Check address failure. If ALR-1 prints "ALAR C" and (A) and (Q), a check address has failed. In the alarm print (A_R) is the address of the next

cell to be loaded and (Q) is the check address that was read in from paper tape.

Starting at this time will cause the machine to ignore the failure and operation will continue normally.

4. Check sum failure. If ALR-1 prints "ALAR M" and (A) and (Q), the check sum on the tape has failed to agree with the computed sum. The computed sum is in A at the time the alarm print occurs.

   Starting at this point will cause the routine to ignore the failure and to begin to read in the tape again.

   If at any time (ES) need to be restored from its image, starting at 40040b will transfer the image to ES and transfer control to the FRI-0 exit.

5. And "end of tape" (or "end of block") punch must be present on the tape to halt read in. This consists of seventh level punches in two consecutive frames on the tape at the point where the read in is to be stopped. This seventh level combination acts as a signal to FRI-0 to restore (ES) and stop the Ferranti reader. It is compatible with the ERA photoelectric reader in that it is an illegal combination which halts the ERA reader.

DIVISION San Diego

MODEL All

REPORT ZM 490

DATE 8/4/54

Revised: 12/15/55

TITLE

F L I P

A FLOATING POINT SUBROUTINE SYSTEM

FOR

ERA 1103 AND 1103A COMPUTER

PART I DESCRIPTION

PREPARED BY CHARLES J. SWIFT

GROUP Digital Computing Lab

REFERENCE

CHECKED BY Members of: Digital Computing Lab

APPROVED BY [signature]

NO. OF PAGES 31

NO. OF DIAGRAMS 0

REVISIONS

NO.	DATE	BY	CHANGE	PAGES AFFECTED

Preface to Revised Report

Flip III is faster than Flip 1. Several restrictions in Flip I which were discovered over a period of time were removed. These occurred in Commands 40, 42, 44 and 51. Division by zero causes an alarm halt in Flip III, but division by an unnormalized number may still cause a computer fault as in Flip I.

The magnetic tape readin has been changed in two respects. The sum check print is now "FLIP OK". If the 1103 is restarted at this point the entire memory except that part already sum checked is cleared to zero and "clear" is printed out. Except when other information is already in the memory, this step should always be included.

A new method of loading or "ACTIVATING" Flip is available in Flip III.

A great deal of this report has been completely rewritten to render it more comprehensible on first reading. Many members of the Digital Computing Laboratory have assisted in this. All coding has been relegated to Part II of this report which will be issued shortly.

Flip is the result of suggestions, coding and checking by so many individuals that it is unfortunately impossible to give specific credits for the various contributions.

CONTENTS

FLIP III

I DESCRIPTION:

Flip III is an interpretive system for the 1103 and 1103A, useful for a large class of problems. Using the Interpret command, the machine commands are augmented by 58 "pseudo commands" which can be used as though they were regular machine commands. These enable the inclusion of floating point arithmetic, complex arithmetic, and special functions which include an integration step for differential equations. Also certain features, such as index registers and tracing (to be described in detail later), are built into these pseudo commands. Conversions to and from floating point representation enable the programmer to use the convenience of the Flip input and output while doing the "core" of the problem in fixed point for speed.

Since this system augments the basic commands, knowledge of most of them is necessary.

The speed of operation ranges from 3.9 milliseconds for a floating point multiplication to 4.3 milliseconds for floating point addition. Transcendental functions compare closely to fixed point operations in regards to timing.

II NUMBER REPRESENTATION:

All input and output occurs in a floating point decimal form. However, coders probably should know the internal representation. Let a number N be given by $N = q \cdot 2^p$; $\frac{1}{2} \leq |q| < 1$; $-127 \leq p \leq 127$

The number q occupies the first 28 bits of the cell and the number p the last eight, each beginning with a sign bit. Negative q's and p's are represented by complements. Zero is represented by a true machine zero.

An unnormalized two cell representation occurs in the "floating to fixed" and "fixed to floating" commands. Detailed description of it occurs under those commands.

III COMMAND STRUCTURES:

A. One address commands, octal form: (Commands 55, 56, 57) *

1, 4 } Interpret command code

C, C } Pseudo command code

J — Command Parameter

N, N } Command Counter

V, V, V, V, V } Regular machine address, not to be index modified

B. Two address Commands, octal form. (All other commands)

1, 4 } Interpret command code

C, C } Pseudo command code

X, X, X, X } Index counter tags and first address, x.

Y, Y, Y, Y } Index counter tags and second address, y.

The first octal digit of these addresses consists of three Binary bits. The first of these bits, if a one, causes the first index counter, b_1, to be added to the address during command execution. (Not in its memory position). The second bit does likewise with the second index

*RESTRICTION: See page 56-2.

counter, b_2. Both bits may be one, in which case both counters are added.

Since there are only ten remaining address bits, only 1024 cells can be directly addressed. If. however, $b_1 \cdot 2^{10}$ and $b \cdot 2^{11}$, all cells in the 4096 1103A high speed store may be properly addressed. b_1 occuppies cell 01776 and b_2 occuppies cell 01777.

IV REGISTERS

Flip commands refer only to the right 36 bits of the accumulator, the other bits of A and all of Q being lost. These 36 bits form a register called R which is restored by all Flip index or threshold jump commands. If R is to be used as an operand, it may be addressed, using 1774 as the address, instead of 20000 as used in regular machine commands.

V COMMAND LIST

A. Floating Point Arithmetic Commands.

The two octal digits of these commands perform separate functions. Recognizing this fact will facilitate learning them.

(FLIP)

Code	Symbolic Operation	Name
00	$y + x \rightarrow y, R$	Replace Add
01	$R + x \rightarrow y, R$	Add and transmit
02	$y + x \rightarrow R$	Add
03	$R + y + x \rightarrow R$	Accumulate Add
04	$y - x \rightarrow y, R$	Replace subtract
05	$R - x \rightarrow y, R$	Subtract and transmit
06	$y - x \rightarrow R$	Subtract
07	$R + y - x \rightarrow R$	Accumulate add and subtract
10	$y + R \cdot x \rightarrow y, R$	Replace add a product

11	$(1+x) \cdot R \rightarrow y, R$	Increment and Multiply
12, 13	$y+R \cdot x \rightarrow R$	Positive Polynomial
14	$y - R \cdot x \rightarrow y, R$	Replace add a negative product
15	$(1 - x) \cdot R \rightarrow y, R$	Decrement and Negative Multiply
16, 17	$y - R \cdot x \rightarrow R$	Alternating Polynomial
20	$y \cdot x \rightarrow y, R$	Replace multiply
21	$R \cdot x \rightarrow y, R$	Multiply and Transmit
22	$y \cdot x \rightarrow R$	Multiply
23	$R + y \cdot x \rightarrow R$	Accumulate Multiply
24	$-y \cdot x \rightarrow y, R$	Negative replace Multiply
25	$-R \cdot x \rightarrow y, R$	Negative multiply and transmit
26	$-y \cdot x \rightarrow R$	Negative Multiply
27	$R - y \cdot x \rightarrow R$	Accumulate negative Multiply
30	$y \div x \rightarrow y, R$	Replace Divide
31	$R \div x \rightarrow y, R$	Divide and transmit
32	$y \div x \rightarrow R$	Divide
33	$R + (y \div x) \rightarrow R$	Accumulate Divide
34	$-y \div x \rightarrow y, R$	Negative replace Divide
35	$-R \div x \rightarrow y, R$	Negative Divide and transmit
36	$-y \div x \rightarrow R$	Negative Divide
37	$R - (y \div x) \rightarrow R$	Accumulate negative divide

B. Other Commands.

This list is designed for quick reference. For detailed descriptions of most of these commands see subsequent pages. Some of these commands, normally operated in high speed storage, are not needed in many programs. These are not included in high speed storage unless specified and then require extra storage. See section (VII) for details of assigning storage.

COMMAND CODE	EXTRA CELLS (DECIMAL)	DESCRIPTION
40	NONE	Index jump, counter b_1.
41	NONE	Jump if $\|(x)\| > \|(R)\|$
42	NONE	N $(x,y) \rightarrow R$
43	57	Flexo writer input conversion
44	NONE	Index jump, counter b_2.
45	NONE	Jump if $\|(x)\| < \|(R)\|$
46	31	Integrate Differential equations
47		Not used
50	17	$\sqrt{x} \rightarrow y$
51	NONE	$R \rightarrow N$ (x,y)
52	56	Typewriter output
53	2+56*	Paper tape data output
54	31	$\log_e x \rightarrow y$
55	NONE	Charactron output
56	NONE	punched card output
57	NONE	punched card input
60	49	$\sin x \rightarrow y$
61	2+49*	$\cos x \rightarrow y$
62	45	$\tan^{-1} x \rightarrow y$
63	2+45*	$\cot^{-1} x \rightarrow y$
64	NONE	$J_o(x) \rightarrow y$
65	NONE	$J_o(x) \rightarrow y$; $Y_o(x) \rightarrow 00037$

66	NONE	$J_1(x) \rightarrow y$
67	NONE	$J_1(x) \rightarrow y$; $Y_1(x) \rightarrow 00037$
70	27	$e^x \rightarrow y$
71		Not used
72	42	$Y+X \rightarrow S$ (Complex Arithmetic)
73	42	$Y-X \rightarrow S$ (Complex Arithmetic)
74	42	$Y \cdot X \rightarrow S$ (Complex Arithmetic)
75	42	$Y \div X \rightarrow S$ (Complex Arithmetic)
76		Not used
77	26	Trace (Magnetic tape version)

*For these commands see restrictions under detailed descriptions that follow.

VI STORAGE

CELLS	USE
00002 to 00037 74000 to 75777	TEMPORARY STORAGE
00001 01477 to 01777 70000 to 73777 76000 to 77777	ROUTINE
00040 to 00077	PERMANENT CONSTANTS

As used at Convair, Flip is stored permanently on magnetic tape. A manually initiated operation transfers it to magnetic drum locations. The transfer to high speed storage is called "activating" and is programmed.

(See next section).

The permanent constant pool is available to the programmer, of course, and should never be changed. It is given below.

CONSTANT POOL

00040	00	00000	00000	Zero	
00041	00	00000	00002	2	
00042	61	00000	00045	and Flex Code Carriage Return	
00043	00	00000	00003	3.	
00044	00	00000	00004	4. and flex code space	
00045	00	00000	00037	Flex code 0 dec. 31.	
00046	00	00000	00052	Flex code 1 dec. 42.	
00047	00	00000	00074	Flex code 2 dec. 60.	
00050	00	00000	00070	Flex code 3 dec. 56.	
00051	00	00000	00064	Flex code 4 dec. 52.	
00052	00	00000	00062	Flex code 5 dec. 50.	
00053	00	00000	00066	Flex code 6 dec. 54.	
00054	00	00000	00072	Flex code 7 dec. 58.	
00055	00	00000	00060	Flex code 8 dec. 48.	
00056	00	00000	00033	Flex code 9 dec. 27.	
00057	00	00000	00013	11.	
00060	00	00000	00012	10.	
00061	00	00000	00056	Flex code minus "-"	
00062	31	10375	52421	$\pi/4$	(35)
00063	31	46314	63146	10^{-1}	(38)
00064	00	00000	00077	six-bit extractor	
00065	21	67643	24177	degree to radian	(40)
00066	20	00000	00000	.5 decimal	(35)

00067	00	00000	00007	7.
00070	37	77777	77777	.9̇ (35)
00071	00	77777	00000	Extractor u
00072	00	00000	77777	Extractor v
00073	00	00001	00000	Advance of u
00074	00	00000	00001	Advance of v
00075	00	00001	00001	Advance of u and v
00076	00	07777	07777	u and v 4-octal-digit extractor
00077	00	00000	00110	72.

VII ACTIVATING:

Flip is readied for use in high speed storage by "activating" it. A return jump followed by parameters accomplishes this. The parameters specify the extra storage space for all extra storage commands to be used. A zero follows the parameters, followed by the next command.

The return jump is 37 70160 70140

The parameters are:

C
C } Flip pseudo command code for an extra storage command

0
0
0 Must be zeroes
0
0

V
V
V } Location (usually high speed storage) for the extra storage required.
V
V

The parameters must be followed by a zero cell as a flag.

NOTE: Previous methods used to activate Flip are now obsolete but will still work.

VIII OVERFLOW AND OTHER SPECIAL NOTES.

If overflow of the exponent or other impossible conditions arise, an alarm print occurs. The command address and the content of the two operand cells is printed.

All operations incur the possibility of round off error. Because of this, exact equality of two floating point numbers will seldom occur no matter how simple a calculation was used to obtain them. Therefore, equality tests should not be used and threshold jumps should always provide a tolerance.

For all commands which store the result in R and not elsewhere, the accumulator contains D (R). For all other cases, the left side of the accumulator contains the sign of the exponent (characteristic) of the number.

For debugging, cells 01736, 01776, 01777 and 00002 to 00014 usually contain all the information required to tell what Flip is doing.

The "Alarm, Octal and Flexprint" routine (CV-115) is included in Flip. An alarm in Flip prints out the instruction, (R), (x), (y), b_1 and b_2 in two lines, preceded by the word "ALARM".

INSPECT AND CHANGE ROUTINE

I. Operating Instructions

1. Set PAK to 70700. Start. The 1103 clears A and Q and halts.

2. Set ∅ (Q) to a desired address and (optional) set (R) to the new content desired for that address. Start. The 1103 prints change, the address, and the present content. It halts with the old content in Q and the new content in R.

3a. Set or change (R) if desired. Start. The 1103 stores (R) in the cell, prints out this new content, and halts at the same point as step 1.

OPTION

3b. If, after step 2, it is desired not to change the cell content, send control to 40000. The 1103 prints out no change and halts at the same point as step 1. (At this point, (40000) has its original value, unless it was the cell changed.)

II. Specifications

Drum address 70700 through 70773 (60)

Number of commands to be modified - 60

No standard constants or temporaries are used.

Entrance 70700 (or 01000). This routine is available either with addresses starting in 70700 or 01000. It can be used to modify cell 00000, providing no other cell is subsequently modified.

FLIP CARD TO PAPER TAPE ROUTINE

DESCRIPTION:

This routine reads blocks of numbers from Flip Cards and punches them in binary form on a bioctal tape. Each block of N numbers is placed in consecutive cells starting with address A. The first card of each block has A (octal) in columns one to five and N (decimal) in columns six and seven. The last card of each block is filled out with zeros. A blank card following any block causes all input cards to advance into the receiving hopper and halts the program.

INSTRUCTIONS:

1. Read Flip onto the magnetic drum.
2. Start at 71724.

LOCATION:

71724 to 71775

FLIP MAGNETIC TAPE STORAGE ROUTINE

Location 71500 to 71565

This Routine stores a region of the internal memory on magnetic tape in a self-reloading form. It can be used automatically or manually. Cells 00000 to 00037 and part of basic Flip are used as temporaries, the latter being restored. No check sum is provided. These instructions assume a locater block and a normally rewound tape.

A-Transfer to Magnetic Tape.

1. Manual Operation

   a. Place an advance tape command, 66j n in Q.

   b. Place the first address F and the last address L in u and v of A.

   c. Start at 71500. The tape will advance MT n+1 blocks, store information on M blocks*, rewind the tape and final stop.

2. Automatic Operation

   a & b. Same as manual.

   c. Execute the command 37 71562 71501. Results are the same as manual except that no stop occurs, control returning to the next line.

*NOTE: M = Integral part of $\left(\frac{L-F+1}{28}+2\right)$

B-Transfer from MT.

Use the locator block with n in A. If the transfer to magnetic tape was manual, a final stop occurs. If not, control is transfered to the point at which the dump was made.

DOUBLE ENTRY TABLE INTERPOLATION AND LOOKUP
(DETAIL)

Using Flip

I SPECIFICATIONS:

Storage: The sub-routine is stored in cells 01300 to 01476. Flip and all Flip temporaries are needed in ES. Tables of any size are normally on MD.

Time: Roughly .3 seconds plus .3 seconds for each function evaluated.

Scheme used: A function $F(x,y) = \sum_{i=0}^{2} \sum_{j=0}^{2} c_{ij} x^i y^j$ is passed through a 9 point square array of points, chosen from the given points in the x-y plane.

II DESCRIPTION:

If from one to five quantities $\underline{Z}^\nu$ are each functions of two variables $\underline{x}$ and $\underline{y}$ and are given in the form of tabled values, (not necessarily at equal distant points), then this sub-routine evaluates these functions by interpolation. The nearest tabled values of x and y (designated $X_{(r)}$ and $Y_{(s)}$) are first found by the sub-routine.. Then these points and their nearest neighbors on each side are used to give a nine point interpolation in each table of corresponding Z^ν values. If x and/or y lie nearest to or beyond an endpoint, the three endpoints are used. Surfaces of the form: $Z^\nu = \sum_{i=0}^{2} \sum_{j=0}^{2} c_{ij} x^i y^j$ are passed through the nine tabular points and evaluated at the given point (x,y) for each Z^ν.

III LOCATION AND STORAGE:

This routine operates in cells 01300 to 01466. It uses cells 01300 to 01307 for parameters, and cells 00002 to 00037 for temporaries. It is entered by the command 37 01471 01310.

IV STORAGE:

A. Arguments.

(00036) = X

(00037) = Y

B. Parameters:

Cell	CC	(U)	(V)
01300	00	D_x	D_y
01301	00	i	j
01302	00	n_x	n_y
01303	00	D_1	L_1
01304	00	D_2	L_2
01305	40	D_3	L_3
01306			
01307			

D_x = location of table of X values.

D_y = location of table of Y values.

i = Index of current location in X table. (Start at zero)

j = index of current location in Y table. (Start at zero)

n_x = number of values in X table.

n_y = number of values in Y table.

D_ν = location of table of Z^ν

L_ν = storage location for Z^ν (X,Y)

NOTE: The last parameter is indicated by a 40 in the first two octal digits. (Cell 01305 in above example)

C. X & Y TABLES:

Cell	Content
D_x	X_1
$D_x + 1$	X_2
$D + n_x - 1$	X_{n_x}

D. Z TABLES:

Cell	Content
D_ν	$Z^\nu\ (X_1 Y_1)$
$D_\nu + 1$	$Z^\nu\ (X_2 Y_1)$
$D_\nu + n_x - 1$	$Z^\nu\ (X_{n_x} Y_1)$
$D_\nu + n_x$	$Z^\nu\ (X_2 Y_1)$
$D_\nu\ (n_x - 1)(n_y - 1)$	$Z^\nu\ (X_{n_x}\ Y_{n_y})$

E. RESULTS:

$(L_\nu) = Z^\nu(X, Y)$ for $\nu = 1, 2, 3$, etc.

ROUTINE TO LOAD FLIP ON MAGNETIC TAPE

INSTRUCTIONS

1. Rewind tape
2. Place 66 jn - in Q. (n is the number used in A_r to locate Flip).
3. Start 40000 and restart until dump is made.

NOTES:

1. If Flip has just been read from MT or paper tape "NO GO'S" should not be printed. If changes have been made they will be. This routine corrects the sum adjuster after two "NO GO" prints.
2. To dump a new paper tape stop at 56 10000 00100. Use the Flip bioctal tape dump and dump cells 00000 to 00277, 40000 to 40037, 70000 to 73777 and 76000 to 77777.

Flip Bioctal Tape Dump

Location: 71000 to 71105

Permanent Constants used: (00074) = 1, (00073) = 2^{15} (00040) = 0

Tempories (not restored): 00000, 00002 to 00037, A and Q.

Manual Operation

1. Start at PAK = 71000 with Parameter word in Q:

   00 fffff 11111

   Where fffff is the address of the first word, 11111 is the address of the last word.

2. After operation, the routine stops with PAK = 71000.

Automatic Operation

1. Enter with command: 37 71007 71003, followed by parameter words. The last parameter word is followed by the next instruction.

Restrictions and Details

1. The first and last addresses cannot be identical. (If so, that complete storage class is dumped as a closed set).

2. A check address, leader and insert address are automatically punched after every 400 (octal) words.

3. Cells 00001 and 00040 to 01777 are not disturbed at any time.

4. ES and MD are treated as closed sets.

INDEX JUMPS

COMMAND CODES

40 Index counter b_1 conditional jump

44 Index counter b_2 conditional jump

The first address refers to a fixed point Integer, m. If b_i is greater than or equal to m, b_i is cleared to zero. If not, b_i is increased by one and a jump is made to the second address. If b_i is originally zero and this jump command returns control to some previous point, the intervening commands will be executed m+1 times with tagged addresses properly modified. b_1 is in cell 01776 and b_2 is in cell 01777. Both are fixed point integers and are initially zero.

FIXED POINT CONVERSION COMMANDS

COMMAND CODES

42 N (x,y) → R

51 R → N (y,x)

In order to convert a number N from fixed point to floating point, or vice versa, its scale factor s must be specified. The notation N (y,x) means that a fixed point number is stored in the first cell and the quantity 35-s is stored in the last eight bits of the second cell. * Command 42 produces the Flip floating point form in R. Command 51 takes the Flip floating point form from R and produces the fixed point form in y. The quantity 35-s must be given in both cases.

RESTRICTION:

The cell containing 35-s in the last eight bits may not be zero even though 35-s may be zero.

*NOTE: These commands use the quantity 35-s because that is more convenient to compute if it is to be computed instead of prestored. It can be seen that N (x,y) is similar to the Flip floating point form except that the number is not normalized and the exponent of 2 is stored in a separate cell.

FLIP

Flexowriter Input Conversion Routine

Command Code: 43

Number of Cells: 58

I Description

This command converts a two-word input representation of a number to the normalized FLIP form and stores the result in y, but not in R. The two words are (x) and (x + 1). When taken 6 bits at a time they are 12 flexowriter character codes. If the number represented is: -

$$N = q' \cdot 10^{p'}; \ 0.1 \leq q' < 1$$

these flexowriter characters are: -

1	2	3	4	5	6	7	8	9	10	11	12
±	q	q	q	q	q	q	q	±	p	p	

The decimal point (not present) precedes character number 2. The sign positions are considered plus unless occuppied by a minus sign (octal 56). The twelfth digit is immaterial and will usually be a carriage return.

If the number represented exceeds 2^{127} in absolute magnitude, or if any digit q or p is not a flexowriter numerical digit, an alarm halt will occur.

INTEGRATION SUBROUTINE

Command Code: 46

Drum Location: 71600 to 71721

Number of Cells in ES: 31 (decimal)

Temporaries Used: 00002 to 00037

I. INTRODUCTION:

This command is a revision of Gulp (ZM491-CN003). It uses the Gill-Runge-Kutte Method* to integrate a set of first order differential equations:

$$y_i' = \frac{dy_i}{dx} = f_i\ (y_1\ ,\ y_2\ \ldots\ldots y_n, x)\ (1)$$

from

$$x \text{ to } x + \Delta x.$$

II. SPECIFICATIONS:

A subroutine to compute these functions must be supplied starting in a cell specified by the address y. The number of equations minus one (n-1), the increment Δx, and the quantities $y_1 \ldots y_n$, $y_1' \ldots y_n'$, $q_1 \ldots q_n$ are stored in consecutive cells starting with the address x. All the numbers are in Flip form except n-1 which is in fixed form at 2^0. The q's are zero or infinitesmals except within the command.

III. RESTRICTIONS AND NOTES:

1. The Flip counters in 01776 and 01777 are used but restored for the main routine. If used by the derivatives subroutine, they must be restored before its exit.

2. The number of variables, n, must be at least two.

3. If the independent variable x occurs on the right side of any of equations (1), it must be treated as a dependent variable with a derivative of unity.

4. The subroutine to compute the functions must be entered in its third cell and must exit by a jump from its second cell. ("Standard Form")

5. If any derivative is zero, most of the integration cycle is skipped, hence dummy variables may economically be included for later use.

6. The quantities n-1 and Δx may be varied at any time. However, the locations of the derivatives, q's, and the derivative subroutine are determined by the values that the parameters have the first time the subroutine is used. They remain fixed thereafter.

7. The Gill Method will only integrate smooth functions. If inequalities cause changes (breaks) in the functions, these must be taken care of between integration cycles. New derivatives must be computed before the next intergration command is given whenever the functions are so altered.

8. The truncation error is proportional to the fifth power of Δx. A test integration of the sine and cosine functions using $\Delta x = 22\frac{1}{2}^{\circ}$ caused a total error of approximately .15%, after sixty integration cycles. Due to the use of an unrounded floating point system, the double precision effect of the Gill Method does not work here, and round-off error may build up.

*Gill, S., Cambridge Philosophical Society Proceedings 47, 96-108 (1950)

Square Root Subroutine

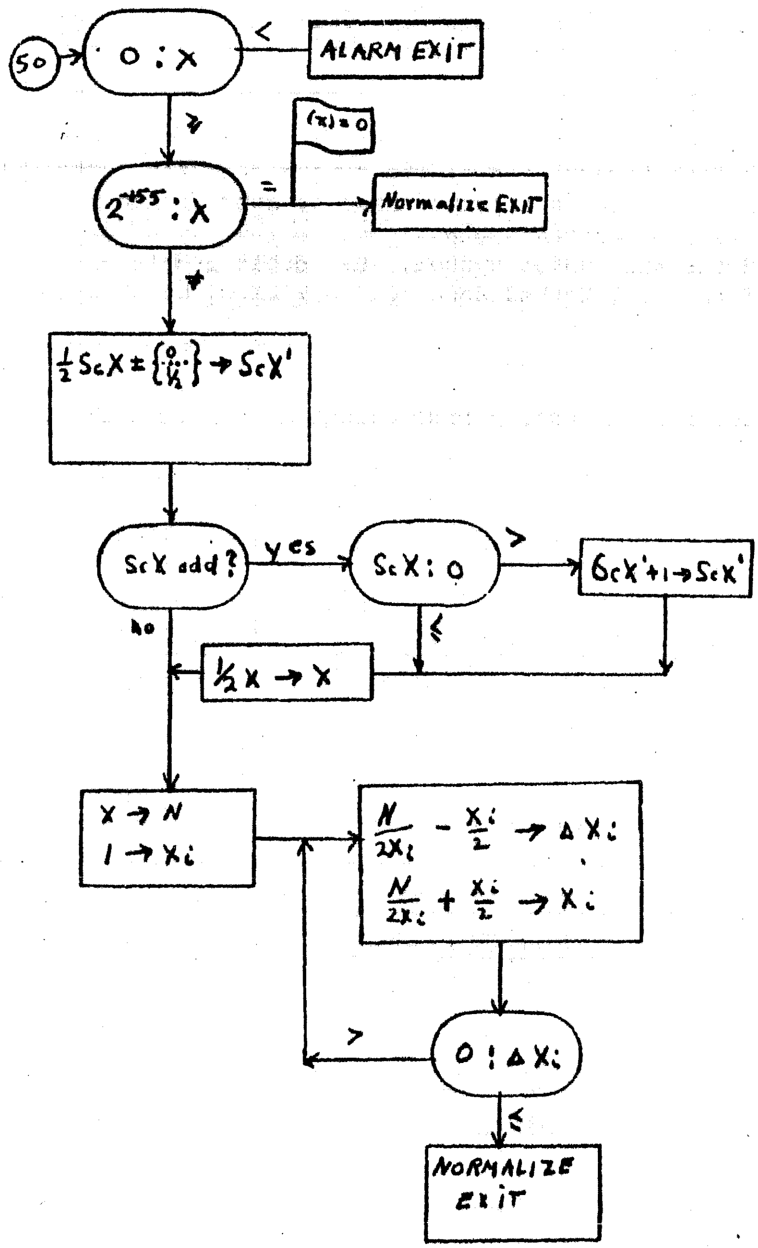
Command Code: 50

Number of Cells: 17

I **Description**

This command computes the square root of (x) and stores it in y and R. with full accuracy. An Alarm Halt occurs for negative arguments.

II **Flow Chart**



FLIP

Print and Punch Subroutine

Print Command Code: 52

Punch Command Code: 53

Number of Cells: 58

I Description

These two commands convert (x) to a floating decimal form and print it on the flexowriter or punch a flexowriter tape. For these commands y is immaterial. If the Print subroutine is used, the Punch subroutine must be also specified to the loader. If the location of the first cell of the print subroutine is y, that of the punch subroutine must be y + 2.

Fifteen digits are printed or punched as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
±	q	.	q	q	q	q	q	q	sp	±	p	p	sp	sp

Where the number (in floating decimal form) is

$$n = q \cdot 10^p$$

No carriage return is included.

FLIP

Logarithm Subroutine

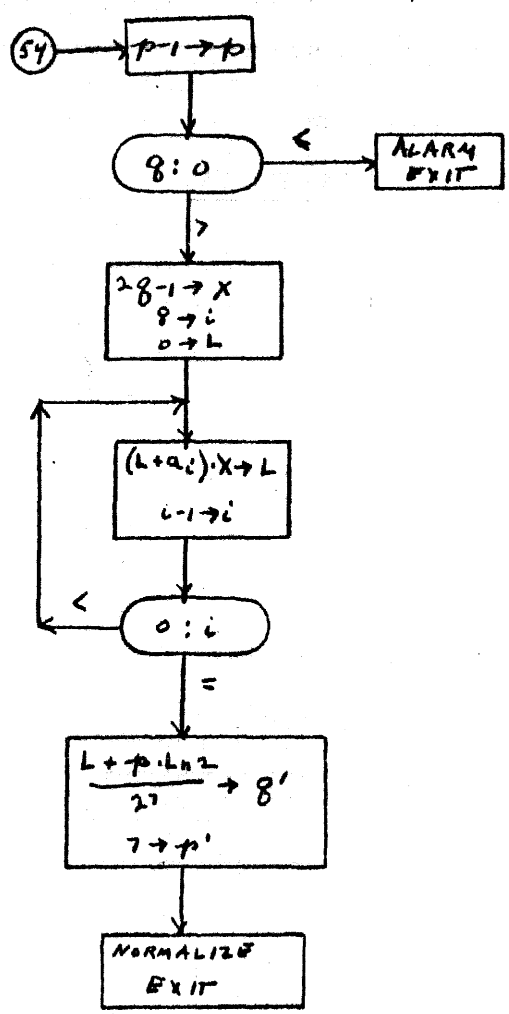
Command Code: 54

Number of Cells: 31

I Description

This command computes $\log_e (x)$ and stores the result in y and R. A polynomial approximation is used(1) which gives a maximum error of the order of $3 \cdot 10^{-8}$.

II Flow Chart



$$(x) = q \cdot 2^{p}$$
$$X = 2q - 1$$
$$Ln\,(x) = (p-1)\,Ln\,2 + Ln\,(X+1)$$
$$= q' \cdot 2^{p'}\ ;\ p' = 7$$
$$Ln\,(X+1) = a_1 X + a_2 X^2 + \cdots a_8 X^8$$

(1) See sheet 56, Approximations in Numerical Analysis, a publication of the Rand Corporation

FLIP CHARACTRON OUTPUT

Command Code : 55

Drum Location : 72444 to 72777

Temporaries Used : 00002 to 00037 and 74100 to 75777

Do not specify this to the loader

I. GENERAL DESCRIPTION

This command converts a specified block of numbers from flip form to a floating decimal form and prints them on film with the charactron. Up to 992 numbers can be handled with one command at speeds up to 66 numbers per second. It is intended to cover both regular output and code checking output requirements.

The general format of each page has a heading, a page number at the bottom, and positions for 128 numbers in four columns, 32 in each column. The way in which these positions are filled is described under the various options below.

II. COMMAND FORM

1455 JNN VVVVV

J specifies various options. Let the last five bits of NN form a number m ($0 \leq m \leq 31$) and let the first bit of NN be the number ℓ where ℓ is zero or one. Then the number of words converted and printed is

$$N = m(1 + 31\ell)$$

III. BASIC OPTION (J = 0)

Each page is titled Flip Output and pages are numbered consecutively from 001 to 999. Consecutive numbers follow each other vertically down the four columns. Each new use of the output command causes a new column to start, and leaves any remainder of the previous column blank. If the block specified by any command will not go onto the remainder of the current page, a new page is started.

IV. J OPTIONS

1. J odd. — This causes a new page to be started with this information.
2. J = 2 or 3. — This causes the numbers to follow immediately after the preceding numbers instead of starting a new column.
3. J = 4 or 5. — This causes the numbers to follow each other in horizontal rows across the page. Each new command causes the remainder of the last line and one blank line to be skipped.
4. J = 6 or 7. — This is similar to 4 or 5 except that no blank line is left before this information.

V. SPECIAL NOTES

a. Horizontal and vertical format commands cannot be interspersed without danger of printing on top of other information.

b. The title consists of 48 charactron characters occupying consecutive positions from left to right in cells 72746 to 72755. These may be replaced with another title if desired. Use octal 77 for blank positions.

c. The position on the page of the last number printed is described by a number N stored in cell 72702 scaled at 2^0. N starts at 1 in the upper left hand corner and goes to 128 in the lower right hand corner. It can be manipulated to give special formats.

d. The page number of the preceding page is always stored in cell 72703. It is stored with a scale factor of $10^{-8} \cdot 2^{36}$ and without a sign bit. The constant $10^{-8} \cdot 2^{36}$ is stored in cell 72716.

e. In order to print a number of words greater than 32 but not a multiple of 32, two commands must be used. The second one will have less than 32 words and use j = 2 or 6.

VI. TIMING

If the charactron time delays do not hold up the 1103 (see below), the total time consumed by each Flip charactron output command is

$$T = .31 \text{ sec.} + n \; .013 \text{ sec.}$$

where n is the number of output words. There are two charactron time delays

$$d_1 = .4 \text{ sec.}$$

$$d_2 = 2.6 \text{ sec.}$$

that can occur before resume pulses are sent back from the charactron. These delays effect the output as follows:

1. If at least $d_1 + d_2$ seconds of other computation occur for each page and not more than one page (128 numbers) is printed with each command, no delays occur.

2. If at least $d_1 + d_2$ seconds of other computation occurs for each page and not more than two pages (256 numbers) are printed with each command, add d_1 to the execution time for each command.

3. For all other cases, the effect of d_2 enters. The maximum continuous output rate is $d_1 + d_2$ seconds per page.

4. For the non-developing camera, d_2 is zero and d_1 may vary.

NOTE: <u>RESTRICTION</u>: See page 56-2 for a restriction on this command.

FLIP CARD ROUTINES

Punch Command Code: 56

Read Command Code: 57

Storage Locations 73000 to 73312 inc.

73400 to 73657 inc.

73771 to 73777 inc.

I. Description:

These commands read or punch cards which contain numbers in a floating decimal point form. Conversions from or to the Flip floating point form are performed during each card cycle. Either of two standard card forms can be used as described under the options below. These commands are one-address, non-indexable commands of the form:

14 CC J NN VVVVV

Where V is the first storage address (1), and N is the octal number of cards, N can vary from 00 to 77 octal. A 12 punch in the first column of any card being read will override N and terminate reading with that card. Whether this has occurred can be tested with the sign of (00004). This will be minus 1 if the last card to be read had a 12 punch in column 1; otherwise it will be zero. J controls the card form as indicated below.

(1). RESTRICTION: (Not applicable to the 1103A). Flip treats all commands as two address commands prior to inspecting the command code. If the index counter modification which occurs here leads to an illegal address, a fault will ensue.

EXAMPLE: The command 1456 001 47770 has a "y address" of 1770 with tags for both index counters. If either counter is 10 (octal) or more, an address will occur which is illegal on the 1103.

The numbers 3, 4, 5, 6, and 7 are reserved for future possible changes and should not be used for J. After any read command, a "check sum" of the input is in Cell 00005. When quite a few cards are read, it is recommended that this sum be printed out so that readin differences can be spotted between repeated runs.

II. Six field card (See card ECLF 12C).

A. (For j=0,) the card number columns are ignored on reading. On punching, consecutive numbers are punched here. The number for the last card punched is kept in cell 73254. This number is reset to zero (i.e. first card numbered will be one) when Flip is reloaded from MT or PT. It can, of course, be set at will by the main program. Six consecutive storage cells are used by each card.

B. For j=1, the card number is read and punched. On the card a seven digit decimal integer followed by sign is used. In the memory a binary integer is stored. If the binary integer exceeds 9,999,999 when punching, 9,999,999 is punched. Seven consecutive storage cells are used by each card with the card number in the first of the seven.

III. Eight field card. (For j=2).

This card form has eight similar fields of 10 columns each. These consist of (in consecutive order):

7 columns of decimal digits for the fractional part of the representation.

1 column for the sign of this part.

1 column for the digits of the exponent (characterestic) of the representation. Exponents greater than nine are indicated by double or triple punching. For a punch in the 12 row add 10 and for a punch in the 11 row add 20. The maximum decimal exponent possible is 37. Higher exponents will be read incorrectly.

1 column for the sign of this part.

IV. Use of the reproducer.

1. Prior to reading or punching any cards, execute the command 17 00000 73376, which picks one card in each channel. Do this <u>only once</u>, since Flip read and punch commands pick the next card each time. The extra station on the punch side is taken care of by the flip punch command and causes one extra cycle of the reproducer.

2. If improper functioning of the reproducer occurs, restore the reproducer with at least two cards picked, if punching, and the first to be read picked, if reading. Then start at 73771. This restores ES and stops. Press start to repeat that Flip read or punch command.

3. The command 37 73374 73374 will advance all cards processed into the receiving hoppers without disturbing further Flip read or punch commands. It takes an average of 4½ card cycles.

4. The command 17 00000 00077 will place a blank card in the output. This command consumes only a few micro seconds of time.

FLIP

Sine and Cosine Subroutine

Sine Command Code: 60

Cosine Command Code: 61

Number of Cells: 51

I Description

These two commands compute the sine or cosine of (x) and store it in y and R. A polynomial approximation is used(1) which gives a maximum error of the order of $-5 \cdot 10^{-9}$. For arguments so large that the roundoff error of the argument obscures the result, an alarm halt occurs. If the cosine subroutine is used, the sine subroutine must also be specified to the loader; and if the location of the first cell of the cosine subroutine is y, that of the sine subroutine must be y + 2.

(1) See sheet 14, "Approximations in Numerical Analysis", a publication of the Rand Corporation

FLIP

Arc Tangent and Arc Cotangent Subroutine

Arc Tangent Command Code: 62

Arc Cotangent Command Code: 63

Number of Cells: 47

I Description

These two commands compute $\tan^{-1}X$ or $\cot^{-1}X$ and store the result in y and R. A polynomial approximation is used.(1) The error is of the order of $1/2 \cdot 10^{-7}$. If the arc cotangent subroutine is used, the arctangent subroutine must also be specified to the loader; and if the location of the first cell of the arctangent subroutine is y, that of the arc cotangent subroutine must be y + 2.

All results lie between $-\pi/4$ and $+3\pi/4$.

(1) See sheet 13, "Approximations in Numerical Analysis", a publication of the Rand Corporation

FLIP BESSEL FUNCTIONS SUB ROUTINE

COMMAND CODES

64 $J_0(x) \rightarrow y$

65 $J_0(x) \rightarrow y$; $Y_0(x) \rightarrow 00037$

66 $J_1(x) \rightarrow y$

67 $J_1(x) \rightarrow y$; $Y_1(x) \rightarrow 00037$

LOCATION: 72000 to 72443

TEMPORARIES: 75110 to 75777

Do not specify to the Flip loader.

These functions are computed by numerical approximations (1) and are believed correct to seven decimal digits. Commands 64 and 66 are much faster than 65 and 67.

(1) M T A C, October 1954, pp 240-241

FLIP

Exponential Subroutine

Command Code: 70

Number of Cells: 27

I Description

This command computes the exponential of (x) and stores it in y and R. Full accuracy is obtained by a power series. For values of $(x) \geqslant 64$, an alarm halt occurs. e^{-x} is obtained by computing e^{x} and reciprocating.

FLIP COMPLEX ARITHMETIC SUBROUTINE

Command Codes

72 Y+X → S

73 Y−X → S

74 Y×X → S

75 Y÷X → S

Number of Cells: 42 (decimal)

Temporaries Used: 00002 to 00037

Use command code 72 to specify this to the loader.

These commands treat (x) + i (x +1) as X, (y) + i (y+1) as Y, and (00036) + i (00037) as S, where $i=\sqrt{-1}$. x or y or both can be 0036 and can be modified by counters b_1 and b_2. Both components of X, of Y, and of S are packed floating point numbers.

FLIP

Trace Subroutine

Command Code: 77

Number of Cells: 26

I <u>Description</u>

The trace routine operates in two phases. Phase I operates concurrently with the running of the routine being tested and stores information on Magnetic Tape 2 (MT2). Phase II operates separately from the routine being tested. It reads the information which was stored by Phase I on MT2, processes it and punches a paper tape output. The content of ES will be automatically restored after this phase.

II <u>Phase I</u>

The trace subroutine must be specified to the loader. Its "command code" for this purpose is 77. It requires 26 cells. When loaded, it will operate whenever MJ1 is on. The MJ instruction is in cell 01735. This subroutine uses cells 74000 to 74041 as temporaries.

III <u>Phase 2</u>

This operation uses the ES and cell 40000 but will restore both when completed. Its operating instructions are: -

1. Set PAK to 77600. Press Start. The 1103 prints out "Rewind MT2" and halts.
2. After rewinding MT2, start. (if PAK was disturbed, set it to 40000). The routine will search the tape for the data, then process it one block at a time. The output is punched on paper tape. The end of data will be apparent when the routine searches MT2 without punching paper tape. Halt.
3. To continue the problem, set PAK to 40000 and start. The 1103 will restore ES and 40 00 and halt with a 56 00000 40000 command.

FLIP Direct Charactron Trace

(71140 to 71400 and 77570 to 77630)

1. Activating the trace:

Start at 77600. Cells 00002 to 00014 and 74340 to 75777 will be disturbed by activating.

2. Tracing:

Start your routine at any desired point. Tracing will be printed on the Charactron.

3. Fault in any Flip command:

Start at 01735. This insures that this command appears on the trace.

4. Deactivating:

Start at 77600. This is necessary to finish the trace. The frames will be properly run out, but the film must be manually indexed within a few minutes to prevent sticking. The same cells are disturbed as in activating.

5. Continuing run:

Start at any desired point. Flip will run normally.

Tracing speed:- About 8 commands a second.

Restriction:- After tracing, Flip on MD must be restored before it can be loaded into ES again.

IV Output

The trace routine output prints a 32 digit line for each FLIP instruction. If a jump occurs, either an erroneous line appears or no line at all. Some FLIP subroutine commands include FLIP basic commands. These will appear as extra lines before the FLIP subroutine command line. The lines have the form:

AAAA OP xxxx yyyy ± q.qqqqqq ± pp

where AAAA is the last four digits of the address of the instruction.

OP is the command code.

xxxx is the basic x address.

yyyy is the basic y address.

The result of the operation, in floating decimal form, is $q \cdot 10^p$ where $1 \leqslant q < 10$. Some of the FLIP subroutine commands do not leave their result in R. For these, the result, $q \cdot 10^p$ will be erroneous.

In order to avoid confusion when several problems are traced using the same magnetic tape, Phase II overwrites the trace information as it is processed.

CONSOLIDATED VULTEE

DIVISION San Diego

MODEL All

REPORT ZM 490

DATE 1-18-56

TITLE

F L I P

A FLOATING POINT SUBROUTINE SYSTEM FOR ERA 1103 AND 1103A COMPUTER

PART II CODES AND FLOW CHARTS

PREPARED BY Charles J. Swift

GROUP Digital Computing Lab

REFERENCE

CHECKED BY Members of: Digital Computing Lab

APPROVED BY

NO. OF PAGES 100

NO. OF DIAGRAMS 7

REVISIONS

NO.	DATE	BY	CHANGE	PAGES AFFECTED

GD 241A

PX 71900-8-11

CV-4 - September, 1954 Rev. 6/28/54

Assembly Program - SC 001

Each subroutine to be assembled by the assembly program must be coded in absolute form as if its initial instruction were to be located at address 01000. In actual fact, however, the subroutine is placed in electrostatic storage at an almost arbitrary location selected by the coder. It is the function of the assembly routine to modify all addresses in the subroutine commands that are dependent on the location of the initial command. In order to perform this function correctly, the assembly routine must be informed of the actual electrostatic address s of the initial command of each subroutine and of the number n of successive commands to be modified in each case. It will then scan (s + j-1) j = 1,2,...n for each subroutine altering the ∅ and θ addresses by adding s-01000 to them whenever necessary. A particular address will be modified if and only if the 10th bit from the right in its 15 bit array is a one i.e. if it is of the form xxx xx1 xxx xxx xxx. For electrostatic addresses this means that all addresses of the form 01XXX (X an arbitrary octal digit) and only such addresses will be modified. Hence, all electrostatic addresses which are to be absolute i.e. independent of the location of the initial instruction must have the form 00XXX. The addresses of the accumulator and Q-register should be given as 20000 and 10000 respectively in order to avoid assembly modification. Since the assembly routine modifies n successive commands of the subroutine starting from some initial one, subroutine constants should commence in address 10000 + n.

The assembly subroutine is coded as if its initial command were stored at address 00100 and its final command at address 00121 and must be so located in electrostatic memory to be properly used. It is permanently stored at drum locations 77756-77777 (18).

Before entering the assembly subroutine the coder must set the exit address ∅ (00101). This can be done expeditiously by entering the subroutine with the instruction 37 00101 00100. This stores the return address into ∅ (00101) before entering, so that control will be returned to the next address following the return jump.

The coder must provide the assembly routine with the various numbers s & n which it needs. If there are k subroutines to be assembled, this information must be [illegible] at addresses 00121 + i, i = 1,2,...,k, where op (00121 + i) = 00, θ (00121 + i) = n. ∅ (00121 + i) = s. The (00121 + k + 1) must always be set identically equal to zero. Thus for example, if k = 3 (i.e. three subroutines are assembly modified) and n_1 = 13, n_2 = 5, n_3 = 21; s_1 = 01054, s_2 = 00200, and s_3 = 01700 (all numbers being octal) the data for the assembly routine would be as follows:

Address	Contents
00122	00 00013 01054
00123	00 00005 00200
00124	00 00021 01700
00125	00 00000 00000

The coder should bear in mind that the assembly routine itself uses as temporary storage the locations 00005, 00006, and 00007.

Summary

Only suitably coded subroutines may be assembled. A subroutine is considered to be suitably coded if it satisfies the following conditions.

(1) it is written in absolute form as if its initial instruction were to be located at address 01000.

(2) all addresses independent of the location of the initial instruction are of the form xxx xx0 xxx xxx xxx.

(3) all addresses to be modified are of the form xxx xx1 xxx xxx xxx.

(4) all subroutine constants are located immediately following the last subroutine instruction.

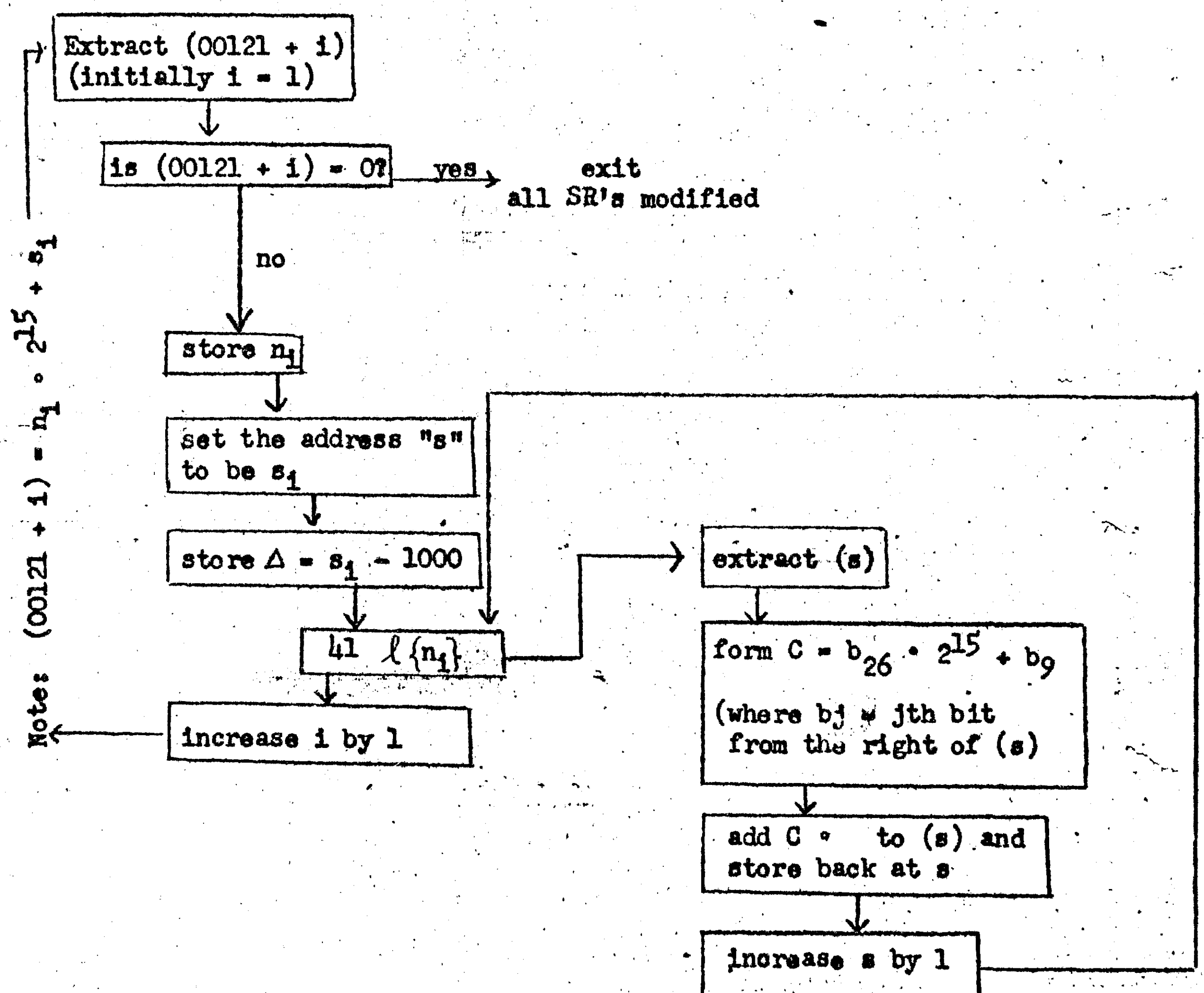
The coder must load all subroutines to be modified into the locations in electrostatic storage which he has chosen.

The coder must block transfer the assembly program from drum addresses 77756-77777 to electrostatic addresses 00100-00121.

The coder must load the numbers n and s for each subroutine to be modified into addresses 00121 + i, i = 1,2,... k, where k is the number of subroutine to be modified.

The coder must load zero into the address 00121 + k + 1.

The coder must set ∅ (00101) before entering the assembly program, and must provide for entry to the assembly program.



PX 71900-8-11

TABLE OF CONTENTS

(Con't)

PX 71900-8-11

TABLE OF CONTENTS (Con't)

FLIP (revised 12/9/54)

Subroutine Specifications

FLIP subroutines are coded to start in cell 01000 and are modified by the assembly routine under control of the FLIP loader routine. Each subroutine is assigned a command code number.

I Command Code Parameter

If the subroutine is assigned command number OP (octal), a parameter is placed in drum location 76555 + 2 · OP. It has the form:-

m_1	D.A.	n	m_2
xx	xxxxx	xxx	0 x

where D.A. = Drum address of the subroutine

$64m_2 + m_1$ = number of cells occuppied

n = number of cells modified

II Input Information

At the entry to the subroutine, the temporaries contain this information:

Second Octal Digit of Command Code

	0	1	2	3	4	5	6	7
00005	(x)	(x)	(x)	(x)	-(x)	-(x)	-(x)	-(x)
00006	(y)	(R)	(y)	(y)	(y)	(R)	(y)	(y)
00007	0	0	0	(R)	0	0	0	(R)
∅ (01734)	y	y	20000	20000	y	y	20000	20000

00004 Command code in 1st two octal digits.

00010 36 bit extension of exponent from (00005)

00011 36 bit extension of exponent from (00006)

00013 Execution addresses x and y in the u and v positions.

R 36 bit extension of exponent from (00005)

10000 Command code in last two octal digits.

Notes: If any argument was zero, it has been transformed to 000000000200 (octal) with the exponent 777777777600 (octal).

FLIP (revised 12/9/54)

III Exit Information

Subroutine exits are to 01607, 01721, 01734, 01735 or 76045.

a. Exit at 01607. Place zero in A and go to 01734 (see below)

b. Exit at 01721. Normalize, test, pack and store. In this case the subroutine should leave:

q (35) in 00005

p (0) in 00010

Where the desired result is $q \cdot 2^p$. q and p may be any possible 36 bit numbers. The routine will normalize, test, pack into R, and go to 01734.

c. Exit at 01734, 01735 or 01737

01734	11 20000	[00000]	STORE RESULT	(see table above)
01735	45 10000	[01736]	TRACE?	(exits to trace if used)
01736	45 00000	[00000]	Exit	
01737	37 00000	76045	ALARM EXIT	

Notes:

1. All references to exits above must be in unmodified orders.
2. Double duty subroutines (e.g. sine and cosine) are assigned two command codes and treated as two overlapping subroutines.

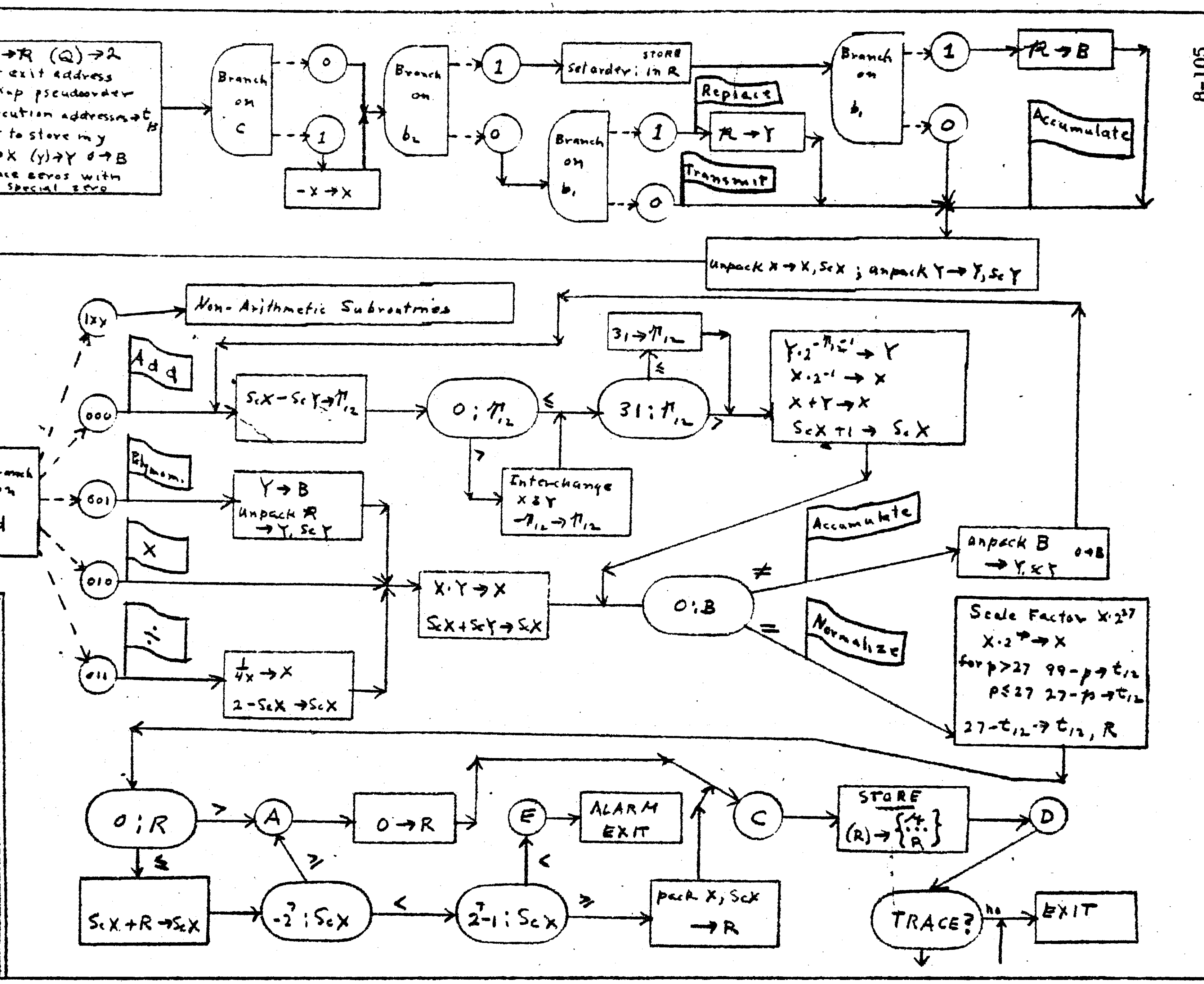
IV "Own Code" Subroutines

Frequently occurring subroutines can be coded as FLIP subroutines. Before using other FLIP commands in such subroutines, it is necessary to transfer ∅ (1736) and such other information as must be saved to new locations. FLIP basic orders do not use cells 00015 to 00037 or cell 00002. If the subroutine needs no modification or change of location, use 00 00001 00000 for its command code parameter.

ANALYSIS
PREPARED BY C. J. Swift
CHECKED BY
REVISED BY

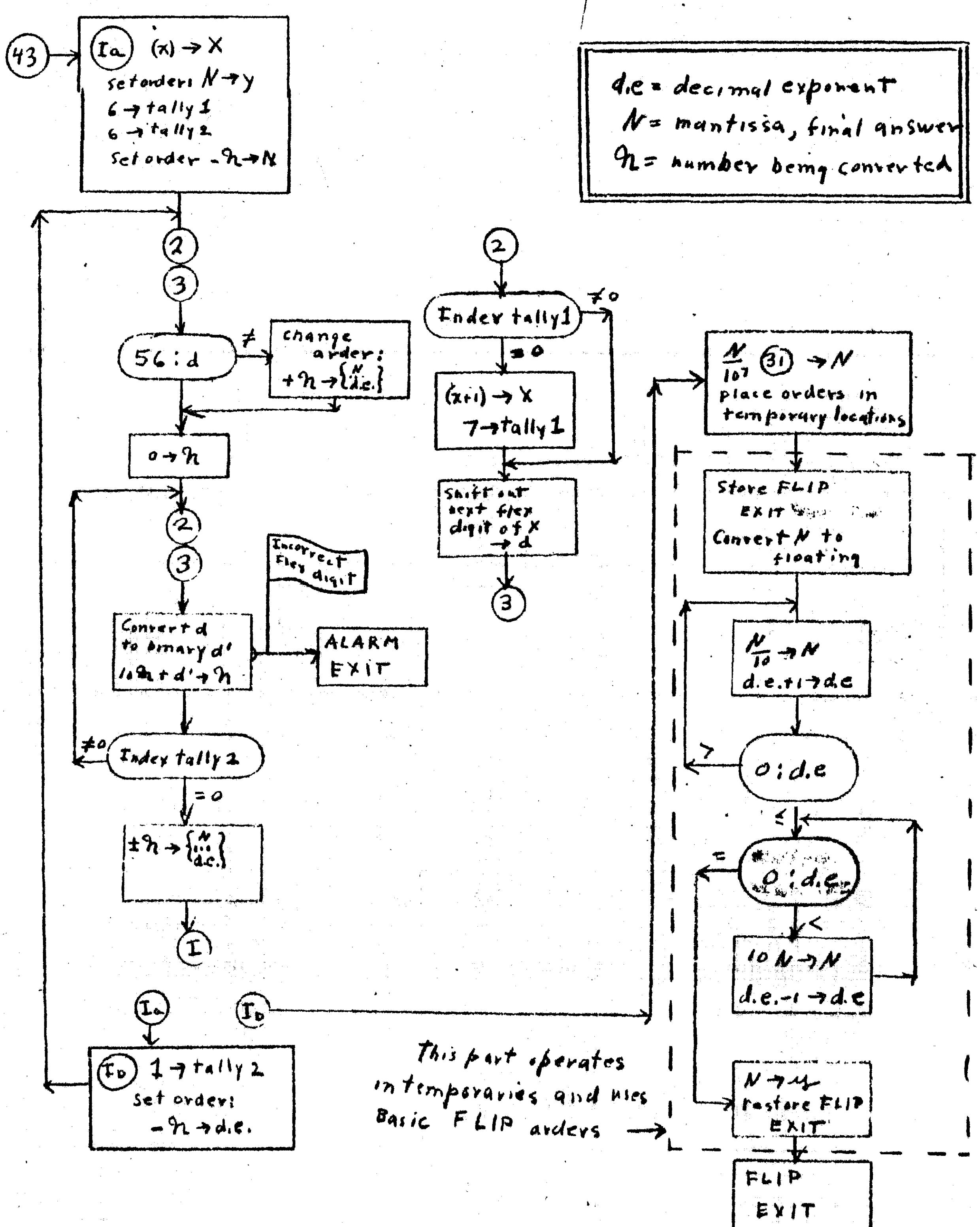
CONSOLIDATED VULTEE AIRCRAFT CORPORATION
SAN DIEGO DIVISION
CV-11

FLOW CHART ARITHMETIC OPERATIONS



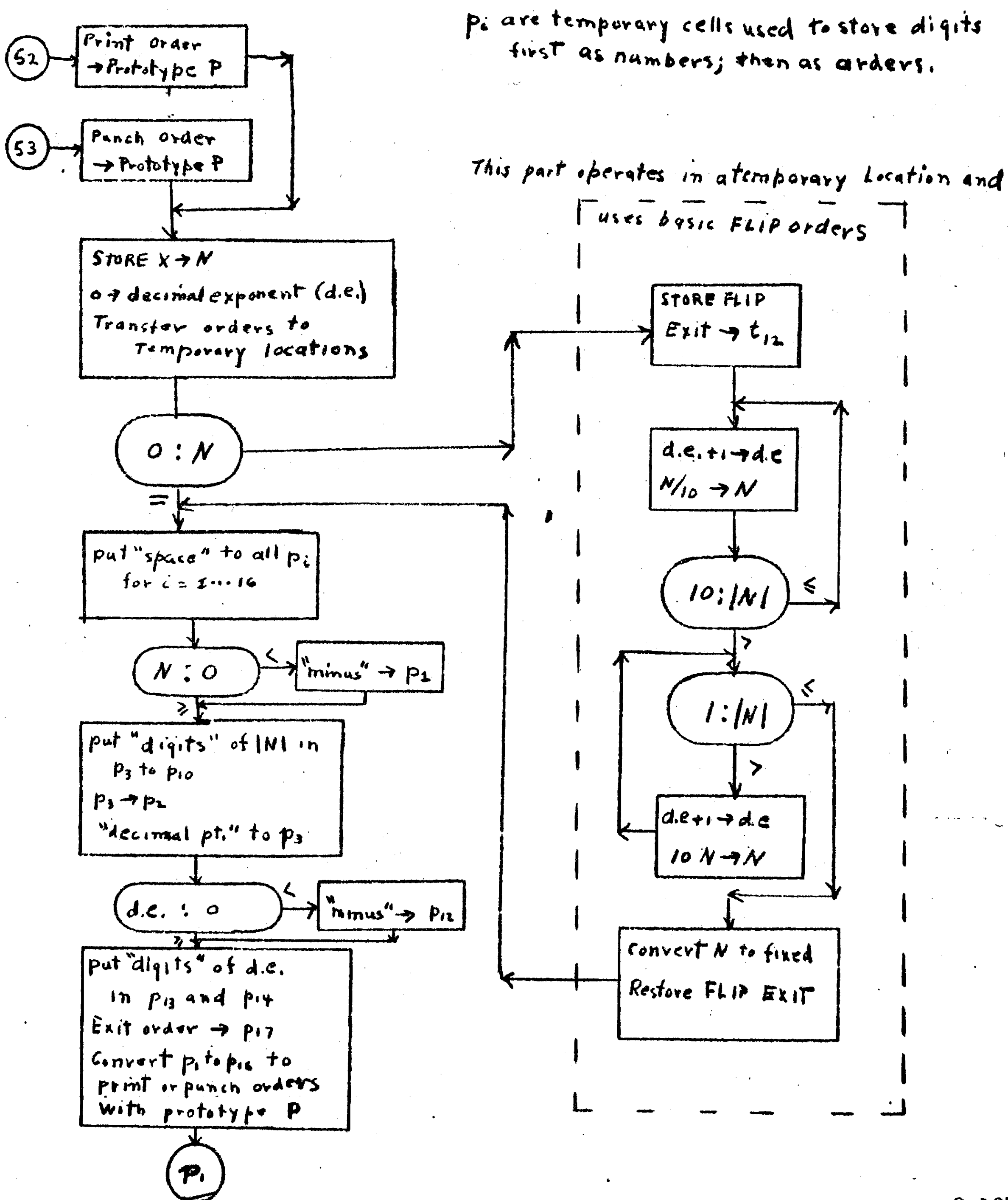
FLEXOWRITER INPUT

II Flow Chart



Print and Punch Subroutine

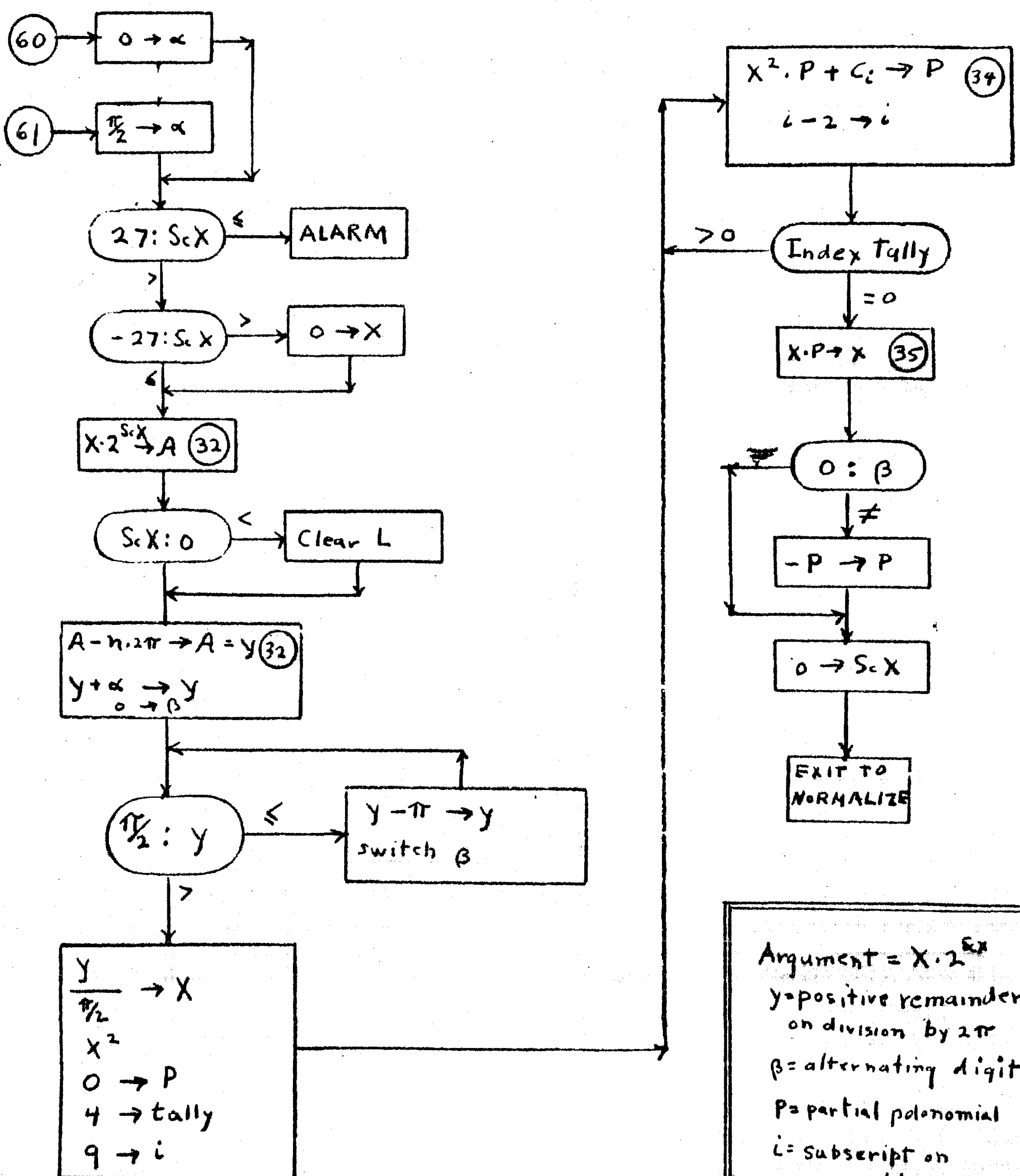
II Flow Chart



PX 71900-8-11

Sine and Cosine Subroutine

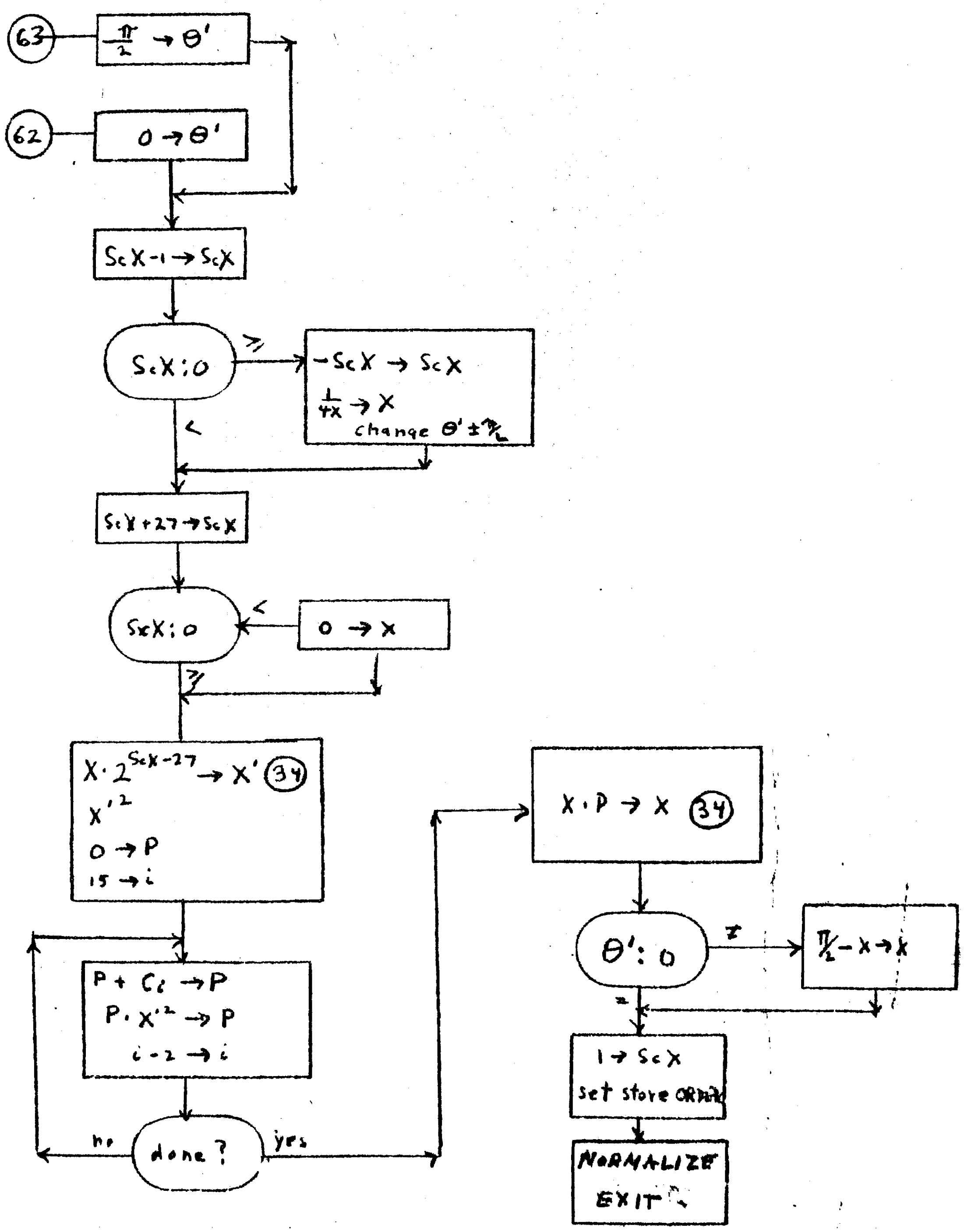
II Flow Chart



Argument = $X \cdot 2^{Sx}$
y = positive remainder on division by 2π
β = alternating digit
P = partial polonomial
i = subscript on coefficients

Arc Tangent and Arc Cotangent Subroutine

II Flow Chart



PX 71900-8-11

FLIP CARD ROUTINES

FLOATING BINARY TO FLOATING DECIMAL CONVERSION

(Approx 8 milliseconds on 1103)

Initially

$$n = N \cdot 2^{s} \qquad 1 > |N| \geq 1/2 \qquad 2^{-7} < s < 2^{7}$$

Define $R = 10^3/2^{10}$

Step 1

Convert to Minus Exponent

$$S' \equiv S - 120 \qquad -247 \leq S' \leq 7$$

$$N' \equiv N \cdot R^{-12} \cdot 2^{-3}$$

$$n = N' \cdot 10^{36} \cdot 2^{S'} \cdot 2^{3} \qquad 2^{-2} > |N'| > 2^{-4}$$

Step 2

Change Base to 2^{10}

$$S' \equiv 10S'' + t \qquad 0 \leq t \leq 9 \qquad -25 \leq S'' \leq 0$$

$$n = N' \cdot (2^{10})^{S''} \cdot 10^{36} \cdot 2^{3+t}$$

Step 3

Change Base to 10^3

$$N'' \equiv N' \cdot (R)^{-S''} \quad [\text{use table of } R^{2^i} \text{ for } i = 0 \cdots 4]$$

$$n = N'' \cdot 10^{36+3S''} \cdot 2^{3+t} \qquad 2^{-4} < |N''| < 1$$

Step 4

Change Base to 10

$$p \equiv 36 + 3S''$$

$$N''' \equiv 2^{3+t} N''$$

Normalize decimally to get

$$N'''' \equiv 10^{-\alpha} \cdot N'''$$

$$n = N'''' \cdot 10^{p+\alpha}$$

FLIP CARD ROUTINES

FLOATING DECIMAL TO FLOATING BINARY CONVERSION

(Approx 10 ms on 1103)

Initially

$$n = N \cdot 10^{p} \qquad 1 > |N| > 1/10 \qquad |p| < 39$$

define $R \equiv 10^3/2^{10}$

Step 1

Convert to Positive Decimal Exponent

$$p' \equiv p + 39 \qquad 0 < p' < 78$$

$$N' \equiv N \cdot (R)^{-13} \cdot 2^{-2} \qquad 1/20 < N' < 1/2$$

$$n = N' \cdot 2^{-132} \cdot 10^{p'}$$

Step 2

Change Base to 10^3

$$p' \equiv 3p'' + t \qquad 0 \le t < 3 \quad 0 \le p'' < 26$$

[this is done by dividing p' by 3]

$$n = N' \cdot 2^{-132} \cdot (10^3)^{p''} \cdot 10^{t}$$

Step 3

Change Base to 2^{10}

$$p'' \equiv \alpha_4 \cdot 2^4 + \alpha_3 \cdot 2^3 + \alpha_2 \cdot 2^2 + \alpha_1 \cdot 2^1 + \alpha_0 \cdot 2^0$$

(α_i is zero or one)

$$N'' \equiv N' \prod_{i=0}^{4} (R)^{\alpha_i \cdot 2^i} = N'(R)^{p''} \qquad 1/2 > N'' > 1/80$$

$$n = N'' \, 2^8 \, N'' \cdot 10^{t}$$

Step 4

Change to Base 2

$$N''' \equiv 2^8 N'' \cdot 10^{t} \qquad 2^8/80 < N''' < 2^7 \cdot 10^2$$

$$n = N''' \cdot 2^{10p'' - 140}$$

Normalize N''' to get

$$N'''' = N''' \cdot 2^{-\alpha}$$

$$n = N'''' \cdot 2^{10p'' - 136 + \alpha}$$

PX 71900-8-11

MAGNETIC TAPE ROUTINE FOR SERVICE ROUTINES

70000	70000	11	00000	74000
70001	70001	11	70043	00000
70002	70002	75	30037	70004
70003	70003	11	00001	74001
70004	70004	16	70045	76000
70005	70005	15	70045	76005
70006	70006	16	70045	76030
70007	70007	11	70044	76037
70010	70010	16	70045	76041
70011	70011	15	70045	77027
70012	70012	16	70045	77030
70013	70013	11	77037	77031
70014	70014	16	70045	77234
70015	70015	11	70044	77241
70016	70016	11	70044	77242
70017	70017	16	70045	77255
70020	70020	11	70044	77275
70021	70021	11	70044	77300
70022	70022	23	10000	10000
70023	70023	75	24000	70025
70024	70024	32	70000	00000
70025	70025	75	22000	70027
70026	70026	32	76000	00000
70027	70027	11	20000	20000
70030	70030	47	70037	70031
70031	70031	75	10007	70031
70032	70032	61	00000	70047
70033	70033	75	30037	70035
70034	70034	11	74001	00001
70035	70035	11	74000	00000
70036	70036	57	00000	00000
70037	70037	61	00000	70047

MAGNETIC TAPE ROUTINE FOR SERV

```
70040   70040   61 00000 70046
70041   70041   56 10000 70042
70042   70042   64 30001 00000
70043   70043   45 00000 00001
70044   70044   00 00000 00 *
70045   70045   00 30000 30000
70046   70046   00 00000 00026
70047   70047   00 00000 00045
70050   70050   00 00000 00 24
70051   70051   00 00000 00004
70052   70052   00 00000 00012
70053   70053   00 00000 00004
70054   70054   00 00000 00003
70055   70055   00 00000 00036
70056   70056   11 06724 56451
```

MAGNETIC TAPE ROUTINE FOR FLIP

```
70100   70100   11 70112 00000     SET F1
70101   70101   75 30036 70103     SAVE ES
70102   70102   11 00002 71742      WORKING SPACE
70103   70103   64 00001 00000     READ ONE BLOCK OF MT
70104   70104   11 00000 20000     LOCATER
70105   70105   43 70113 70110       BLOCK #
70106   70106   35 70107 70107     BACK UP MT
70107   70107   22 00001 00000      TO CORRECT PLACE
70110   70110   66 00000 00000     ADVANCE TO FLIP III
70111   70111   64 00001 00000     READ IN FIRST BLOCK MT
70112   70112   45 00000 00002     JUMP
70113   70113   45 07777 00002
```

ACTIVATOR

70140	00074	75	30070	70142	SAVE CELLS
70141	00075	11	00100	74100	TO BE USED
70142	00076	31	70160	00017	SET COMMAND TO
70143	00077	15	20000	70145	PICKUP PARAMETERS
70144	00100	75	30040	70146	PICKUP
70145	00101	11	30000	74200	PARAMETERS
70146	00102	75	30016	00120	LOAD ROUTINE
70147	00103	11	70150	00104	TO ES
70150	00104	21	00105	00116	LOCATE
70151	00105	11	00121	20000	ZERO
70152	00106	47	00104	00107	FLAG
70153	00107	31	00105	00071	SET
70154	00110	23	20000	00117	EXIT
70155	00111	35	00114	70160	COMMAND
70156	00112	16	00111	76645	ENTER OLD
70157	00113	45	00000	76575	LOADER ROUTINE
70160	00114	75	30070	30000	RESTORE ES
70161	00115	11	74100	00100	AND EXIT
70162	00116	00	00001	00000	CONSTANTS
70163	00117	00	00011	00121	
70164	00120	75	30040	00104	LOAD PARAMETERS
70165	00121	11	74200	00122	TO ES

PX 71900-8-11

COMPLEX ARITH FLIP

```
70620  01000  15 00013 01006     SET X PICKUP
70621  01001  31 00013 00017     SET Y
70622  01002  15 20000 01024       PICKUP
70623  01003  75 30020 00032     LOAD ROUTINE
70624  01004  11 01005 00015       TO TEMPORARIES
70625  00015  75 30002 00017     PICKUP
70626  00016  11 30000 00032        X
70627  00017  55 00004 10004     TEST
70630  00020  44 00021 00023       COMMAND CODE
70631  00021  75 30007 00021     LOAD ADD
70632  00022  11 01025 00021       AND SUBTRACT
70633  00023  75 30011 00025     LOAD COMMON PART
70634  00024  11 01034 00013       MULTIPLY AND DIVIDE
70635  00025  44 00026 00015     TEST COMMAND CODE
70636  00026  75 30005 00013     LOAD
70637  00027  11 01045 00023       DIVIDE
70640  00030  16 00002 01736     RESTORE FLIP EXIT
70641  00031  45 06003 01735     EXIT
70642  00032  16 01736 00002     SAVE FLIP EXIT
70643  00033  75 30002 00015     PICKUP
70644  00034  11 30000 00034       Y
70645  00021  44 00022 00024     ADD OR SUBTRACT
70646  00022  15 00031 00024     CHANGE COMMANDS TO
70647  00023  15 00031 00026        SUBTRACT X
70650  00024  1402 0032 0034     ADD
70651  00025  11 20000 00036       REAL PARTS
70652  00026  1402 0033 0035     ADD IMAGINARY
70653  00027  11 20000 00037       PARTS
70654  00013  15 00016 00021     CHANGE COMMANDS TO
70655  00014  15 00024 00016       REVERSE SIGN OF IMAG X
70656  00015  1422 0032 0034     REAL
70657  00016  1427 0033 0035       PART OF
```

CV-11

COMPLEX ARITH FLIP

70660	00017	11 20000 00036	PRODUCT
70661	00020	1422 0032 0035	IMAGINARY
70662	00021	1423 0033 0034	PART OF
70663	00022	11 20000 00037	PRODUCT
70664	00023	45 00000 00030	JUMP TO EXIT
70665	00023	1422 0032 0032	COMPUTE
70666	00024	1423 0033 0033	DENOMINATOR
70667	00025	11 20000 00032	FOR QUOTIENT
70670	00026	1430 0032 0036	FINISH
70671	00027	1430 0032 0037	DIVISION

INSPECT AND CHANGE

70700	70700	11	70727	00000	SET FI
70701	70701	23	10000	10000	CLEAR A AND Q
70702	70702	56	00000	70703	HALT
70703	70703	11	40000	70744	SAVE 40000
70704	70704	11	20000	70745	STORE NEW
70705	70705	16	10000	70726	SET
70706	70706	55	10000	00017	VARIABLE
70707	70707	15	10000	70716	COMMANDS
70710	70710	61	00000	70773	PRINT
70711	70711	75	10007	70713	OUT
70712	70712	61	00000	70752	CHANGE
70713	70713	55	10000	00006	PRINT
70714	70714	11	70747	70746	OUT
70715	70715	37	70743	70735	ADDRESS
70716	70716	11	30000	10000	PRINT OUT
70717	70717	37	70743	70734	OLD
70720	70720	11	70745	20000	REPLACE NEW
70721	70721	11	70730	40000	SET 40000
70722	70722	56	00000	70723	HALT
70723	70723	11	20000	10000	PRINT
70724	70724	37	70743	70734	NEW
70725	70725	11	70744	40000	RESTORE 40000
70726	70726	11	10000	30000	STORE NEW
70727	70727	45	00000	70701	JUMP
70730	70730	45	00000	70731	JUMP
70731	70731	11	70744	40000	RESTORE 40000
70732	70732	75	10012	70700	PRINT
70733	70733	61	00000	70747	NO CHANGE
70734	70734	11	70757	70746	OCTAL
70735	70735	61	00000	70747	WORD
70736	70736	55	10000	00003	PRINT
70737	70737	51	70761	20000	SUB

INSPECT AND CHANGE

70740	70740	35	70762	70741	ROUTINE
70741	70741	00	00000	00 *	
70742	70742	41	70746	70736	
70743	70743	45	00000	30000	
70744	70744	00	00000	00 0	
70745	70745	00	00000	00 0	
70746	70746	00	00000	00*0	
70747	70747	00	00000	00004	
70750	70750	00	00000	00 *6	
70751	70751	00	00000	00003	
70752	70752	00	00000	00004	
70753	70753	00	00000	00016	
70754	70754	00	00000	00005	
70755	70755	00	00000	00030	
70756	70756	00	00000	00006	
70757	70757	00	00000	00013	
70760	70760	00	00000	00 20	
70761	70761	00	00000	00007	
70762	70762	61	00000	70763	
70763	70763	00	00000	00037	
70764	70764	00	00000	00052	
70765	70765	00	00000	00074	
70766	70766	00	00000	00070	
70767	70767**	00*	00000	00064	
70770	70770	00	00000	00 62	
70771	70771	00	00000	00066	
70772	70772	00	00000	00072	
70773	70773	00	00000	00045	

PX 71900-8-11

BIOCTAL TAPE DUMP

```
71000   71000   37 71052 71014      MANUAL PRESET
71001   71001   23 10000 10000      CLEAR A AND Q
71002   71002   56 00000 71000      HALT
71003   71003   31 71007 00017      PICKUP
71004   71004   15 20000 71005        NEXT
71005   71005   31 01004 00005          PARAMETER
71006   71006   43 20000 71010      DONE ?
71007   71007   45 00000 01005      EXIT
71010   71010   55 20000 00037      PARAMETER IN A AND Q
71011   71011   21 71007 00074      STEP EXIT
71012   71012   37 71052 71014      AUTOMATIC PRESET
71013   71013   45 00000 71003      RETURN
71014   71014   75 30030 71016      LOAD ROUTINE
71015   71015   11 71053 00010        IN TEMPORARIES
71016   71016   31 10000 00017      SET FINAL
71017   71017   15 20000 00034        ADDRESS
71020   71020   15 10000 00010      SET PICKUP
71021   71021   31 10000 00071      TEST STORAGE
71022   71022   46 71023 71024        CLASS
71023   71023   54 00036 00004      SET EXTRACTOR FOR MD
71024   71024   37 71046 71025      SET SWITCH
71025   71025   75 00260 71103      PUNCH
71026   71026   63 00000 00040        LEADER
71027   71027   31 00073 00010      SET UP
71030   71030   35 00010 00035        INTERIM
71031   71031   11 00036 10000          CHECK
71032   71032   53 00034 00035            ADDRESS
71033   71033   15 00010 01775      INSERT ADDRESS
71034   71034   55 01775 10025        TO Q
71035   71035   15 00023 00020      SET TO PUNCH
71036   71036   15 00022 00021          INSERT ADDRESS
71037   71037   37 00024 00011      PUNCH INSERT ADDRESS
```

PX 71900-8-11

BIOCTAL TAPE DUMP

71040	71040	15	00022	00020	SET TO PUNCH
71041	71041	45	00000	00010	DATA WORDS
71042	71042	15	00010	01775	CHECK ADDRESS
71043	71043	55	01775	10025	TO Q
71044	71044	15	00023	00021	PUNCH CHECK
71045	71045	37	00024	00011	ADDRESS
71046	71046	75	00100	71025	PUNCH
71047	71047	63	00000	00040	TRAILER
71050	71050	37	00030	00010	PUNCH LAST FRAME
71051	71051	37	71046	71042	PUNCH CHECK ADDRESS
71052	71052	45	00000	71001	RETURN
71053	00010	11	34000	10000	PICKUP
71054	00011	16	00016	00013	PRESET
71055	00012	55	10000	00006	ASSEMBLE
71056	00013	11	10000	34000	
71057	00014	21	00013	00074	BIOCTAL
71060	00015	42	00037	00012	DIGITS
71061	00016	63	00000	00002	PUNCH
71062	00017	63	00000	00003	
71063	00020	63	00000	00004	
71064	00021	63	10000	00005	
71065	00022	63	00000	00006	
71066	00023	63	10000	00007	DIGITS
71067	00024	37	00024	34000	SWITCH
71070	00025	21	00010	00073	STEP
71071	00026	11	00036	10000	PICKUP
71072	00027	53	00034	00010	COMMAND
71073	00030	37	00030	00031	SWITCH
71074	00031	43	00034	71050	FINAL END
71075	00032	43	00035	71042	INTERIM END
71076	00033	45	00000	00010	
71077	00034	11	34000	10000	FINAL ADDRESS

BIOCTAL TAPE DUMP

71100	00035	11	34000	10000	INTERIM CHECK ADDRESS
71101	00036	00	36000	00000	XTRACTOR TO PREVENT CARRY
71102	00037	11	10000	00010	COMPARATOR
71103	71103	63	10000	00040	PUNCH INITIAL FRAME
71104	71104	23	01775	01775	CLEAR TEMPORARY
71105	71105	45	00000	71027	RETURN

By: C. J. Swift

FLIP CHARACTRON TRACE

PROCESSING SUBROUTINE

71140	434	75	31340	71142	SAVE
71141	435	11	00440	74440	ES IN MD
71142	436	75	30240	00675	LOAD INTO
71143	437	11	71144	00440	ES
71144	440	16	01600	75506	RESTORE TALLY COMMAND
71145	441	75	30140	00443	LOAD TRACE
71146	442	11	77637	01330	INFORMATION TO ES
71147	443	16	00473	01735	REMOVE TRACE JUMP
71150	444	17	00000	00610	START CHARACTRON
71151	445	75	10011	00450	PRINT
71152	446	77	10000	00637	HEADING
71153	447	00	00000	0000	
71154	450	75	30003	00452	PICKUP
71155	451	11	01330	00640	INFORMATION
71156	452	54	00641	20052	TEST
71157	453	11	20000	20000	FOR
71160	454	43	00613	00474	14 COMMAND
71161	455	21	00451	00616	STEP COMMAND
71162	456	42	00473	00561	DONE BIN ?
71163	457	21	71176	00614	STEP PAGE NO
71164	460	11	20000	10000	PLACE IN Q
71165	461	11	00615	00627	SET POSITION
71166	462	37	00604	00572	PRINT DECIMAL DIGIT
71167	463	37	00604	00572	PRINT DECIMAL DIGIT
71170	464	17	00000	00611	STOP CHARACTRON
71171	465	17	00000	00612	TURN PAGE

PX 71900-8-11

By: C. J. Swift

FLIP CHARACTRON TRACE

71172	466	23	00523	00470	TEST WHETHER
71173	467	47	00470	00521	ACTIVATING
71174	470	75	31340	01566	RESTORE
71175	471	11	74440	00440	ES
71176	472	76	56043	77777	PAGE NUMBER
71177	473	11	01461	01736	
71200	474	21	00451	00616	STEP COMMAND
71201	475	15	00632	00627	PRESET VERTICAL POSITION
71202	476	55	00640	10006	PICKUP ADDRESS TO Q LEFT
71203	477	11	00617	00644	5 TO TALLY
71204	500	37	00604	00501	PRINT FIVE
71205	501	41	00644	00570	OCTAL DIGITS
71206	502	21	00627	00620	STEP HORIZONTALLY
71207	503	11	00044	00644	4 TO TALLY
71210	504	11	00641	10000	COMMAND TO Q
71211	505	37	00604	00506	PRINT FOUR
71212	506	41	00644	00570	OCTAL DIGITS
71213	507	21	00627	00607	STEP HORIZONTALLY
71214	510	11	00044	00644	4 TO TALLY
71215	511	37	00604	00512	PRINT FOUR
71216	512	41	00644	00570	OCTAL DIGITS
71217	513	21	00627	00607	STEP HORIZONTALLY
71220	514	11	00044	00644	4 TO TALLY
71221	515	37	00604	00516	PRINT FOUR
71222	516	41	00644	00570	OCTAL DIGITS
71223	517	21	00627	00621	STEP HORIZONTALLY
71224	520	45	00000	00524	JUMP
71225	521	75	10140	00470	CLEAR TRACE

FLIP CHARACTRON TRACE

71226	522	11	00040	77637	BIN
71227	523	75	31340	01566	COMPARATOR
71230	524	11	00040	00645	ZERO TO DECIMAL EXPONENT
71231	525	21	00645	00074	STEP DECIMAL EXPONENT
71232	526	14	30062	40642	DIVIDE X BY 10
71233	527	14	45062	20525	COMPARE WITH ONE
71234	530	14	45062	10540	COMPARE WITH ONE TENTH
71235	531	23	00645	00074	STEP DECIMAL EXPONENT
71236	532	14	20062	40642	MULTIPLY X BY TEN
71237	533	47	00530	00540	
71240	534	13	00007	10000	
71241	535	75	10176	00243	
71242	536	12	10000	01601	
71243	537	21	10000	01776	
71244	540	14	51062	20642	CONVERT
71245	541	54	00642	20001	TO FIXED
71246	542	46	00543	00545	SIGN
71247	543	77	10000	00626	TEST
71250	544	33	20000	00000	AND PRINT
71251	545	35	00631	10000	ROUND
71252	546	77	10000	00625	PRINT DECIMAL POINT
71253	547	11	00067	00644	SEVEN TO TALLY
71254	550	37	00604	00551	PRINT SEVEN
71255	551	41	00644	00572	DECIMAL DIGITS
71256	552	71	00645	00614	SHIFT EXPONENT
71257	553	46	00554	00555	SIGN
71260	554	77	10000	00630	TEST
71261	555	12	20000	10000	AND PRINT

By, C. J. Swift

CV-11

FLIP CHARACTRON TRACE

71262	556	21	00627	00620	STEP HORIZONTALLY
71263	557	37	00604	00572	PRINT TWO
71264	560	37	00604	00572	DECIMAL DIGITS
71265	561	75	20004	00450	STEP
71266	562	21	00625	00606	VERTICALLY
71267	563	00	00000	0000	
71270	564	00	00000	0000	
71271	565	00	00000	0000	
71272	566	00	00000	0000	
71273	567	00	00000	0000	
71274	570	31	10000	00003	OCTAL DIGIT ENTRANCE
71275	571	45	00000	00574	
71276	572	31	10000	00002	DECIMAL DIGIT ENTRANCE
71277	573	32	10000	00001	
71300	574	11	20000	10000	REMAINDER TO Q
71301	575	34	20000	00063	CLEAR ACC AND SHIFT DIGIT
71302	576	35	00605	00577	SET COMMAND
71303	577	11	00650	00037	PLACE DIGIT
71304	600	15	00627	00447	SET
71305	601	16	00627	00447	POSITION
71306	602	77	10000	00447	PRINT DIGIT
71307	603	21	00627	00607	STEP HORIZONTALLY
71310	604	45	00000	30000	EXIT
71311	605	11	00650	00447	PROTOTYPE
71312	606	00	00000	00036	VERTICAL STEP
71313	607	00	00024	00000	HORIZONTAL STEP
71314	610	00	00000	05000	CHARACTRON
71315	611	00	00000	04400	OPERATING

PX 71900-8-11

By: C. J. Swift

CV-11

FLIP CHARACTRON TRACE

71316	612	00	00000	06000	CODES
71317	613	00	00000	00014	12
71320	614	00	50756	00000	10^{-2} TIMES 2^{36}
71321	615	00	00753	01777	POSITION FOR PAGE NUMBER
71322	616	00	00003	00000	INCREMENT
71323	617	00	00000	00005	5
71324	620	00	00050	00000	INCREMENTS
71325	621	00	00074	00000	
71326	622	37	77777	30000	ONE ROUNDED DOWN
71327	623	31	46314	13374	ONE TENTH ROUNDED DOWN
71330	624	24	00000	00004	10
71331	625	30	01150	00106	DEC PT
71332	626	40	01120	00074	MINUS
71333	627	00	00230	00074	LOCATION
71334	630	40	01424	00074	MINUS
71335	631	00	00000	12000	ROUND
71336	632	00	00230	00000	HORIZONTAL PRESET
71337	633	00	00000	00000	
71340	634	00	00000	00000	
71341	635	00	00000	00000	T
71342	636	00	00000	00000	E
71343	637	23	00620	00000	M
71344	640	33	00644	00000	P
71345	641	01	00666	00000	O
71346	642	41	00712	00000	R
71347	643	45	00762	00000	A
71350	644	43	01006	00000	R
71351	645	14	01032	00000	I

PX 71900-8-11

By: C. J. Swift

FLIP CHARACTRON TRACE

71352	646	20	01056	00000	E
71353	647	22	01102	00000	S
71354	650	00	00000	00 0	D
71355	651	01	00000	00000	I
71356	652	02	00000	00000	G
71357	653	03	00000	00000	I
71360	654	04	00000	00000	T
71361	655	05	00000	00000	T .
71362	656	10	00000	00000	A
71363	657	11	00000	00000	B
71364	660	12	00000	00000	L
71365	661	13	00000	00000	E
71366	662	00	00000	00000	
71367	663	11	01701	20000	
71370	664	36	20000	72612	CLEAR PAGE NUMBER
71371	665	17	00000	71316	MOVE A PAGE
71372	666	23	00674	72500	TEST FOR OUTPUT
71373	667	47	00440	00670	SUBROUTINE
71374	670	11	72612	20000	TEST FOR
71375	671	43	00673	00664	NEW
71376	672	47	00665	00440	PAGE
71377	673	00	00000	00200	CONSTANT
71400	674	75	30274	01100	TEST COMMAND
71401	675	75	30021	00666	RESTORE
71402	676	11	77027	01551	FLI^ IN ES

PX 71900-8-11

MAGNETIC TAPE STORAGE

```
71500   71500   16 71543 71562      SET MANUAL EXIT
71501   71501   11 71560 00000      SET F1
71502   71502   75 30036 71504      LOAD
71503   71503   11 71506 00001       ES PART I
71504   71504   75 30030 00004      LOAD
71505   71505   11 71541 01551       ES PART II
71506   00001   64 00001 00000      ROUTINE
71507   00002   75 30034 00001       FOR
71510   00003   11 00004 30000        RELOADING
71511   00004   11 20000 00037      SAVE F AND L
71512   00005   21 00006 10000      ADVANCE
71513   00006   00 00001 00000        TAPE
71514   00007   11 01571 10000      EXTRACTOR TO Q
71515   00010   53 00006 00001      SET
71516   00011   53 20000 01555       TAPE
71517   00012   53 20000 01565        COMMANDS
71520   00013   15 00037 01553      SET
71521   00014   54 00037 10025        COMMANDS
71522   00015   54 00037 00071         FOR PICKUPS
71523   00016   16 20000 00003          AND STORE
71524   00017   31 10000 00063      COMPUTE
71525   00020   11 20000 00004       TALLY
71526   00021   23 00004 00032        FOR
71527   00022   36 00037 00004          TAPE
71530   00023   73 01575 01577           BLOCKS
71531   00024   34 01575 00017      COMPUTE REMAINDER
71532   00025   13 20000 01576        STORED FROM LAST BLOCK
71533   00026   21 10000 00033      SET
71534   00027   31 20000 00017       BACKUP
71535   00030   35 00006 01571        COMMAND
71536   00031   45 00000 01552      JUMP
71537   00032   00 00000 00033      27
```

MAGNETIC TAPE STORAGE

71540	00033	00	00001	00003	
71541	01551	23	00002	01576	CHANGE LAST BLOCK STORAGE
71542	01552	75	30034	01554	LOAD BLOCK OF
71543	01553	11	30000	00004	INFORMATION
71544	01554	16	01573	00000	SET F1
71545	01555	65	00001	00000	DUMP MT
71546	01556	21	01553	01574	STEP
71547	01557	21	00003	01575	COMMANDS
71550	01560	41	01577	01552	INDEX TALLY
71551	01561	37	01561	01551	SWITCH
71552	01562	75	30004	01564	SET AND
71553	01563	11	01570	00001	DUMP
71554	01564	16	01573	00000	TRAILER
71555	01565	65	00001	00000	BLOCK
71556	01566	75	30030	00002	RESTORE
71557	01567	11	77027	01551	FLIP
71560	01570	45	00000	01624	F2 FOR TRAILER BLOCK
71561	01571	00	30000	00000	BACKUP COMMAND
71562	01572	37	71562	00004	EXIT
71563	01573	57	77777	00002	MANUAL ENTRANCE STOP
71564	01574	00	00034	00000	CONSTANTS
71565	01575	00	00000	00034	
71566	01576	00	00000	00*0	TEMPORARIES
71567	01577	00	00000	0000	

PX 71900-8-11

INTEGRATE DIFFERENTIAL EQUATIONS

71600	01000	37	00000	71601	BASIC SUBROUTINE
71601	71601	11	00040	00002	SAVE
71602	71602	16	00000	00002	LOCATION
71603	71603	11	00013	00003	SAVE X AND Y ADDRESSES
71604	71604	75	30025	71606	LOAD SUBROUTINE
71605	71605	11	71646	00013	INTO TEMPORARIES
71606	71606	75	20004	71610	MODIFY V
71607	71607	21	00014	00002	ADDRESSES
71610	71610	21	00024	00002	WITH LOCATION
71611	71611	21	00026	00002	MODIFY
71612	71612	21	00027	00002	Y
71613	71613	21	00032	00002	ADDRESSES
71614	71614	21	00035	00002	WITH
71615	71615	21	00037	00002	LOCATION
71616	71616	31	00002	00017	MODIFY U
71617	71617	35	00024	00024	ADDRESS WITH LOCATION
71620	71620	16	00003	00002	MODIFY SUBROUTINE
71621	71621	71	00075	00002	REFERENCES WITH
71622	71622	35	00036	00036	Y ADDRESS
71623	71623	54	00003	00071	MODIFY SUBROUTINE
71624	71624	21	00023	00003	ADDRESSES
71625	71625	21	00033	00003	WITH
71626	71626	21	00034	00003	X ADDRESS
71627	71627	54	00003	00014	MODIFY SUBROUTINE
71630	71630	21	00023	00003	ADDRESSES
71631	71631	21	00027	00003	WITH
71632	71632	21	00035	00003	X ADDRESS
71633	71633	31	00003	00003	PICK
71634	71634	15	20000	71635	UP
71635	71635	31	00000	00000	N-1
71636	71636	35	00074	00002	N
71637	71637	35	00023	00023	MODIFY

PX 71900-8-11

INTEGRATE DIFFERENTIAL EQUATIO

```
71640  71640  31 00002 00001      COMMANDS
71641  71641  35 00033 00033       WITH
71642  71642  31 00002 00015        QUANTITY
71643  71643  35 00027 00027          N
71644  71644  75 30002 00013    SHIFT
71645  71645  11 00015 71714      COMMANDS
71646  71646  75 30021 71711    LOAD PART OF
71647  71647  11 00016 77777       SUBROUTINE INTO PLACE
71650  71650  75 30016 00001    LOAD REMAINDER OF
71651  71651  11 71673 00020      SUBROUTINE INTO PLACE
71652  71652  75 30042 00001    SAVE EXIT
71653  00000  11 01736 71716      AND COUNTERS
71654  00001  11 00040 01776    CLEAR
71655  00002  11 00040 01777     COUNTERS
71656  00003  14 22000 14002    DELTA X TIMES YI EQUALS K
71657  00004  47 00005 00015    ZERO ?
71660  00005  11 20000 00036    STORE K
71661  00006  14 22003 62022    K TIMES BJ
71662  00007  14 23400 22026     PLUS Q2 TIMES CJ
71663  00010  11 20000 00037    STORE INCREMENT
71664  00011  14 03003 70037    THREE TIMES INCREMENT
71665  00012  14 23003 62032     PLUS K TIMES AJ
71666  00013  14 00177 44002       ADDED TO QI
71667  00014  14 00003 74002    ADD INCREMENT TO YI
71670  00015  14 40000 00003    INDEX N-1
71671  00016  37 00001 00002    EXECUTE SUBROUTINE
71672  00017  14 44004 30003    INDEX 3
71673  00020  75 30042 01735    RESTORE AND
71674  00021  11 71716 01736    EXIT
71675  00022  20 00000 00000    C
71676  00023  22 57541 46376     O
71677  00024  33 24047 46001      N
```

PX 71900-8-11

INTEGRATE DIFFERENTIAL EQUATIO

71700	00025	25 25252 52775		S
71701	00026	40 00000 00000		T
71702	00027	55 20236 31376		A
71703	00030	44 53730 32001		N
71704	00031	52 52525 25376		T
71705	00032	60 00000 00000		S
71706	00033	55 20236 31376		
71707	00034	44 53730 32001		
71710	00035	60 00000 00000		
71711	71711	11 00036 71713	SHIFT COMMAND	
71712	71712	37 00017 00017	EXECUTE SUBROUTINE	
71713	71713	00 00000 00*0	TEMPORARIES	
71714	71714	00 00000 00*0	FOR	
71715	71715	00 00000 0000	COMMANDS	

PX 71900-8-11

CARD TO PAPER TAPE

71724	71724	17	00000	73376	PICK FIRST CARD
71725	71725	23	00122	00122	LOAD
71726	71726	37	76645	76575	FLIP
71727	71727	14	57101	70530	READ FIRST CARD
71730	71730	23	70530	00074	N-1
71731	71731	46	71732	71734	DONE ?
71732	71732	37	73374	73374	FINISH
71733	71733	57	00000	00000	OFF
71734	71734	75	30006	71736	SEPARATE ADDRESS
71735	71735	73	71765	00002	DIGITS AND N-1
71736	71736	75	20004	71740	ASSEMBLE
71737	71737	32	00002	00003	ADDRESS
71740	71740	35	00006	71760	IN CELL
71741	71741	16	20000	71756	SET STORE
71742	71742	31	20000	00017	COMPUTE
71743	71743	15	20000	71760	LAST
71744	71744	21	71760	00007	ADDRESS
71745	71745	54	00007	10017	SET BLOCK
71746	71746	35	71763	71755	TRANSFER FOR STORE
71747	71747	11	00007	20000	COMPUTE NUMBER
71750	71750	73	71765	00010	OF REMAINING CARDS
71751	71751	47	71752	71773	NONE ?
71752	71752	31	00010	00017	READ
71753	71753	35	71762	71754	REMAINING
71754	71754	14	57000	70537	CARDS
71755	71755	75	30001	71757	STORE
71756	71756	11	70531	30000	RESULTS
71757	71757	37	71007	71003	PUNCH OUT
71760	71760	00	30000	30000	TAPE
71761	71761	45	00000	71727	RETURN
71762	71762	14	57000	70537	PROTOTYPE
71763	71763	75	30001	71757	COMMANDS

CARD TO PAPER TAPE

71764	71764	00	00000	00*06	6
71765	71765	00	00036	41100	10 TO THE SIXTH
71766	71766	00	00003	03240	10 TO THE FIFTH
71767	71767	00	00000	23420	10 TO THE FOURTH
71770	71770	00	00000	01750	10 TO THE THIRD
71771	71771	00	00000	00144	10 TO THE SECCOND
71772	71772	00	00000	00001	10 TO THE ZERO

BESSEL FUNCTIONS

72000	01110	16	01122	01735	STOP TRACE
72001	01111	11	00004	00037	SHIFT
72002	01112	55	00037	00006	COMMAND CODE
72003	01113	45	00000	01120	JUMP
72004	01114	75	30670	72006	SAVE
72005	01115	11	01110	75110	ES
72006	01116	75	30441	01110	LOAD
72007	01117	11	72000	01110	THIS
72010	01120	55	00037	10042	TEST
72011	01121	44	01210	01123	ORDER
72012	01122	00	00000	01736	
72013	01123	45	00000	01124	
72014	01124	16	00013	75734	SET STORE
72015	01125	27	00005	01772	RESTORE SIGN
72016	01126	11	00005	01117	OF X
72017	01127	11	00005	20000	TEST
72020	01130	14	45123	01157	RANGE
72021	01131	11	01776	01110	SAVE AND CLEAR
72022	01132	11	00040	01776	INDEX COUNTER
72023	01133	14	21177	41116	COMPUTE
72024	01134	14	30123	21116	POLYNOMIAL
72025	01135	11	00040	20000	FOR
72026	01136	14	12111	65240	FUNCTION OF
72027	01137	14	40123	11136	FIRST
72030	01140	11	20000	01115	KIND (J)
72031	01141	55	00037	10043	TEST FOR
72032	01142	44	01143	01204	OTHER FUNCTION
72033	01143	14	22111	71233	COMPUTE
72034	01144	14	54177	41774	LOG 1/2 J (X)
72035	01145	14	21111	51114	OVER
72036	01146	14	30123	41114	1/2 PI
72037	01147	11	00040	20000	COMPUTE

PX 71900-8-11

BESSEL FUNCTIONS

```
72040   01150   14 12111 65247        POLYNOMIAL
72041   01151   14 40123 11150
72042   01152   11 20000 00037      STORE Y
72043   01153   14 00111 40037        FUNCTION
72044   01154   45 00000 01204
72045   01155   00 00000 00 0
72046   01156   00 00000 00 0
72047   01157   11 00037 01111      SET INDICATOR
72050   01160   14 32111 71230      3 DIVIDED
72051   01161   11 20000 01116           BY X
72052   01162   11 00040 20000      COMPUTE FIRST
72053   01163   14 12111 65256        POLYNOMIAL
72054   01164   14 40123 11163          IN
72055   01165   11 20000 01112            X/3
72056   01166   11 00040 20000      COMPUTE SECOND
72057   01167   14 12111 65265        POLYNOMIAL
72060   01170   14 40123 11167          IN
72061   01171   11 20000 01114            X/3
72062   01172   14 50111 71113      SQUARE ROOT OF X
72063   01173   14 30111 31112      COMPUTE
72064   01174   14 04111 41117       FUNCTION
72065   01175   14 61111 71115        OF FIRST
72066   01176   14 20111 21115          KIND   J
72067   01177   55 01111 10043      TEST FOR
72070   01200   44 01201 01204       OTHER FUNCTION
72071   01201   14 60111 70037      SIN  (COS)
72072   01202   14 20111 20037        TIMES POLYNOMIAL
72073   01203   45 00000 01204      (REVERSE SIGN)
72074   01204   11 01115 20000      STORE J
72075   01205   11 01110 01776      RESTORE INDEX COUNTER
72076   01206   75 30670 01734      RESTORE ES
72077   01207   11 75110 01110        AND EXIT
```

PX 71900-8-11

BESSEL FUNCTIONS

72100	01210	11	01217	01140	CHANGES
72101	01211	11	01220	01152	FOR
72102	01212	11	01221	01175	SECOND
72103	01213	11	01222	01201	ORDER
72104	01214	11	01223	01203	
72105	01215	75	30036	01123	
72106	01216	11	01276	01240	
72107	01217	14	21111	71115	PROTOTYPE
72110	01220	14	31111	70037	COMMANDS
72111	01221	14	60111	71115	
72112	01222	14	61111	70037	
72113	01223	27	00037	01772	
72114	01224	00	00000	00*0	
72115	01225	00	00000	00*0	
72116	01226	00	00000	00*0	
72117	01227	00	00000	00*0	
72120	01230	30	00000	00002	
72121	01231	00	00000	00006	
72122	01232	22	00000	00004	
72123	01233	20	00000	00000	
72124	01234	31	10375	52401	
72125	01235	00	00000	0000	
72126	01236	00	00000	0000*	
72127	01237	00	00000	0000	
72130	01240	33	41467	10363	P
72131	01241	57	65777	45770	O
72132	01242	26	60360	01373	L
72133	01243	53	60051	15376	Y
72134	01244	24	17775	63001	N
72135	01245	56	00000	50002	O
72136	01246	37	77777	77777	M
72137	01247	57	55704	15364	I

BESSEL FUNCTIONS

72140	01250	21	41603	10370	A
72141	01251	52	13536	72773	L
72142	01252	20	14225	71776	
72143	01253	50	15233	55777	C
72144	01254	23	30202	76400	O
72145	01255	27	41111	75776	E
72146	01256	22	76253	35763	F
72147	01257	50	11126	20765	F
72150	01260	26	37021	41766	I
72151	01261	47	04115	32762	C
72152	01262	51	27014	73370	E
72153	01263	46	12340	02753	N
72154	01264	31	42042	46400	T
72155	01265	56	16530	25363	S
72156	01266	23	16241	27764	
72157	01267	21	57052	60765	
72160	01270	52	37270	75767	
72161	01271	24	56574	01761	
72162	01272	25	25175	32773	
72163	01273	31	10375	52400	
72164	01274	00	00000	00 0	
72165	01275	00	00000	00-0	
72166	01276	27	20362	77757	
72167	01277	53	13661	55764	
72170	01300	22	12104	53770	
72171	01301	53	60204	30773	
72172	01302	32	77770	44775	
72173	01303	56	00000	12000	
72174	01304	37	77777	77776	
72175	01305	26	65262	11367	
72176	01306	53	36051	71773	
72177	01307	23	77444	00376	

PX 71900-8-11

BESSEL FUNCTIONS

72200	01310	52	73727	73001	
72201	01311	21	26117	15402	
72202	01312	34	24112	15375	
72203	01313	53	50147	60400	
72204	01314	45	57405	13763	
72205	01315	22	47571	71366	
72206	01316	53	43660	00367	
72207	01317	26	32676	11763	
72210	01320	20	77533	65372	
72211	01321	32	13023	04354	
72212	01322	31	42042	46400	
72213	01323	23	07237	04764	
72214	01324	45	65760	24765	
72215	01325	47	50636	33365	
72216	01326	32	10122	54770	
72217	01327	42	30131	30761	
72220	01330	40	00040	42774	
72221	01331	31	10375	52400	
72222	01332	00	00000	00*0	
72223	01333	00	00000	0000	
72224	01334	11	00005	20000	F
72225	01335	46	76045	01336	L
72226	01336	42	00073	01354	I
72227	01337	54	00010	00107	P
72230	01340	43	20000	01344	S
72231	01341	46	01342	01343	U
72232	01342	35	00074	00010	B
72233	01343	54	00005	00107	R
72234	01344	11	00005	00006	O
72235	01345	11	00070	00005	U
72236	01346	31	00006	00042	T
72237	01347	73	00005	00004	I

BESSEL FUNCTIONS

72240	01350	54	00005	00107	N
72241	01351	23	10000	00005	E
72242	01352	21	00005	00004	S
72243	01353	44	01346	01354	
72244	01354	45	00000	01721	
72245	01355	36	00074	00010	
72246	01356	13	00005	20000	
72247	01357	42	00066	01402	
72250	01360	36	00066	00006	
72251	01361	54	00006	00001	
72252	01362	11	01377	01364	
72253	01363	11	00040	00005	
72254	01364	21	00005	30000	
72255	01365	71	20000	00006	
72256	01366	54	20000	00045	
72257	01367	11	20000	00005	
72260	01370	21	01364	00074	
72261	01371	42	01400	01364	
72262	01372	54	00005	00101	
72263	01373	71	00010	01403	
72264	01374	35	00005	00005	
72265	01375	11	00067	00010	
72266	01376	45	00000	01401	
72267	01377	21	00005	01404	
72270	01400	21	00005	01414	
72271	01401	45	00000	01721	
72272	01402	45	00000	01737	
72273	01403	00	13056	20577	
72274	01404	77	62620	75765	
72275	01405	01	11721	41642	
72276	01406	74	74607	70746	
72277	01407	05	27266	02203	

PX 71900-8-11

BESSEL FUNCTIONS

72300	01410	70	22764	23456
72301	01411	12	47414	37545
72302	01412	60	00203	77320
72303	01413	37	77774	20006
72304	01414	11	01470	00004
72305	01415	45	00000	01417
72306	01416	11	00040	00004
72307	01417	23	00010	00056
72310	01420	46	01421	76045
72311	01421	35	00053	20000
72312	01422	46	01424	01425
72313	01423	00	00000	00036
72314	01424	11	00040	00005
72315	01425	36	01423	10000
72316	01426	35	01466	01427
72317	01427	11	00010	10000
72320	01430	44	01431	01432
72321	01431	11	20000	20000
72322	01432	73	01476	10000
72323	01433	35	00004	20000
72324	01434	11	00066	00004
72325	01435	42	01470	01441
72326	01436	55	00004	00001
72327	01437	36	01467	20000
72330	01440	45	01475	01435
72331	01441	54	20000	00043
72332	01442	73	01470	00005
72333	01443	71	00005	10000
72334	01444	54	20000	00045
72335	01445	11	20000	00006
72336	01446	11	00040	00007
72337	01447	15	01440	01453

PX 71900-8-11

BESSEL FUNCTIONS

72340	01450	11	00044	00010
72341	01451	71	00007	00006
72342	01452	54	20000	00045
72343	01453	35	01476	00007
72344	01454	23	01453	00073
72345	01455	41	00010	01451
72346	01456	71	00007	00005
72347	01457	54	20000	00045
72350	01460	11	20000	00005
72351	01461	11	00004	10000
72352	01462	44	01463	01464
72353	01463	13	00005	00005
72354	01464	11	00074	00010
72355	01465	45	00000	01721
72356	01466	54	00005	24110
72357	01467	14	44176	65200
72360	01470	06	22077	32504
72361	01471	31	10375	52202
72362	01472	65	52420	76452
72363	01473	01	21464	25731
72364	01474	77	73155	46346
72365	01475	00	00117	32757
72366	01476	31	10375	52421
72367	01477	44	01605	01334
72370	01500	27	00005	01772
72371	01501	44	01502	01675
72372	01502	16	01712	01734
72373	01503	44	01504	01657
72374	01504	11	01774	00007
72375	01505	45	00000	01657
72376	01506	33	00051	00107
72377	01507	35	00012	00012

PX 71900-8-11

BESSEL FUNCTIONS

72400	01510	42	00057	01726
72401	01511	45	00000	01607
72402	01512	44	01514	01513
72403	01513	44	01516	01515
72404	01514	44	01520	01517
72405	01515	44	01522	01521
72406	01516	44	01524	01523
72407	01517	44	01526	01525
72410	01520	44	01530	01527
72411	01521	44	01414	01416
72412	01522	44	01737	01737
72413	01523	44	01737	01737
72414	01524	44	01737	01737
72415	01525	44	01737	01737
72416	01526	44	01737	01737
72417	01527	44	01737	01737
72420	01530	44	01737	01737
72421	01531	44	01737	01737
72422	01532	44	01535	01534
72423	01533	44	00700	01737
72424	01534	44	01737	01355
72425	01535	44	73400	73000
72426	01536	44	01512	01537
72427	01537	44	01546	01540
72430	01540	44	01542	01541
72431	01541	44	01545	01543
72432	01542	44	01531	01544
72433	01543	44	01571	01556
72434	01544	44	01574	01563
72435	01545	44	01737	01552
72436	01546	44	01532	01547
72437	01547	44	01533	01477

BESSEL FUNCTIONS

72440	01550	54	00005	00013
72441	01551	23	00011	00057
72442	01552	54	00005	20013
72443	01553	43	20002	01550

PX 71900-8-11

CHARACTRON OUTPUT

```
72444   00774   75 31700 72446      SAVE
72445   00775   11 00100 74100       ES
72446   00776   75 30330 01000      LOAD
72447   00777   11 72450 01000        THIS
72450   01000   31 01736 00017      PICK
72451   01001   15 20000 01306       UP
72452   01002   21 01306 01227        COMMAND
72453   01003   42 01230 01005         FROM
72454   01004   23 01306 01231          PROPER
72455   01005   15 20000 01006           PLACE
72456   01006   11 30000 00002            ON MD
72457   01007   31 00002 00017      SET DATA
72460   01010   15 20000 01306       PICKUP
72461   01011   21 01306 01231        COMMAND
72462   01012   42 01230 01014         FROM
72463   01013   23 01306 01231          PROPER PLACE
72464   01014   15 20000 01125           ON MD
72465   01015   55 00002 10025      EXTRACT
72466   01016   51 00045 00003        M
72467   01017   55 10000 00036      TEST
72470   01020   44 01021 01022        L
72471   01021   54 00003 00005      32 TIMES M TO N
72472   01022   11 01232 20000      TEST
72473   01023   47 01310 01042        L
72474   01024   00 00000 00*0
72475   01025   44 01030 01026      TEST TO SKIP COLUMN REMAINDER
72476   01026   32 01235 00103      MOVE TO
72477   01027   31 20000 00005        NEXT COLUMN
72500   01030   45 00000 01035      JUMP
72501   01031   32 00043 00106      MOVE TO
72502   01032   31 20000 00002       NEXT ROW
72503   01033   44 01035 01034      TEST TO SKIP ROW
```

PX 71900-8-11

CHARACTRON OUTPUT

72504	01034	32	00044	00000	SKIP ROW
72505	01035	11	20000	01232	STORE L
72506	01036	44	01041	01037	TEST FOR NEW PAGE
72507	01037	35	00003	20000	TEST WHETHER DATA
72510	01040	42	01245	01042	WILL FIT PAGE
72511	01041	37	01075	01052	TURN PAGE
72512	01042	11	01232	20000	DONE
72513	01043	42	01234	01045	PAGE?
72514	01044	37	01075	01052	TURN PAGE
72515	01045	41	00003	01077	DONE DATA?
72516	01046	45	00000	01312	JUMP
72517	01047	00	00000	00140	CONSTANT
72520	01050	75	31700	01735	RESTORE ES
72521	01051	11	74100	00100	AND EXIT
72522	01052	17	00000	01236	START CHARACTRON PAGE
72523	01053	15	01226	01055	PRESET PICKUP
72524	01054	11	00040	00017	SET POSITION
72525	01055	11	30000	10000	PICKUP TITLE CHARACTERS
72526	01056	11	01242	00004	SET TALLY
72527	01057	51	01243	20000	XTRACT CHARACTER
72530	01060	55	10000	00006	SHIFT CHARACTERS
72531	01061	35	00017	20000	ADD POSITION
72532	01062	77	10000	20000	PRINT
72533	01063	21	00017	01244	STEP POSITION
72534	01064	41	00004	01057	INDEX TALLY
72535	01065	21	01055	00073	STEP PICKUP
72536	01066	42	01241	01055	DONE
72537	01067	21	01233	01246	STEP PAGE NO.
72540	01070	35	01247	10000	ROUND PAGE NO.
72541	01071	11	01250	00005	LOCATE PAGE NO.
72542	01072	11	00041	00004	PRINT THREE
72543	01073	37	01226	01214	CHARACTERS

PX 71900-8-11

CHARACTRON OUTPUT

72544	01074	11	00040	01232	CLEAR L
72545	01075	75	10002	30000	STOP AND
72546	01076	17	00000	01237	MOVE FILM
72547	01077	17	00000	01236	START CHARACTRON WORD
72550	01100	55	00002	10014	TEST
72551	01101	44	01102	01111	FORMAT
72552	01102	54	01232	20106	COMPUTE
72553	01103	71	20000	01235	HORIZONTAL
72554	01104	35	01251	00005	FORMAT
72555	01105	31	01232	00042	WORD
72556	01106	31	20000	00075	POSITION
72557	01107	35	00005	00005	
72560	01110	45	00020	01120	JUMP
72561	01111	54	01232	20103	COMPUTE
72562	01112	31	20000	00027	VERTICLE
72563	01113	35	01251	00005	FORMAT
72564	01114	55	01232	10037	WORD
72565	01115	31	10000	00005	POSITION
72566	01116	34	10000	00051	
72567	01117	35	00005	00005	
72570	01120	75	30003	01122	POSITION SIGNS
72571	01121	35	01252	00006	AND DECIMAL PT.
72572	01122	21	01130	00073	STEP PICKUP FROM BIN
72573	01123	42	01255	01127	DONE BIN
72574	01124	75	30020	01126	PICKUP DATA
72575	01125	11	30000	00020	INTO BIN
72576	01126	15	01110	01130	RESET PICKUP FROM BIN
72577	01127	21	01125	00073	STEP PICKUP
72600	01130	11	30000	00011	PICKUP FROM BIN
72601	01131	54	00011	10034	SEPARATE
72602	01132	54	10000	00054	EXPONENT
72603	01133	23	20000	01256	S-120 TO S1

CHARACTRON OUTPUT

72604	01134	73	01257	00012	DIVIDED BY 10 TO S2
72605	01135	35	01260	01154	SET SHIFT OF T PLUS 3
72606	01136	71	00043	00012	3xS2 PLUS
72607	01137	35	01261	00013	36 TO P
72610	01140	15	01164	01147	PRESET MULTIPLY
72611	01141	55	00012	00036	SET S2 TO TEST DIGITS
72612	01142	71	00011	01262	N TIMES
72613	01143	54	20000	00045	$R^{-12} \times 2^{-3}$
72614	01144	11	20000	00011	TO N1
72615	01145	55	00012	00001	MULTIPLY
72616	01146	44	01152	01147	N1
72617	01147	71	30000	00011	BY R TO
72620	01150	54	20000	00045	THE MINUS S2
72621	01151	11	20000	00011	AND
72622	01152	21	01147	00073	STORE
72623	01153	42	01263	01145	AS N2
72624	01154	54	00011	20003	$2^{t+3} \times$ N2 TO N2
72625	01155	47	01156	01176	ZERO ?
72626	01156	43	20000	01167	OVERFLOW ?
72627	01157	21	00013	00074	STEP P
72630	01160	54	00011	20003	DIVIDE N2
72631	01161	73	01257	00011	BY 10
72632	01162	23	01154	00043	COMPENSATE SHIFT
72633	01163	42	01307	01165	RIGHT SHIFT
72634	01164	45	01264	01154	JUMP
72635	01165	35	00077	01166	MAKE PROPER
72636	01166	00	00000	00*0	RIGHT SHIFT
72637	01167	21	20000	20000	SHIFT LEFT AND REMOVE SIGN
72640	01170	46	01171	01173	TEST SIGN
72641	01171	13	20000	20000	COMPLEMENT N2
72642	01172	77	10000	00006	PRINT MINUS
72643	01173	35	01247	10000	ROUND

CHARACTRON OUTPUT

```
72644   01174   31 10000 00010      OVER
72645   01175   43 20000 01157        FLOW?
72646   01176   11 01271 00004      PRINT 6
72647   01177   37 01226 01214        DIGITS
72650   01200   77 10000 00010      PRINT DECIMAL PT.
72651   01201   71 00013 01272      SCALE P INTO A
72652   01202   46 01203 01205      TEST SIGN
72653   01203   13 20000 20000      CHANGE SIGN
72654   01204   77 10000 00007      PRINT MINUS
72655   01205   35 01247 10000      ROUND TO Q
72656   01206   21 00005 01273      STEP POSITION
72657   01207   37 01226 01214      PRINT 2
72660   01210   37 01226 01214        DIGITS
72661   01211   17 00000 01237      STOP CHARACTRON
72662   01212   21 01232 00074      STEP L
72663   01213   45 00000 01042      EXIT
72664   01214   31 10000 00002      10 Q                DIGIT
72665   01215   32 10000 00001       TO A
72666   01216   11 20000 10000      REPLACE
72667   01217   34 20000 00102        REMAINDER
72670   01220   42 01274 01222      SHIFT
72671   01221   32 01275 00000      CORRECT
72672   01222   32 00005 00000        CHARACTRON CODE
72673   01223   77 10000 20000      PRINT DIGIT
72674   01224   21 00005 01244      STEP POSITION
72675   01225   41 00004 01214      DONE ALL?
72676   01226   45 01276 30000      EXIT
72677   01227   00 73777 00000
72700   01230   00 76000 00000
72701   01231   00 74000 00000
72702   01232   00 00000 00200      L   POSITION INDEX OF WORDS
72703   01233   77 73716 66217      PAGE NO.
```

PX 71900-8-11

CHARACTRON OUTPUT

72704	01234	00	00000	00200	
72705	01235	00	00000	00037	
72706	01236	00	00000	05000	CHARACTRON
72707	01237	00	00000	04400	OPERATE
72710	01240	00	00000	06000	CODES
72711	01241	11	01306	00000	
72712	01242	00	00000	00005	
72713	01243	77	00000	00000	
72714	01244	00	00024	00000	
72715	01245	00	00000	00201	
72716	01246	00	04061	11560	$10^{-3} \cdot 2^{36}$
72717	01247	00	00000	06554	
72720	01250	00	00764	01777	
72721	01251	00	00040	00040	
72722	01252	40	01750	00000	
72723	01253	00	00264	00000	
72724	01254	70	01530	00000	
72725	01255	11	00040	00000	
72726	01256	00	00000	00170	
72727	01257	00	00000	00012	
72730	01260	54	00011	10003	
72731	01261	00	00000	00044	
72732	01262	05	24220	44463	$R^{-12} \cdot 2^{-3}$
72733	01263	71	01271	00000	
72734	01264	25	71230	64027	R^{16}
72735	01265	32	36041	57154	R^{8}
72736	01266	35	06512	24172	R^{4}
72737	01267	36	41100	00000	R^{2}
72740	01270	37	20000	00000	R^{1}
72741	01271	00	00000	00006	
72742	01272	00	50753	41100	
72743	01273	00	00050	00000	

CHARACTRON OUTPUT

72744	01274	06	00000	00000	
72745	01275	02	00000	00000	
72746	01276	77	77777	77777	TITLE
72747	01277	77	77777	77777	
72750	01300	77	77777	77777	
72751	01301	23	33014	17777	CHARAC-
72752	01302	00	50454	15045	
72753	01303	77	77777	77777	
72754	01304	77	77777	77777	
72755	01305	77	77777	77777	TERS
72756	01306	00	00000	00*0	
72757	01307	54	00011	10000	
72760	01310	55	00002	10014	TEST FOR HORIZONTAL
72761	01311	44	01031	01025	OR VERTICAL FORMAT
72762	01312	11	01232	20000	
72763	01313	55	00002	10014	
72764	01314	44	01315	01316	
72765	01315	36	01323	20000	
72766	01316	42	01047	01321	
72767	01317	44	01321	01320	
72770	01320	37	01075	01052	
72771	01321	75	30002	01050	
72772	01322	11	01232	72702	
72773	01323	00	00000	00020	
72774	01324	00	00000	0000	
72775	01325	00	00000	0000	
72776	01326	00	00000	0000*	
72777	01327	00	00000	0000	

PX 71900-8-11

FLIP CARD WRITE

73000		75	31700	73002	STORE
73001		11	00100	74100	ES
73002		75	30307	00100	LOAD
73003		11	73004	00100	THIS
73004	00100	17	00000	00077	PRIME
73005	00101	11	00352	00025	PRESET TO
73006	00102	11	00067	00026	EIGHT FIELDS
73007	00103	75	10004	00105	CLEAR
73010	00104	11	00040	00002	CELLS
73011	00105	16	01736	00005	PICKUP
73012	00106	21	00005	00403	
73013	00107	42	00405	00111	THIS
73014	00110	23	00005	00404	
73015	00111	16	00005	00112	FLIP
73016	00112	21	00004	30000	COMMAND
73017	00113	16	00004	00005	SET
73020	00114	21	00005	00404	
73021	00115	42	00405	00117	
73022	00116	23	00005	00404	PICKUP
73023	00117	31	00005	00017	
73024	00120	15	20000	00137	ADDRESS
73025	00121	55	00004	10015	TEST
73026	00122	44	00131	00123	OPTION
73027	00123	11	00074	00003	SET
73030	00124	23	00026	00041	FOR
73031	00125	23	00025	00073	SIX
73032	00126	44	00131	00127	TEST
73033	00127	23	00025	00073	SET TO
73034	00130	11	00041	00003	OMIT NUMBER
73035	00131	55	00004	00025	SET CARD
73036	00132	51	00064	00005	COUNT
73037	00133	47	00342	00343	TEST NO CARDS

FLIP CARD WRITE

73040	00134	47	00135	00302	TEST LAST CARD
73041	00135	17	00000	00406	PUNCH AND PICK
73042	00136	75	30010	00140	PICKUP
73043	00137	11	30000	00011	DATA
73044	00140	21	00137	00025	STEP
73045	00141	75	10044	00143	CLEAR
73046	00142	11	00040	00407	IMAGE
73047	00143	11	00026	00010	PRESETS
73050	00144	11	00364	00023	FOR
73051	00145	11	00365	00024	EACH
73052	00146	15	00353	00166	CARD
73053	00147	31	00003	00000	TEST
73054	00150	47	00151	00166	FOR
73055	00151	43	00041	00346	OPTION
73056	00152	21	00166	00073	SEVEN FIELDS
73057	00153	11	00011	20000	TEST
73060	00154	46	00155	00157	CARD
73061	00155	13	00011	00011	NUMBER
73062	00156	11	00074	00002	SIGN
73063	00157	31	00011	00000	TEST
73064	00160	42	00374	00162	FOR
73065	00161	11	00374	20000	SIZE
73066	00162	32	00040	00043	CONVERT
73067	73067	73	00375	20000	NUMBER
73070	00164	37	00260	00233	PLACE IN IMAGE
73071	00165	11	00377	00023	SET BIT
73072	00166	11	30000	00006	PICKUP
73073	00167	21	00166	00073	STEP
73074	00170	54	00006	10034	CONVERT
73075	00171	54	10000	00054	NEGATIVE
73076	00172	23	20000	00354	BINARY EXPONENT
73077	00173	73	00060	00021	AND BASE

PX 71900-8-11

FLIP CARD WRITE

73100	00174	35	00372	00217	TO 2
73101	00175	71	00043	00021	EXP 10
73102	00176	35	00362	00007	
73103	00177	16	00353	00212	PRESET
73104	00200	55	00021	00036	FOR LOOP
73105	00201	71	00006	00351	CONVERT
73106	00202	54	20000	00045	FRACTION
73107	00203	11	20000	00006	ALSO
73110	00204	11	00044	00022	SET INDEX
73111	00205	44	00206	00210	TEST AND
73112	00206	11	00074	00002	SET FOR
73113	00207	13	00006	00006	SIGN
73114	00210	55	00021	00001	MULTIPLY
73115	00211	44	00215	00212	
73116	00212	71	00006	30000	
73117	00213	54	20000	00045	BY
73120	00214	11	20000	00006	
73121	00215	21	00212	00074	
73122	00216	41	00022	00210	R EXP P
73123	00217	30	00000	0000	SHIFT
73124	00220	47	00221	00235	TEST ZERO
73125	00221	42	00230	00233	TEST NORMALIZED
73126	00222	21	00007	00074	NORMALIZE
73127	00223	54	00006	20003	IF
73130	00224	73	00060	00006	NECESSARY
73131	00225	23	00217	00043	
73132	00226	42	00373	00231	TEST RIGHT SHIFT
73133	00227	45	00000	00217	RETURN
73134	00230	37	77777	74512	UNITY LESS ROUNDING
73135	00231	35	00077	00232	RIGHT
73136	00232	30	00000	0000	SHIFT
73137	00233	31	20000	00001	SCALE 2 EXP 36

FLIP CARD WRITE

```
73140   00234   35 00371 00006      ROUND
73141   00235   11 00363 00022      SET INDEX
73142   00236   31 00006 00002      PLACE
73143   00237   32 00006 00001        BIT
73144   00240   11 20000 00006         IN
73145   00241   34 20000 00063           IMAGE
73146   00242   35 00024 00243            FOR
73147   00243   30 00000 0000              DIGIT
73150   00244   37 00244 00245      SWITCH
73151   00245   55 00023 00043      SHIFT BIT
73152   00246   44 00247 00250      TEST FIELD
73153   00247   21 00024 00366      ADVANCE FIELD
73154   00250   37 00250 00251      SWITCH
73155   00251   41 00022 00236      TEST END WORD
73156   00252   31 00002 00000      TEST
73157   00253   47 00254 00256        SIGN
73160   00254   11 00400 20000         AND
73161   00255   37 00244 00242           SET SIGN
73162   00256   55 00023 00043      SHIFT BIT
73163   00257   11 00040 00002      CLEAR
73164   00260   37 00260 00261      SWITCH
73165   00261   11 00007 20000      TEST
73166   00262   46 00263 00265        SIGN
73167   00263   13 00007 00007         AND
73170   00264   11 00074 00002          SET SIGN
73171   00265   31 00007 00000      EXP
73172   00265   73 00060 10000       OVER
73173   00267   11 20000 00007        NINE
73174   00270   31 00003 00000      TEST
73175   00271   47 00272 00275      OPTION
73176   00272   31 10000 00017      SET EXP
73177   00273   37 00250 00242        FOR SIX
```

FLIP CARD WRITE

73200	00274	45	00000	00307	OPTION
73201	00275	55	10000	00042	TEST FOR
73202	00276	44	00277	00301	12 ROW BIT
73203	00277	11	00400	20000	AND
73204	00300	37	00244	00242	PLACE IN IMAGE
73205	00301	44	00305	00307	TEST FOR 11 ROW BIT
73206	00302	17	00000	00304	PUNCH WITHOUT PICKING
73207	00303	45	00000	00136	RETURN
73210	00304	00	00000	00102	CONSTANT
73211	00305	11	00401	20000	PLACE BIT
73212	00306	37	00244	00242	IN IMAGE
73213	00307	31	00007	00017	PLACE LAST
73214	00310	37	00250	00242	DIGIT EXP IN IMAGE
73215	00311	37	00260	00252	DO SIGN
73216	C0312	31	00003	00000	TEST
73217	00313	47	00314	00315	OPTION
73220	00314	55	00023	00043	SKIP PERIOD POSITION
73221	00315	41	00010	00166	INDEX
73222	00316	11	00057	00022	
73223	00317	31	00003	00000	TEST
73224	00320	47	00321	00327	OPTION
73225	00321	21	00407	00376	PLACE
73226	00322	21	00421	00376	
73227	00323	21	00414	00376	PERIODS
73230	00324	21	00423	00376	
73231	00325	21	00435	00376	IN
73232	00326	21	00430	00376	IMAGE
73233	00327	16	00367	00334	PRESET
73234	00330	16	00402	00335	COMMANDS
73235	00331	15	00367	00332	TO PUNCH
73236	00332	55	30000	00010	PICKUP
73237	00333	77	00000	10000	AND

PX 71900-8-11

FLIP CARD WRITE

73240	00334	77	10000	30000	PUNCH
73241	00335	77	10000	30000	DATA
73242	00336	23	00332	00073	STEP
73243	00337	23	00334	00074	ALL
73244	00340	23	00335	00074	COMMANDS
73245	00341	41	00022	00332	INDEX CARD
73246	00342	41	00005	00134	INDEX GROUP CARDS
73247	00343	11	00350	73254	RESTORE CARD NUMBER
73250	00344	75	31700	01735	RESTORE
73251	00345	11	74100	00100	ES
73252	00346	21	00350	00074	STEP CARD
73253	00347	45	00000	00160	NUMBER
73254	00350	00	00000		CARD NUMBER
73255	00351	05	24220	44463	R EXP-12 TIMES 2 EXP-3
73256	00352	00	00010	00000	
73257	00353	00	00011	00355	
73260	00354	00	00000	00170	
73261	00355	25	71230	64027	R EXP 16 TIMES 2 EXP 35
73262	00356	32	36041	57154	R EXP 8 TIMES 2 EXP 35
73263	00357	35	06512	24172	R EXP 4 TIMES 2 EXP 35
73264	00360	36	41100	00000	R EXP 2 TIMES 2 EXP 35
73265	00361	37	20000	00000	R EXP 1 TIMES 2 EXP 35
73266	00362	00	00000	00044	
73267	00363	00	00000	00*06	
73270	00364	40	00000	00000	
73271	00365	21	00411	00023	PROTOTYPE COMMAND
73272	00366	00	00014	00000	
73273	00367	00	00452	00422	
73274	00370	00	00000	00*11	
73275	00371	00	00000	06554	
73276	00372	54	00006	10003	
73277	00373	54	00006	10000	

FLIP CARD WRITE

73300	00374	00	00461	13177	9999999
73301	00375	00	00461	13200	10000000
73302	00376	00	10001	00010	PERIOD CONSTANT
73303	00377	00	04000	00000	
73304	00400	77	77776	77777	11 ROW
73305	00401	77	77775	77777	12 ROW
73306	00402	00	00000	00436	
73307	00403	00	00000	73777	
73310	00404	00	00000	74000	
73311	00405	00	00000	76000	
73312	00406	00	00000	00112	

FLIP CARD READ

73400		75	31700	73402	STORE
73401		11	00100	74100	ES
73402		75	30254	00100	ROUTINE
73403		11	73404	00100	TO ES
73404	00100	75	10004	00102	CLEAR
73405	00101	11	00040	00002	TEMP
73406	00102	16	01736	00004	FLIP
73407	00103	21	00004	00327	COMMAND
73410	00104	42	00325	00106	
73411	00105	23	00004	00324	TO
73412	00106	16	00004	00107	
73413	00107	21	00003	30000	ACCUMULATOR
73414	00110	16	20000	00004	SET
73415	00111	21	00004	00324	STORAGE
73416	00112	42	00325	00114	
73417	00113	23	00004	00324	ORDER
73420	00114	16	00004	00317	
73421	00115	55	00003	10015	
73422	00116	11	00334	00360	NUMBER WORDS STORAGE
73423	00117	11	00067	00357	NUMBER-1
73424	00120	44	00131	00121	TEST OPTION
73425	00121	11	00074	00002	6 OPTION
73426	00122	23	00357	00041	WITH
73427	00123	23	00360	00074	CARD
73430	00124	23	00316	00073	NUMBER
73431	00125	37	00125	00126	
73432	00126	44	00131	00127	TEST OPTION
73433	00127	11	00041	00002	WITHOUT
73434	00130	37	00125	00123	CARD NUMBER
73435	00131	55	00003	00025	
73436	00132	51	00064	20000	
73437	00133	43	00040	00322	EXIT IF N IS 0

FX 71900-8-II

FLIP CARD READ

73440	00134	36	00074	00004	N-1
73441	00135	17	00000	00353	READ AND PICK READ CARD
73442	00136	11	00357	00361	SET INDEX
73443	00137	11	00330	00006	9 TO LINE DIGIT
73444	00140	16	00337	00313	SET TEMPORARY STORAGE
73445	00141	75	10016	00143	CLEAR
73446	00142	11	00040	00021	MATRIX
73447	00143	76	00000	00363	READ
73450	00144	76	10000	10000	ROW
73451	00145	76	10000	00362	
73452	00146	54	00363	00034	
73453	00147	37	00147	00150	
73454	00150	11	00331	00160	SET INITIAL STORAGE
73455	00151	11	00333	00017	SET INDEX 5
73456	00152	31	00066	00001	2 EXP 35 TO A
73457	00153	32	00040	00006	SHIFT 6
73460	00154	44	00155	00156	BIT ZERO TEST
73461	00155	32	00006	00000	ADD LINE DIGIT
73462	00156	46	00157	00153	DONE 6 DIGITS TEST
73463	00157	11	20000	20000	CLEAR A LEFT
73464	00160	30	00000	0000	STORE MATRIX WORD
73465	00161	21	00160	00075	ADVANCE
73466	00162	41	00017	00152	6 TIMES
73467	00163	37	00163	00164	
73470	00164	11	00362	10000	
73471	00165	37	00163	00151	
73472	00166	11	00363	10000	
73473	00167	11	00074	00017	
73474	00170	37	00163	00152	
73475	00171	37	00171	00172	
73476	00172	23	00006	00074	REDUCE LINE DIGIT
73477	00173	46	00174	00143	TEST FOR 11 ROW

FLIP CARD READ

73500	00174	11	00336	00006	20 TO LINE DIGIT
73501	00175	37	00171	00143	11 ROW
73502	00176	11	00060	00006	10 TO LINE DIGIT
73503	00177	37	00147	00143	START 12 ROW
73504	00200	31	10000	00001	TEST FOR
73505	00201	44	00202	00203	FLAG
73506	00202	13	00074	00004	OVERRIDE INDEX
73507	00203	55	20000	00043	
73510	00204	37	00171	00150	FINISH 12 ROW
73511	00205	11	00004	20000	TEST FOR
73512	00206	42	00074	00210	LAST CARD
73513	00207	17	00000	00353	
73514	00210	11	00332	00355	SET FOR WORD CHANGE
73515	00211	15	00331	00231	SET FOR EXTRACTION
73516	00212	11	00002	20000	TEST
73517	00213	47	00214	00224	OPTION
73520	00214	37	00244	00224	CARD NUMBER COMPUTED
73521	00215	11	00002	20000	TEST
73522	00216	43	00041	00221	OPTION
73523	00217	11	00017	00007	IDENT NUMBER USED
73524	00220	21	00313	00074	ADVANCE
73525	00221	31	00002	00000	TEST
73526	00222	47	00223	00224	OPTION
73527	00223	37	00232	00226	PERIOD
73530	00224	11	00332	00354	INDEX 6
73531	00225	11	00040	00017	CLEAR
73532	00226	41	00355	00231	TEST TO
73533	00227	21	00231	00073	CHANGE
73534	00230	11	00333	00355	MATRIX
73535	00231	55	30000	00006	WORD
73536	00232	37	00232	00233	POSITION DIGIT
73537	00233	31	00017	00002	X10

FLIP CARD READ

```
73540  00234  32 00017 00001      ADD
73541  00235  52 00064 00017        DIGIT
73542  00236  41 00354 00226
73543  00237  37 00237 00240
73544  00240  37 00232 00226     TEST
73545  00241  51 00064 20000        FOR
73546  00242  47 00243 00244          SIGN
73547  00243  13 00017 00017
73550  00244  37 00244 00245
73551  00245  11 00017 00020     STORE N
73552  00246  11 00040 00017
73553  00247  11 00002 20000     TEST
73554  00250  47 00251 00252      OPTION
73555  00251  37 00237 00226     EXP 6 OPTION
73556  00252  37 00244 00226     BOTH OPTIONS
73557  00253  71 00020 00340     CONVERT
73560  00254  11 00040 00356        TO
73561  00255  43 00040 00313        POSITIVE
73562  00256  54 20000 00060          DECIMAL
73563  00257  11 20000 00020             EXPONENT
73564  00260  21 00017 00341
73565  00261  73 00043 00017     CHANGE
73566  00262  31 20000 00017     TO BASE
73567  00263  35 00342 00301        10 EXP 3
73570  00264  71 00060 00017     CHANGE
73571  00265  36 00343 00354
73572  00266  16 00344 00273        TO
73573  00267  11 00044 00356
73574  00270  55 00017 00036
73575  00271  55 00017 00001            BASE
73576  00272  44 00273 00276
73577  00273  71 00020 30000
```

PX 71900-8-11

FLIP CARD READ

73600	00274	54	20000	00045	
73601	00275	11	20000	00020	2 EXP 10
73602	00276	21	00273	00074	
73603	00277	41	00356	00271	
73604	00300	11	00074	20000	
73605	00301	75	00000	00303	
73606	00302	71	20000	00060	
73607	00303	71	20000	00020	
73610	00304	54	20000	00010	
73611	00305	74	20000	00356	NORMALIZE
73612	00306	11	20000	00020	
73613	00307	11	00352	10000	
73614	00310	21	00356	00354	
73615	00311	53	00020	00356	PACK
73616	00312	35	00005	00005	SUM
73617	00313	11	00356	30000	TEMPORARY STORAGE
73620	00314	21	00313	00074	ADVANCE
73621	00315	41	00361	00221	END OF CARD
73622	00316	75	30010	00320	FINAL
73623	00317	11	00007	30000	STORAGE
73624	00320	21	00317	00360	ADVANCE
73625	00321	41	00004	00136	END TEST
73626	00322	75	31700	01735	RESTORE
73627	00323	11	74100	00100	ES
73630	00324	00	00000	74000	
73631	00325	00	00000	76000	
73632	00326	00	00000	00777	
73633	00327	00	00000	73777	
73634	00330	00	00000	00 11	
73635	00331	35	00021	00021	
73636	00332	00	00000	00#06	
73637	00333	00	00000	00 05	

PX 71900-8-11

FLIP CARD READ

73640	00334	00	00000	00 10	
73641	00335	13	00020	00020	
73642	00336	00	00000	00024	
73643	00337	00	00000	00 07	
73644	00340	22	21147	04413	R EXP-13X10 EXP-7X2 EXP 57
73645	00341	00	00000	00*47	
73646	00342	75	00000	00303	
73647	00343	00	00000	00210	
73650	00344	00	00000	00345	
73651	00345	25	71230	64050	R EXP 16 X 2 EXP 35
73652	00346	32	36041	57166	R EXP 8 X 2 EXP 35
73653	00347	35	06512	24200	R EXP 4X 2 EXP 35
73654	00350	36	41100	00000	R EXP 2 X 2EXP 35
73655	00351	37	20000	00000	R EXP 1 X 2 EXP 35
73656	00352	77	77777	77400	EXTRACTION
73657	00353	00	00000	00105	

CARD INSTRUCTION REPEAT

73771	16	75736	00002	RESTART
73772	23	00002	00074	
73773	16	20000	73776	WITH
73774	75	31700	73776	LAST
73775	11	74100	00100	FLIP
73776	56	00000	30000	COMMAND

PX 719CO-8-11

CV-11

ALARM, OCTAL AND FLEXPRINT SUBROUTINE *

76000	76000	45	00000	76000
76001	76001	16	76026	76022
76002	76002	11	20000	00000
76003	76003	61	00000	76047
76004	76004	55	76012	00005
76005	76005	55	76012	00011
76006	76006	55	76012	00012
76007	76007	34	20000	00003
76010	76010	32	76037	00000
76011	76011	11	20000	76012
76012	76012	00	01000	10001
76013	76013	44	76014	76007
76014	76014	11	10000	76012
76015	76015	61	00000	76021
76016	76016	37	76016	76017
76017	76017	31	00000	00044
76020	76020	11	76000	00000
76021	76021	[illegible]	76016	76004
76022	76022	37	76022	76023
76023	76023	31	76000	00017
76024	76024	15	20000	76025
76025	76025	16	76000	00000
76026	76026	15	76023	76025
76027	76027	16	76027	76000
76030	76030	31	76042	00047
76031	76031	37	76016	76047
76032	76032	41	00000	76033
76033	76033	31	20000	00071
76034	76034	37	76016	76005
76035	76035	56	00000	00000
76036	76036	00	00000	00
76037	76037	61	00000	76037

*See writeup in ZM 491

PX-71900-8-11

ALARMY OCTAL AND FL

```
76040   76040   00 00000 00 52
76041   76041   00 00000 00*74
76042   76042   43 01130 12070
76043   76043   00 00000 00 64
76044   76044   00 00000 00 62
76045   76045   00 00000 00*66
76046   76046   00 00000 00*72
76047   76047   61 00000 20045
76050   76050   34 20000 00006
76051   76051   47 76047 76015
```

FLIP ALARM ROUTINE

```
76052   76052   61 00000 76047     PRINT
76053   76053   31 76042 00047       OUT
76054   76054   37 76016 76047         ALARM
76055   76055   31 01736 00017     SET
76056   76056   34 00073 00000       PICKUP
76057   76057   15 20000 76062         COMMAND
76060   76060   31 20000 00052     PRINT
76061   76061   37 76016 76005       ADDRESS
76062   76062   31 30000 00044     PRINT
76063   76063   37 76016 76004       COMMAND
76064   76064   31 01774 00044     PRINT
76065   76065   37 76016 76004       R
76066   76066   15 00013 76070     SET PICKUP
76067   76067   16 00013 76071       COMMANDS
76070   76070   31 30000 00044     PICK UP X
76071   76071   27 20000 30000     PICKUP Y
76072   76072   37 76022 76002     PRINT X AND Y
76073   76073   31 76104 00044     PRINT
76074   76074   37 76016 76047       OUT
```

PX-71900-8-11

CV-11

FLIP ALARM ROUTINE

76075	76075	31	01776	00071	FIRST
76076	76076	37	76016	76005	INDEX
76077	76077	31	76105	00044	PRINT
76100	76100	37	76016	76047	OUT
76101	76101	31	01777	00071	SECOND
76102	76102	37	76016	76005	INDEX
76103	76103	56	00000	01735	HALT
76104	76104	1406	2227	0452	CONSTANTS
76105	76105	1406	2227	0474	

REFERENCES TO ALARM ROUTINE

72225	72225	46	76052	01336
72272	72272	45	00000	76052
72310	72310	46	01421	76052
76406	76406	46	01607	76052
76507	76507	45	00000	76052
76554	76554	46	76052	01002
77215	77215	45	00000	76052
77366	77366	75	20013	76052
77456	77456	46	01003	76052
76415	76415	30	[illegible]	[illegible]
71600	71600	37	[illegible]	71601
76645	76645	75	[illegible]	01774

62 Arc Tangent Subroutine

76300	01000	11 00040 00012	$0 \rightarrow \theta^1$
76301	01001	23 00010 00074	Scx-1 → Scx
76302	01002	46 01007 01003	0;Scx
76303	01003	13 00010 00010	-Scx → Scx
76304	01004	31 00066 00042	$\frac{1}{4x} \rightarrow x$
76305	01005	73 00005 00005	
76306	01006	27 00012 00062	reverse θ' ($\pi/2$ (34) or zero)
76307	01007	21 00010 01043	Scx + 27 → Scx
76310	01010	46 01011 01012	0;Scx
76311	01011	11 00040 20000	0 → Scx
76312	01012	35 01054 01013	$x.2^{Scx-27} \rightarrow x^1$ (34)
76313	01013	[54 00005 50055]	
76314	01014	71 00005 10000	$(x^1)^2$ (34)
76315	01015	54 20000 00046	
76316	01016	11 20000 00006	
76317	01017	11 00040 00010	0 → P
76320	01020	15 01033 01023	15 → i
76321	01021	71 00010 00006	$P\cdot(x^1)^2 + C_i \rightarrow P$ (35)
76322	01022	54 20000 00046	
76323	01023	35 [01044] 00010	
76324	01024	21 01023 00073	i-2 → i
76325	01025	42 01034 01021	done?
76326	01026	71 00005 00010	$P\cdot x^1 \rightarrow x$ (34)
76327	01027	54 20000 00045	
76330	01030	11 20000 00005	
76331	01031	11 00012 20000	$0; \theta^1$
76332	01032	47 01035 01040	

76333	01033	00 01044 00000	} constants
76334	01034	35 01054 00000	
76335	01035	44 01036 01037	$0 : x^1$
76336	01036	13 20000 20000	$-\theta^1 \rightarrow \theta^1$
76337	01037	36 00005 00005	$\theta^1 - \pi/2 \rightarrow \theta^1$
76340	01040	11 00074 00010	$1 \longrightarrow$ scx
76341	01041	16 00013 01734	set order
76342	01042	45 00000 01721	exit
76343	01043	00 00000 00033	27.
76344	01044	77 67545 00613	c_{15}
76345	01045	00 54613 12165	c_{13}
76346	01046	76 15376 17035	c_{11}
76347	01047	03 05357 57500	c_9
76350	01050	73 43116 35123	c_7
76351	01051	06 30402 45553	c_5
76352	01052	65 25317 10166	c_3
76353	01053	37 77777 23166	c_1
76354	01054	54 00005 00055	prototype order
76721	76721	55 76300 03600	loader parameter

63 ARC COTANGENT SUBROUTINE

76356	01000	11 00062 00012	$\pi/2$ (34) $\rightarrow \theta^1$
76357	01001	45 00000 01003	jump
76723	76723	02 76356 00200	loader parameter

PX 71900-8-11

EXPONENTIAL

76360	01000	23	20000	00067	SCX-7 TO ACC
76361	01001	46	01003	01025	OUT OF RANGE ?
76362	01002	23	00005	00005	CLEAR X AND ACC
76363	01003	42	01030	01002	TOO SMALL ?
76364	01004	35	01032	01005	SET COMMAND
76365	01005	54	00005	00111	X SCALED 2^{28}
76366	01006	71	20000	01031	X
76367	01007	54	20000	00045	TIMES
76370	01010	11	20000	20000	LN 2
76371	01011	73	01027	00010	INTEGER PART TO 00010
76372	01012	31	20000	00042	FRACTIONAL PART DIVIDED
76373	01013	73	01031	00011	BY LN 2 TO 00011
76374	01014	21	00010	00067	STORE EXPONENT OF RESULT
76375	01015	75	10003	01022	SET T, N, P
76376	01016	11	01027	00005	TO UNITY
76377	01017	73	00006	00007	DIVIDE BY N, STORE IN TERM, T
76400	01020	21	00005	00007	ADD TO TO POLONOMIAL, P
76401	01021	21	00006	01027	STEP N
76402	01022	71	00011	00007	X TIMES TERM, T
76403	01023	47	01017	01024	TERM ZERO YET ?
76404	01024	45	00000	01721	EXIT TO NORMALIZE
76405	01025	11	00005	20000	NUMBER NEGATIVE
76406	01026	46	01607	76045	OR POSITIVE ?
76407	01027	00	20000	00000	UNITY SCALED 2
76410	01030	77	77777	77734	-35
76411	01031	27	05243	54513	LN 2 SCALED 2
76412	01032	54	00005	00111	PROTO TYPE COMMAND

PX 71900-8-11

INPUT SUM CHECK

```
76414  76414  45 00000 76575     EMPTY
76415  76415  53 16045 07410     SUM ADJUSTER
76416  76416  11 76414 00000     SET F1
76417  76417  23 10000 10000     CLEAR A AND Q
76420  76420  75 30040 76422     LOAD PERMANENT
76421  76421  11 77716 00040       CONSTANTS TO ES
76422  76422  75 12000 76424     CLEAR TEMPORARY
76423  76423  11 10000 74000       STORAGES
76424  76424  75 27777 76426     SUM
76425  76425  32 70001 00000       FLIP
76426  76426  55 20000 00000     SUM TO Q AND A RIGHT
76427  76427  47 76440 76430     TEST SUM
76430  76430  31 76453 00052     PRINT
76431  76431  37 76452 76450
76432  76432  31 76454 00052       FLIP
76433  76433  37 76452 76450
76434  76434  31 76455 00052
76435  76435  37 76452 76450        OK
76436  76436  56 00000 70400     HALT
76437  76437  56 00000 40000     OBSOLETE
76440  76440  31 76456 00052     PRINT
76441  76441  37 76452 76450
76442  76442  31 76457 00052        NO
76443  76443  37 76452 76450        GO
76444  76444  56 10000 70103     HALT
76445  76445  00 00000 76436     CONSTANT
76446  76446  00 00000 00-0
76447  76447  00 00000 00*0
76450  76450  61 00000 20000     PRINT
76451  76451  34 20000 00006      SUB
76452  76452  47 76450 76436       ROUTINE
76453  76453  45 47261 11415     FLIP
```

PX 71900-8-11

INPUT SUM CHECK

76454	76454	04	03365	74500	OK
76455	76455	00	00000	00*0	EMPTY
76456	76456	45	47020	60304	NO
76457	76457	13	03025	70000	GO

PX 71900-8-11

54 LOGARITHM SUBROUTINE

76462	01000	36	00074	00010	p-1 → p
76463	01001	13	00005	20000	q:0
76464	01002	42	00066	01025	
76465	01003	36	00066	00006	2q-1 → x (35)
76466	01004	54	00006	00001	
76467	01005	11	01022	01007	8 → i
76470	01006	11	00040	00005	0 → L
76471	01007	21	00005	30000	L + a_i → L
76472	01010	71	20000	00006	x · L → L (35)
76473	01011	54	20000	00045	
76474	01012	11	20000	00005	
76475	01013	21	01007	00074	i-1 → i
76476	01014	42	01023	01007	0:i
76477	01015	54	00005	00101	L (28)
76500	01016	71	00010	01026	p.ln 2 → R (28)
76501	01017	35	00005	00005	L+p.ln 2 → q^1
76502	01020	11	00067	00010	7 → p^1
76503	01021	45	00000	01024	exit
76504	01022	21	00005	01027	prototypes
76505	01023	21	00005	01037	
76506	01024	45	00000	01721	normal exit
76507	01025	45	00000	01737	alarm exit
76510	01026	00	13056	20577	ln 2 (28)
76511	01027	77	62620	75765	a_8

PX 71900-8-11

76512	01030	01	11721	41642	a_7
76513	01031	74	74607	70746	a_6
76514	01032	05	27266	02203	a_5
76515	01033	70	22764	23456	a_4
76516	01034	12	47414	37545	a_3
76517	01035	60	00203	77320	a_2
76520	01036	37	77774	20006	a_1
76705	76705	37	76462	02400	loader parameter

PX 71900-8-11

50 SQUARE ROOT SUBROUTINE

76553	01000	11	00005	20000	X → A
76554	01001	46	76045	01002	x:0
76555	01002	42	00073	01020	2^{-155}:x (ie is (x) = 0?)
76556	01003	54	00010	00107	½ sc x ± {0, ½} → scx
76557	01004	43	20000	01010	scx odd?
76560	01005	46	01006	01007	scx:0
76561	01006	35	00074	00010	scx + 1 → scx
76562	01007	54	00005	00107	½ x → x
76563	01010	11	00005	00006	x → N
76564	01011	11	00070	00005	$1 - 2^{-35}$ → xi
76565	01012	31	00006	00042	½N (70) → A
76566	01013	73	00005	00004	½N ÷ Xi → C, t_4 (35)
76567	01014	54	00005	00107	½ Xi → A, t_5 (35)
76570	01015	23	10000	00005	½N ÷ Xi − ½Xi → Q = Δ Xi
76571	01016	21	00005	00004	½N ÷ Xi + ½Xi → Xi + 1
76572	01017	44	01012	01020	Δ Xi:0
76573	01020	45	00000	01721	exit
76675	76675	21	76553	02000	loader parameter

PX 71900-8-11

Loader

76575	01537	11 76655 00000	Preset F_1
76576	01540	75 30061 01615	Transfer loader to ES
76577	01541	11 76600 01542	
76600	01542	75 30062 01544	Transfer Constants and Assembly to ES
76601	01543	11 77716 00040	
76602	01544	11 00122 00003	Requisition parameter P_i
76603	01545	11 00003 00004	$P_i \rightarrow t_4$
76604	01546	31 00004 00017	$\phi \; (P_i) \rightarrow \theta \; (t_5)$
76605	01547	11 20000 00005	
76606	01550	55 00003 00026	$2 \cdot OP \rightarrow \theta \; (t_3)$
76607	01551	11 01621 10000	Mask $\rightarrow$ q
76610	01552	53 00003 01554	Set order: "requisition subroutine parameters"
76611	01553	75 30002 01555	Requisition Subroutine Parameter P_1, P_2
76612	01554	11 01[600] 00006	
76613	01555	15 00006 01576	Set order: "Transfer subroutine to ES"
76614	01556	16 00004 01576	
76615	01557	55 00006 00011	Place parameter for assembly routine
76616	01560	31 00005 00071	
76617	01561	52 01621 00122	
76620	01562	47 01563 01601	Is this the termination flag?
76621	01563	55 00006 00014	Set repeat order: "Transfer subroutine to ES"
76622	01564	11 01577 20000	
76623	01565	52 01621 01575	
76624	01566	16 00007 01571	Set subroutine references in Basic Code
76625	01567	55 00007 00025	
76626	01570	16 10000 01572	
76627	01571	16 00004 [30000]	

PX 71900-8-11

76630	01572	15	00005	[30000]	
76631	01573	21	01544	00073	Step to next subroutine
76632	01574	21	01561	00074	
76633	01575	[77	77777	77777]	Transfer Subroutine to ES
76634	01576	11	[00000	00000]	
76635	01577	75	30000	01544	Prototype order for 01575
76636	01600	00	00000	00000	"Parameter for zero subroutine"
76637	01601	11	01620	01774	Preset exit of Loader
76640	01602	11	01617	00001	Preset F_2
76641	01603	11	01622	00002	obsolete order
76642	01604	37	00101	00100	Modify subroutines
76643	01605	75	10003	01607	Clear FLIP temporaries
76644	01606	11	00040	01775	
76645	01607	75	30275	01774	Transfer in Basic FLIP
76646	01610	11	76755	01477	
76647	01611	75	30222	01776	Obsolete
76650	01612	11	77027	01551	
76651	01613	75	30236	01776	
76652	01614	11	77014	01536	
76653	01615	75	30074	01542	Transfer Subroutine parameters to ES
76654	01616	11	76661	01704	
76655	01617	45	00000	01624	(F_2')
76656	01620	56	00000	00010	Exit order for Loader
76657	01621	00	00177	00000	Extractor
76660	01622	45	00000	76761	obsolete order

PX 71900-8-11

COMMAND CODE PARAMETERS

76661	01704	01 01731 00000		42
76662	01705	00 00003 00003		
76663	01706	72 77352 05000		43
76664	01707	00 77023 00003		
76665	01710	01 01731 00000		44
76666	01711	00 00003 00003		
76667	01712	01 01731 00000		45
76670	01713	00 00003 00003		
76671	01714	01 01731 00000		46
76672	01715	00 00003 77007		
76673	01716	01 01731 00000		47
76674	01717	00 77007 00003		
76675	01720	21 76553 02000		50
76676	01721	00 00003 77026		
76677	01722	24 76360 00500		51
76700	01723	00 00003 00003		
76701	01724	02 77450 00200		52
76702	01725	00 00003 77011		
76703	01726	70 77262 03400		53
76704	01727	00 77011 00003		
76705	01730	37 76462 02400		54
76706	01731	00 00003 77012		
76707	01732	01 01731 00000		55
76710	01733	00 77012 00003		
76711	01734	01 01731 00000		56
76712	01735	00 00003 77013		
76713	01736	01 01731 00000		57
76714	01737	00 77013 00003		
76715	01740	61 77454 04700		

PX 71900-8-11

76716	01741	00	00003	76777	
76717	01742	02	77452	00200	61
76720	01743	00	76777	00003	
76721	01744	55	76300	03600	62
76722	01745	00	00003	77000	
76723	01746	02	76356	00200	63
76724	01747	00	77000	00003	
76725	01750	01	01731	00000	64
76726	01751	00	00003	77001	
76727	01752	01	01731	00000	65
76730	01753	00	77001	00003	
76731	01754	01	01731	00000	66
76732	01755	00	00003	77002	
76733	01756	01	01731	00000	67
76734	01757	00	77002	00003	
76735	01760	31	76360	01200	70
76736	01761	00	00003	77003	
76737	01762	01	01731	00000	71
76740	01763	00	77003	00003	
76741	01764	01	01731	00000	72
76742	01765	00	00003	77004	
76743	01766	01	01731	00000	73
76744	01767	00	77004	00003	
76745	01770	01	01731	00000	74
76746	01771	00	00003	77005	
76747	01772	01	01731	00000	75
76750	01773	00	77005	00003	
76751	01774	01	01731	00000	76
76752	01775	00	00003	77006	
76753	01776	32	76521	03100	77
76754	01777	00	00003	77213	

PX 71900-8-11

BASIC FLIP

```
76755  01477  44 01605 01737   SORT
76756  01500  27 00005 01772     REVERSE SIGN OF X
76757  01501  44 01502 01675     STORE IN Y
76760  01502  16 01712 01734        OR R?
76761  01503  44 01504 01657     DECIDE TO
76762  01504  11 01774 00007       ACCUMULATE
76763  01505  45 00000 01657     JUMP
76764  01506  33 00051 00107     -28 TO A
76765  01507  35 00012 00012     -P TO T12
76766  01510  42 00057 01726     11 : -P
76767  01511  45 00000 01607     JUMP TO ZERO OUT
76770  01512  44 01514 01513     S
76771  01513  44 01516 01515      W
76772  01514  44 01520 01517       I
76773  01515  44 01522 01521        T
76774  01516  44 01524 01523         C
76775  01517  44 01526 01525          H
76776  01520  44 01530 01527           I
76777  01521  44 01737 01737            N
77000  01522  44 01737 01737             G
77001  01523  45 00000 72004     W
77002  01524  45 00000 72004      H
77003  01525  44 01737 01737       I
77004  01526  45 00000 01737        F
77005  01527  45 00000 01526         F
77006  01530  44 01737 01737          L
77007  01531  44 01737 01737           E
77010  01532  44 01535 01534            T
77011  01533  44 01737 01737             R
77012  01534  44 72444 01732              E
77013  01535  44 73400 73000               E
77014  01536  44 01512 01537
```

PX 71900-8-11

BASIC FLIP

```
77015  01537  44 01546 01540
77016  01540  44 01542 01541
77017  01541  44 01545 01543
77020  01542  44 01531 01544
77021  01543  44 01571 01556
77022  01544  44 01574 01563
77023  01545  44 01737 01550
77024  01546  44 01532 01547
77025  01547  44 01533 01477
77026  01550  11 00011 20000    SCY MINUS     (TALLY FOR TRACE)
77027  01551  36 00060 00010     10 TO SC X
77030  01552  54 00005 20044    SHIFT X IN A
77031  01553  45 00000 01722    JUMP
77032  01554  11 00005 20000    IS X
77033  01555  47 01500 01501      ZERO?
77034  01556  21 01776 00074    STEP B1
77035  01557  36 00005 20000    B1-X
77036  01560  42 00074 01602    TEST
77037  01561  11 00040 01776    CLEAR B1
77040  01562  45 00000 01603    EXIT
77041  01563  27 00005 01772    RESTORE SIGN X
77042  01564  21 01777 00074    STEP B2
77043  01565  36 00005 20000    B2-X
77044  01566  42 00074 01602    TEST
77045  01567  11 00040 01777    CLEAR B2
77046  01570  45 00000 01603    EXIT
77047  01571  16 01734 00012    INTERCHANGE
77050  01572  16 01736 01734     Y AND EXIT
77051  01573  16 00012 01736      ADDRESSES
77052  01574  23 20000 00011    SCX-SCY
77053  01575  46 01602 01576    TEST
77054  01576  47 01603 01577    TEST
```

BASIC FLIP

77055	01577	12	00005	00005	|X|
77056	01600	12	00006	20000	|Y|
77057	01601	42	00005	01603	TEST
77060	01602	16	01734	01736	CHANGE EXIT
77061	01603	11	01774	20000	RESTORE R
77062	01604	45	00000	01735	EXIT
77063	01605	23	20000	00011	SCX-SCY
77064	01606	42	01765	01610	TEST SHIFT
77065	01607	23	00006	00006	CLEAR Y
77066	01610	46	01737	01611	LEFT SHIFT ?
77067	01611	13	20000	20000	SCX-SXY
77070	01612	35	01773	01613	SHIFT
77071	01613	[54	00006	00110]	Y
77072	01614	45	00000	01734	EXIT
77073	01615	11	00006	00007	Y TO B
77074	01616	31	01774	00034	SCR TO A
77075	01617	47	01620	01612	ZERO ?
77076	01620	11	20000	00011	SCR TO
77077	01621	54	00011	00054	SCY
77100	01622	11	01774	00006	R TO Y
77101	01623	45	00000	01744	EXIT TO MULTIPLY
77102	01624	16	00000	01736	STORE EXIT
77103	01625	11	20000	01774	STORE R
77104	01626	31	00000	00017	PICKUP
77105	01627	35	01766	01630	COMMAND
77106	01630	11	[30000]	10000	IN Q
77107	01631	75	30005	01633	DIVIDE COMMAND
77110	01632	51	01760	00004	IN PARTS
77111	01633	47	01634	01640	INDEX MODIFICATION REQUIRED ?
77112	01634	71	00005	01777	PRODUCE
77113	01635	31	20000	00001	INDEX
77114	01636	72	00006	01776	MODIFICATIONS

BASIC FLIP

```
77115   01637   55 20000 00031          REQUIRED
77116   01640   32 00007 00003      PRODUCE
77117   01641   11 20000 00013       PROPER
77120   01642   54 00013 10025        ADDRESSES
77121   01643   55 10000 00014          IN
77122   01644   16 10000 01734           CELL
77123   01645   16 10000 00013            00013
77124   01646   55 10000 00017
77125   01647   15 10000 01653      SET Y PICKUP
77126   01650   15 00013 01652      SET X PICKUP
77127   01651   11 00040 00007      CLEAR B
77130   01652   11 [30000] 00005    PICKUP
77131   01653   11 [30000] 00006      OPERANDS
77132   01654   55 00004 00011      DECIDE ON
77133   01655   44 01554 01501         SIGN FOR X
77134   01656   11 01774 00006      R TO Y
77135   01657   31 00006 00034      UNPACK
77136   01660   47 01662 01661       AND
77137   01661   31 00066 00001        TEST
77140   01662   11 20000 00011         Y FOR
77141   01663   54 00011 00054          ZERO
77142   01664   31 00005 00034      UNPACK
77143   01665   47 01667 01666        AND
77144   01666   31 00066 00001         TEST
77145   01667   11 20000 00010          X FOR
77146   01670   54 00010 00054           ZERO
77147   01671   55 00004 00041      IS THIS AN
77150   01672   44 01536 01676        ARITHMETIC COMMAND?
77151   01673   11 00005 20000      DIVIDING BY
77152   01674   47 01740 01737        ZERO?
77153   01675   44 01656 01657      IS R AN OPERAND?
77154   01676   44 01677 01700      DECIDE
```

BASIC FLIP

```
77155   01677   44 01673 01744        WHAT OPERATION
77156   01700   44 01615 01701          TO PERFORM
77157   01701   23 20000 00011       SCX-SCY TO T11
77160   01702   46 01703 01710       WHICH TO BE SHIFTED ?
77161   01703   11 00005 10000       INTERCHANGE
77162   01704   11 00006 00005        THE
77163   01705   11 10000 00006         ARGUMENTS
77164   01706   11 00011 00010       SCY TO SCX
77165   01707   13 20000 20000       SHIFT TOO
77166   01710   42 00045 01712          LARGE ?
77167   01711   11 00045 20000       SET MAX SHIFT
77170   01712   13 20000 20000       SHIFT
77171   01713   35 01767 01714         Y CORRECT
77172   01714   54 00006 00107          AMOUNT
77173   01715   54 00005 00107       SHIFT X
77174   01716   35 00006 00005       ADD
77175   01717   21 00010 00074       CORRECT SCX
77176   01720   45 00000 01750       JUMP
77177   01721   54 00005 00032       SHIFT X IN A
77200   01722   11 00040 00012       CLEAR T 12
77201   01723   74 20000 00012       X TIMES 2 TO THE P
77202   01724   11 20000 00005          I8-P TO T12
77203   01725   47 01506 01734       IS ANSWER ZERO ?
77204   01726   21 00010 00012       SCX -P TO SCX
77205   01727   42 01771 01607       CHECK
77206   01730   42 01770 01732         SIZE OF
77207   01731   45 00000 01737          SCX
77210   01732   11 01772 10000       PACK
77211   01733   53 00005 00010        ANSWER
77212   01734   11 20000 [00000]     STORE
77213   01735   45 10000 [01736]     TRACE SWITCH
77214   01736   45 00000 [00000]     EXIT
```

PX 71900-8-11

BASIC FLIP

77215	01737	45	00000	76045	ALARM EXIT
77216	01740	11	00041	20000	RECIPROCATE
77217	01741	36	00010	00010	X
77220	01742	31	00066	00042	FOR
77221	01743	73	00005	00005	DIVISION
77222	01744	71	00006	00005	MULTIPLY
77223	01745	54	20000	00045	X TIMES Y
77224	01746	11	20000	00005	AND STORE
77225	01747	21	00010	00011	AS X
77226	01750	54	00007	10034	TEST WHETHER
77227	01751	47	01752	01721	TO ACCUMULATE
77230	01752	11	00007	00006	SHIFT
77231	01753	54	10000	00054	B TO
77232	01754	11	20000	00011	Y STORAGE
77233	01755	11	00010	20000	SCX TO A
77234	01756	11	00040	00007	CLEAR B
77235	01757	45	00000	01701	JUMP
77236	01760	00	77000	00000	EX
77237	01761	00	00200	02000	TR
77240	01762	00	00400	04000	AC
77241	01763	00	00177	71777	TO
77242	01764	00	00600	06000	RS
77243	01765	00	00000	00043	35
77244	01766	10	77777	10000	PROTOTYPE
77245	01767	54	00006	00107	COMMANDS
77246	01770	00	00000	00200	128
77247	01771	77	77777	77600	EXTRACTOR
77250	01772	77	77777	77400	
77251	01773	54	00006	00110	PROTOTYPE COMMAND

CV-11 53 PUNCH SUBROUTINE

77262	01000	11	01067	00002	prototype → t_2
77263	01001	11	00005	00035	x → t_{35}
77264	01002	11	00040	00036	0 → decimal exponent (d.e.)
77265	01003	75	30016	01005	load temporary orders
77266	01004	11	01051	00015	
77267	01005	54	00005	20015	test X
77270	01006	43	20000	01010	
77271	01007	37	00027	00015	enter temporaries
77272	01010	75	10020	01012	put "space" to all digits
77273	01011	13	00074	00015	
77274	01012	71	00035	01041	$X \cdot 10^7$
77275	01013	46	01014	01015	adjust sign digit
77276	01014	11	01037	00015	
77277	01015	54	20000	00051	$X \cdot 10^7$ as integer
77300	01016	12	20000	20000	$\lvert x \rvert$
77301	01017	35	01050	20000	Round
77302	01020	73	01041	00016	1st digit
77303	01021	13	01040	00017	decimal point
77304	01022	75	30006	01024	Remaining digits
77305	01023	73	01042	00020	
77306	01024	11	01036	00037	set decimal point
77307	01025	11	00036	20000	adjust sign digit of d.e.
77310	01026	46	01027	01030	
77311	01027	11	01037	00027	
77312	01030	12	20000	20000	$\lvert d.e. \rvert$
77313	01031	73	01047	00030	1st digit d.e.
77314	01032	11	20000	00031	2nd digit d.e.
77315	01033	11	01063	00034	jump order
77316	01034	75	20017	00015	convert to orders
77317	01035	21	00015	00002	
77320	01036	00	00000	00042	decimal point

77321	01037	00 00000 00014	12.
77322	01040	00 00000 00006	6.
77323	01041	00 00461 13200	10^7
77324	01042	00 00036 41100	10^6
77325	01043	00 00003 03240	10^5
77326	01044	00 00000 23420	10^4
77327	01045	00 00000 01750	10^3
77330	01046	00 00000 00144	10^2
77331	01047	00 00000 00012	10
77332	01050	00 00000 00005	5
77333	00015	16 01736 00015	save exit address
77334	00016	21 00036 00074	d.e + 1 → d.e.
77335	00017	14 30003 20035	N ÷ 10 → N
77336	00020	14 45003 10016	10:N
77337	00021	14 45003 00025	1:N
77340	00022	23 00036 00074	d.e.-1 → d.e.
77341	00023	14 20003 20035	10N → N
77342	00024	45 00000 00021	jump
77343	00025	14 51003 20035	FLOATING ⟶ FIXED
77344	00026	16 00015 01736	restore exit
77345	00027	45 00000 01735	exit
77346	00030	37 77777 30000	1 (adjusted for rounding)
77347	00031	23 77777 75404	10 (adjusted for rounding)
77350	00032	24 00000 00004	10
77351	01067	63 00000 00045	prototype order
76703	76703	70 77262 03400	loader parameter

52 PRINT SUBROUTINE

77450	01000	11 00042 00002	set prototype
77451	01001	45 00000 01003	jump
76701	76701	02 77450 00200	loader parameter

43 INPUT CONVERSION SUBROUTINE

CV-11

77352	01000	16 01042 01025	set switch Ia
77353	01001	15 00013 01003	set orders
77354	01002	16 00013 01064	
77355	01003	11 30000 00004	$(x) \to x$
77356	01004	11 01051 00012	$6 \to$ tally 2
77357	01005	11 01051 00003	$6 \to$ tally 1
77360	01006	11 01046 01024	set order 1 $-h \to N$
77361	01007	37 01041 01032	1st FLEX digit (sign)
77362	01010	43 00061 01012	Test for minus sign
77363	01011	23 01024 01050	charge order: $+h \to \left\{ \frac{N}{d.c.} \right\}$
77364	01012	23 00005 00005	$0 \to A, h$
77365	01013	37 01041 01032	one digit $\to$ A
77366	01014	75 20013 76045	binary digit $\to d'$
77367	01015	43 00045 01016	
77370	01016	51 00064 00006	
77371	01017	11 00060 00007	
77372	01020	23 00007 00006	
77373	01021	71 00060 00005	$10\,h + d' \to h$
77374	01022	35 00007 00005	
77375	01023	41 00003 01013	Index tally 2
77376	01024	[13 00005 30000]	$\pm h \to \left\{ \frac{N}{d.c.} \right\}$
77377	01025	37 01025 30000	I_b
77400	01026	54 00032 00037	$N \cdot 2^{31} \to A$
77401	01027	73 01070 00032	$N \cdot 2^{31} \div 10^7 \to N$
77402	01030	75 30016 00014	enter temporary orders
77403	01031	11 01052 00014	
77404	01032	41 00012 01037	Index tally 2 — Digit Subroutine
77405	01033	11 01003 20000	$(x+1) \to x$
77406	01034	35 00073 01035	
77407	01035	11 30000 00004	

PX 71900-8-11

CV-11

77410	01036	11	00067	00012	7 → tally 2
77411	01037	55	00004	00006	shift one flex digit,
77412	01040	51	00064	20000	x tract to A
77413	01041	45	00000	30000	exit
77414	01042	00	00000	01043	constant
77415	01043	11	00074	00003	1 → tally 1
77416	01044	11	01047	01024	set order - h → d.e.
77417	01045	45	00000	01007	jump
77420	01046	13	00005	00032	constants
77421	01047	13	00005	00033	constants
77422	01050	02	00000	00000	constants
77423	01051	00	00000	00006	constants
77424	00014	16	01736	00002	store exit address
77425	00015	14	42003	20031	FIXED → FLOATING
77426	00016	11	20000	00032	FIXED → FLOATING
77427	00017	14	30003	10032	N ÷ 10 → N
77430	00020	21	00033	00074	d.e. + 1 → d.e.
77431	00021	46	00017	00022	0:d.e.
77432	00022	43	00040	00026	0:d.e.
77433	00023	14	20003	10032	10N → N
77434	00024	23	00033	00074	d.e. -1 → d.e.
77435	00025	45	00000	00022	jump
77436	00026	11	00032	00000	store
77437	00027	16	00002	01736	exit
77440	00030	45	00000	01735	exit
77441	00031	24	00000	00004	10
77442	01070	00	00461	13200	10^7
76663	76663	72	77352	05000	loader parameters

CV-11

61 COSINE SUBROUTINE

77452	01000	11	01054	00004	$\pi/2 \rightarrow t_4$
77453	01001	45	00000	01003	jump
76717	76717	02	77452	00200	loader parameter

60 SINE SUBROUTINE

77454	01000	11	00040	00004	$0 \rightarrow t_4$
77455	01001	23	00010	00056	scx-27 → scx
77456	01002	46	01003	76045	ALARM?
77457	01003	35	00053	20000	scx-27 + 54 → A
77460	01004	46	01006	01007	Zero Result?
77461	01005	00	00000	00036	30.
77462	01006	11	00040	00005	0 → x
77463	01007	36	01005	10000	scx-27 +54-30 → Q,A
77464	01010	35	01050	01011	$x \cdot 2^{scx} \rightarrow x$
77465	01011	[11	00010	10000]	
77466	01012	44	01013	01014	left shift?
77467	01013	11	20000	20000	0 → L
77470	01014	73	01060	10000	$x - n \cdot 2\pi \rightarrow A = Y$ (32)
77471	01015	35	00004	20000	$Y + \{0, \pi/2\} \rightarrow Y$
77472	01016	11	00066	00004	$+ \text{sign} \rightarrow t_4$
77473	01017	42	01052	01023	$\pi/2 : Y$
77474	01020	55	00004	00001	change sign t_4
77475	01021	36	01051	20000	$Y - \pi \rightarrow Y$
77476	01022	45	01057	01017	jump
77477	01023	54	20000	00043	$Y/2\pi \rightarrow x$ (35)
77500	01024	73	01052	00005	
77501	01025	71	00005	10000	$x^2 \rightarrow t_6$ (35)
77502	01026	54	20000	00045	
77503	01027	11	20000	00006	
77504	01030	11	00040	00007	0 → P

PX 71900-8-11

77505	01031	15	01022	01035	9 → i
77506	01032	11	00044	00010	4 → tally
77507	01033	71	00007	00006	}
77510	01034	54	20000	00045	} x.P + C1 → P (34)
77511	01035	35	01060	00007	}
77512	01036	23	01035	00073	step i
77513	01037	41	00010	01033	done?
77514	01040	71	00007	00005	}
77515	01041	54	20000	00045	} x.P → x (34)
77516	01042	11	20000	00005	}
77517	01043	11	00004	10000	}
77520	01044	44	01045	01046	} check sign t_4
77521	01045	13	00005	00005	-x → x
77522	01046	11	00074	00010	1 → scx
77523	01047	45	00000	01721	exit
77524	01050	54	00005	24110	prototype
77525	01051	14	44176	65200	π (32)
77526	01052	06	22077	32504	π/2 (32)
77527	01053	31	10375	52202	c_1
77530	01054	65	52420	76452	c_3
77531	01055	01	21464	25731	c_5
77532	01056	77	73155	46346	c_7
77533	01057	00	00117	32757	c_9
77534	01060	31	10375	52421	2π (32)
76715	76715	61	77454	04700	loader parameter

PX 71900-8-11

By: C. J. Swift

FLIP CHARACTRON TRACE

ACTIVATING SUBROUTINE

77570	77570	45	26111	41501	FLIPT
77571	77571	45	26110	31501	FLOPT
77572	77572	16	77577	01735	SET ES JUMP TO NOT TRACE
77573	77573	[61	00000	20000	PRINT
77574	77574	34	20000	00006	SUB
77575	77575	47	77573]	77576	ROUTINE
77576	77576	11	01735	77213	PLACE JUMP IN MD
77577	77577	56	00000	01736	STOP
77600	77600	11	77214	01735	PLACE JUMP IN ES
77601	77601	37	71174	71140	PROCESS TWO
77602	77602	37	71174	71140	PAGES OF TRACE
77603	77603	31	77571	00052	FLOPT TO ACC
77604	77604	37	77604	[77606]	TWO WAY
77605	77605	37	77604	77572	SWITCH
77606	77606	31	77570	00052	FLIPT TO ACC
77607	77607	37	01735	77573	SET ES JUMP TO TRACE

FLIP CHARACTRON TRACE

CONCURRENT SUBROUTINE

77610	01551	75	30017	01553	LOAD THIS
77611	01552	11	77612	01553	INTO ES
77612	01553	11	20000	01553	STORE ACC
77613	01554	21	01562	01550	SET COMMAND WITH TALLY
77614	01555	31	01736	00017	STORE TRACED COMMAND
77615	01556	36	00073	01551	ADDRESS
77616	01557	15	20000	01560	STORE TRACED
77617	01560	11	30000	01552	COMMAND
77620	01561	75	30003	01563	STORE INFORMATION
77621	01562	00	01540	57637	IN BIN
77622	01563	21	01550	00043	STEP TALLY (UNUSED BITS IN BASIC FLIP)
77623	01564	42	01571	01566	BIN FULL?
77624	01565	37	71174	71140	PROCESS INFORMATION
77625	01566	11	01553	20000	RESTORE ACC
77626	01567	75	30021	01736	RESTORE BASIC
77627	01570	11	77027	01551	FLIP IN ES
77630	01571	11	00011	20130	PROTOTYPE FOR TEST

PERMANENT CONSTANTS

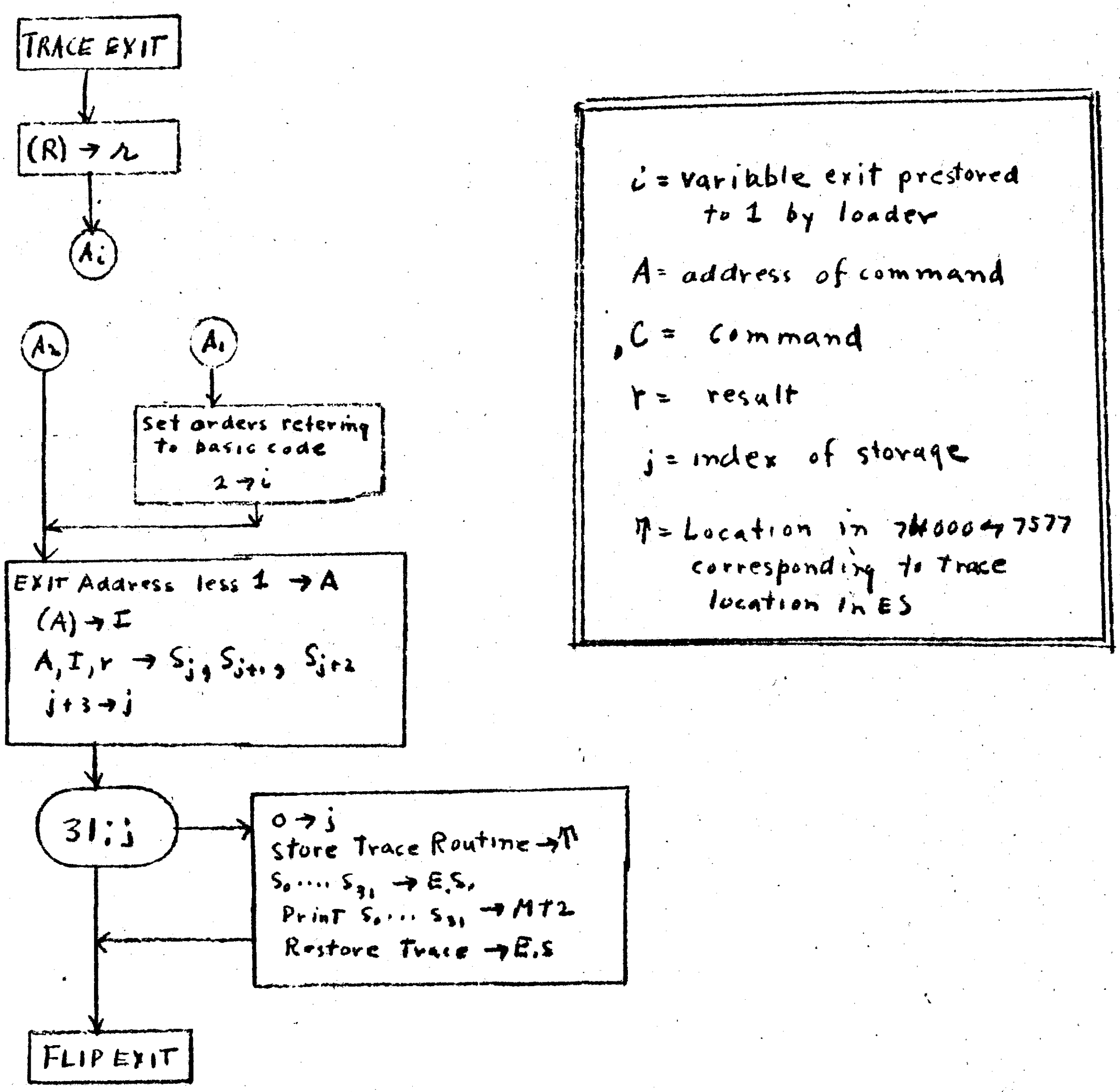
77716	00040	00	00000	0 *
77717	00041	00	00000	00002
77720	00042	61	00000	00045
77721	00043	00	00000	00003
77722	00044	00	00000	00004
77723	00045	00	00000	00037
77724	00046	00	00000	00052
77725	00047	00	00000	00074
77726	00050	00	00000	00070
77727	00051	00	00000	00064
77730	00052	00	00000	00062
77731	00053	00	00000	00066
77732	00054	00	00000	00072
77733	00055	00	00000	00060
77734	00056	00	00000	00033
77735	00057	00	00000	00013
77736	00060	00	00000	00*12
77737	00061	00	00000	00056
77740	00062	31	10375	52421
77741	00063	31	46314	63146
77742	00064	00	00000	00077
77743	00065	21	67643	24177
77744	00066	20	00000	00000
77745	00067	00	00000	00007
77746	00070	37	77777	77777
77747	00071	00	77777	00000
77750	00072	00	00000	77777
77751	00073	00	00001	00000
77752	00074	00	00000	00001
77753	00075	00	00001	00001
77754	00076	00	07777	07777
77755	00077	00	00000	00110

ASSEMBLY ROUTINE

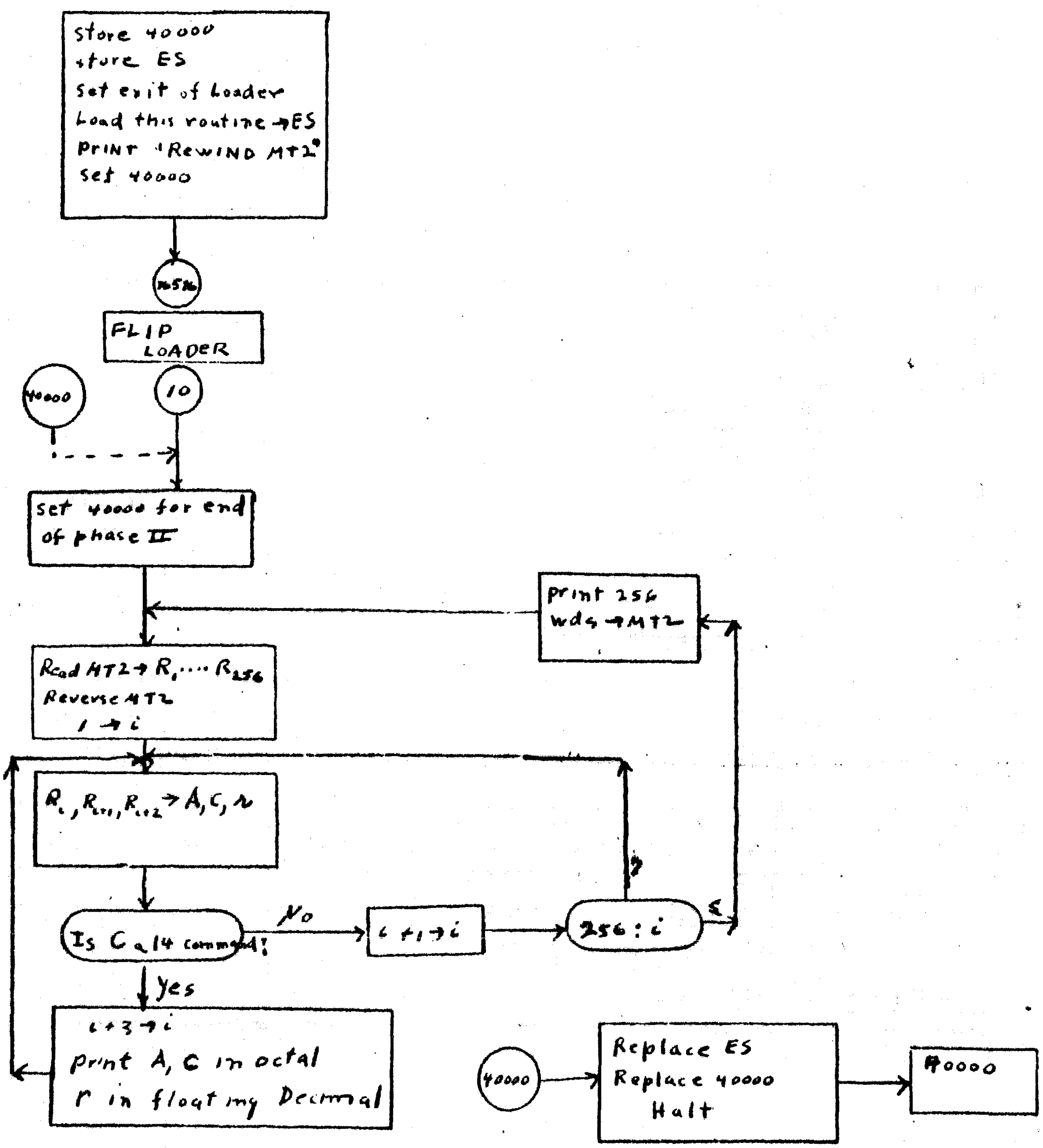
77756	00100	11	00122	20000	PARAMETER TO A
77757	00101	47	00102	00000	DONE
77760	00102	73	00073	00005	N TO TEMP
77761	00103	16	20000	00112	SET PICKUP
77762	00104	36	00121	00006	COMPUTE ADJUSTER
77763	00105	41	00005	00110	DONE ?
77764	00106	21	00100	00073	STEP PARAMETER PICKUP
77765	00107	45	00000	00100	RETURN
77766	00110	16	00112	00116	SET STORE
77767	00111	11	00040	00007	PICKUP
77770	00112	21	00007	00000	NEXT CELL
77771	00113	55	20000	00033	SHIFT IN Q
77772	00114	51	00075	10000	ADJUST AND
77773	00115	71	10000	00006	STORE
77774	00116	35	00007	00000	COMMAND
77775	00117	21	00112	00074	STEP PICKUP
77776	00120	45	00000	00105	
77777	00121	00	00000	01000	CONSTANT

PX 71900-8-11

V Flow Chart, Phase I



VI Flow Chart, Phase II



FORM 1812

PX 71900-8-11

77 PHASE I of TRACE

76521	01000	37 01000 [01023]	2 → i
76522	01001	11 20000 01025	store R
76523	01002	31 [01736] 00017	exit address → A
76524	01003	23 20000 00073	
76525	01004	11 20000 01023	
76526	01005	15 20000 01006	(A) → I
76527	01006	11 [30000] 01024	
76530	01007	75 30003 01011	store A, I, R on drum
76531	01010	11 01023 [74001]	
76532	01011	21 01010 00043	step
76533	01012	42 01026 01030	done?
76534	01013	16 01027 01010	restore
76535	01014	75 30036 01016	place es on MD
76536	01015	11 01000 75000	
76537	01016	75 30036 75020	place information in ES
76540	01017	11 74001 01000	
76541	01020	65 20001 01000	print on M T 2
76542	01021	75 30036 01030	restore ES
76543	01022	11 75000 01000	
76544	01023	15 01031 01002	set order
76545	01024	75 10040 01001	clear temporaries
76546	01025	11 00040 74001	
76547	01026	11 01023 74037	constants
76550	01027	00 00000 74001	
76551	01030	11 01025 20000	restore R
76552	01031	45 01736 01736	exit
76753	76753	32 76521 03100	loader parameter

CV-11

PHASE II of TRACE

77600	00117	11	40000	74000	store 40,000
77601	00120	75	31777	77673	store E.S.
77602	00121	11	00001	74001	
77603	00122	53	00000	01300	Loader parameters
77604	00123	00	00000	00000	
77605	00124	45	00000	00215	return from loader
77606	00125	37	00200	00201	print flex characters
77607	00126	45	12203	11406	Flex characters
77610	00127	22	04470	70157	
77611	00130	74	45000	00000	
77612	00131	00	00000	00000	
77613	00132	11	00124	40000	preset 40,000
77614	00133	45	00000	76575	enter loader
77615	00134	67	20010	00000	back tape
77616	00135	15	00172	00137	preset order
77617	00136	75	30003	00140	pickup A,I,R
77620	00137	11	00700	00020	
77621	00140	11	00021	20000	Test I
77622	00141	42	00170	00143	
77623	00142	42	00171	00146	
77624	00143	21	00137	00073	Step by one
77625	00144	42	00173	00136	done?
77626	00145	45	00000	00222	jump
77627	00146	21	00137	00174	Step by three
77630	00147	63	00000	00042	punch carriage return
77631	00150	55	00020	00011	shift A by 3 octal
77632	00151	37	00166	00157	punch A
77633	00152	55	00021	00006	shift C

PX 71900-8-11

77634	00153	11 00074 00006	1 → tally	
77635	00154	37 00166 00160	punch operation	
77636	00155	37 00166 00157	punch x	
77637	00156	45 00000 00175	jump	
77640	00157	11 00043 00006	3 → tally	SUBROUTINE FOR OCTAL PUNCH
77641	00160	55 10000 00003	SHIFT	
77642	00161	51 00067 20000	X TRACT DIGIT	
77643	00162	35 00167 00163	set order	
77644	00163	[00 00000 00000]	punch digit	
77645	00164	41 00006 00160	Index	
77646	00165	63 00000 00044	punch space	
77647	00166	45 00000 00000	exit	
77650	00167	63 00000 00045	prototype order	
77651	00170	13 77777 77777	constants	
77652	00171	14 77777 77777	constants	
77653	00172	00 00700 00000	constants	
77654	00173	11 01275 00000	constants	
77655	00174	00 00003 00000	constants	
77656	00175	37 00166 00157	punch y	
77657	00176	14 53002 20000	punch r	
77660	00177	45 00000 00136	jump	
77661	00200	45 00000 [00000]	exit	SUBROUTINE FOR FLEX PRINT
77662	00201	31 00200 00017	set order	
77663	00202	15 20000 00204	set order	
77664	00203	21 00200 00074	step	
77665	00204	31 00000 00044	Word → L	
77666	00205	47 00206 00200	done?	
77667	00206	54 20000 00006	shift next digit	
77670	00207	61 00000 20000	punch	

PX 71900-8-11

77671	00210	27	00040	00040	clear R
77672	00211	47	00206	00201	done?
77673	00212	11	77605	00010	set exit of loader
77674	00213	75	30200	00125	put phase II into E.S.
77675	00214	11	77603	00122	
77676	00215	37	40000	00223	set 40,000 for restoration
77677	00216	75	31777	77701	restore ES
77700	00217	11	74001	00001	
77701	00220	11	74000	40000	restore 40,000
77702	00221	56	00000	40000	halt
77703	00222	65	20010	00300	print over old date
77704	00223	64	20010	00700	read new data
77705	00224	75	10400	00134	store zeros for printing
77706	00225	11	00040	00300	

FLEXPRINT SUBROUTINE IE001 (Revised)

This short subroutine is used to print (or punch) a series of flexowriter characters from consecutive six bit positions within words which are stored in consecutive memory locations.

A code delete (octal 77) is used to signal the end of printing (or punching); any remaining digit positions in the same word are filled with zeros.

This subroutine is entered by a 37 order; the flexowriter characters to be printed (or punched) follow immediately after this 37 order.

The routine operates from a fixed MD location, resets itself completely, does not require the constant pool, and uses no temporaries.

ENTRIES:

Punch (P.T.)	37 77215 77216
Print	37 77215 77217
Drum Allocation	77213-77233 $(17)_{10}$ $(21)_8$

(Not standard)

CV-52

FLEXPRINT/PUNCH

77213	15 77217 77223	} RESETS
77214	11 77230 77225	}
77215	37 77215 [77216]	EXIT
77216	11 77231 77225	PUNCH ENTRY
77217	11 77232 10000	PRINT ENTRY
77220	31 77215 00017	} SET
77221	15 20000 77223	} PICKUP
77222	21 77215 77233	STEP EXIT
77223	31 [77232] 00052	PICK UP FLEX WORD
77224	43 10000 77213	FINISHED ?
77225	[61 00000 20000]	PRINT/PUNCH CHARACTER
77226	34 20000 00006	SHIFT NEXT CHARACTER
77227	47 77224 77220	WORD USED UP ?
77230	61 00000 20000	} PRESETS
77231	63 00000 20000	}
77232	00 00000 00077	FLAG
77233	00 00000 00001	ADDRESS STEP

PX 71900-8-52

OPERATIONS RESEARCH OFFICE
7100 Connecticut Avenue
Chevy Chase, Maryland

Complab

Coded by James Chappell | Page 1 of 21

Checked by J. Chappell | Date 13 February 1956

Machine checked by J. Chappell

Title Card Title Subroutine

Use: This routine converts alpha-numeric information coded in Flexowriter code into IBM code on cards for use as page headings, column headings, and line titles. These cards, together with data cards, can be listed to produce tabular formats, for example, on the IBM 407.

Range: A maximum of 72 letters or digits may be punched in columns 1-72, plus a 4-digit card number in columns 76-79, a 3-digit alpha-numeric deck number in columns 73-75, and a 1-column line space code in column 80.

Storage:
156 instructions, 01001b - 01232b
51 constants in program, 01233b - 01315b
36 words temporary storage in program, 01316b - 01361b
19 words temporary storage not in program, 00010b - 00032b

262 words total storage

The Convair constant storage pool is not used by this routine.

Format: The routine is coded in standard form and is self-resetting.

Parameter words: Two types of parameter words are used by this routine. See description of parameter words, page 3.

Modification: See Modification, page 7.

Timing: 1 second per card average. (See Timing, page 7.)

PX 71900-8-118

Description of Service

This subroutine was designed to allow the programmer wishing tabular output to make use of punched card equipment to produce quickly a neat printout of flexible format. The routine uses the standard two-digit octal Flexowriter codes, packed 6 to an 1103 word, as input for setting up the appropriate IBM card code for punching by the 1103 Controlled Reproducer. Since there is not an exact correspondence between all the characters used on the IBM Type 407 Tabulator and the Flexowriter, it was necessary to add some codes and delete others. The codes used are shown on page 10. Inapplicable Flexowriter codes are recognized as illegal codes and are ignored by the routine. A maximum of 12 such coded 1103 words can be punched on one card. This produces 72 characters or spaces in columns 1 - 72 of the card.

The line space code, provided in column 80, is used by the tabulator to determine the number of lines there will be between the last card printed and the card containing the line space code. Although this particular routine is not set up to punch them, the line space code is designed to include numeric data cards. Due to the peculiarities of the tabulator, the dash and the ampersand will not print on alpha-numeric cards. The complete spacing code is shown on page 10. (For 407 wiring, see Listing the Cards, page 8.)

To facilitate the arrangement of the cards for a particular table, a 4-digit card number can be punched in column 76 - 79. For distinguishing sets of cards belonging to different tables, a 3-digit alpha-numeric deck number is provided in columns 73 - 75. These features are particularly helpful if the 1103 programming makes it inconvenient to punch the cards for the lines of the table consecutively. After the cards have been punched, a simple sort on card and deck number will put them in order for listing. The deck number and card number are in addition to the 72 columns of alpha-numeric punching. These may be used at the programmer's option. (See Use of Routine, page 3.)

Use of Routine

Entry: This subroutine is coded in standard form and is entered in the usual manner:

```
RJ  01001    01002
    Parameter words
    . . . . . . . . . .
    . . . . . . . . . .
    . . . . . . . . . .
NI  (i.e., next instruction of the main routine.)
```

The number of parameter words used is flexible, depending upon the needs of the programmer. See Parameter words, below.

Parameter Words: Two types of parameter words are used by this routine which may be designated as primary parameter words and secondary parameter words. A primary parameter word is necessary for every group of cards punched. A secondary parameter word is used only if a card number, a deck number, or a code punch is desired. Any number of Primary parameter words alone or groups of primary and secondary parameter words may follow the RJ entry to the routine, and the routine will exit to the NI, provided that in every case in which a card number, deck number, or code punch is desired, a secondary parameter word follows immediately the associated primary parameter word, and that no secondary parameter word is used where no card nor deck number, nor line space code is used. The format of the two types of parameter words is as follows:

Primary parameter word	OO XXNNZ AAAAA
Secondary parameter word	KK DDDDD CCCCC

The first two octal digits of all primary parameter words must be zeros and the NI must be a legitimate 1103 machine instruction with an operation code greater than 10b since the routine determines the end of the parameter word list by testing for this condition.

X = number of cards to be punched under control of this particular parameter word; $1 \leq X \leq 77b$

N = number of coded words to be used per card; thus 6N letters or digits of information will be placed on each card; $1 \leq N \leq 14b$.

A = address of first coded 1103 word, A may be either a drum or MC address.

The desired letters or numbers are coded in Flexowriter code, 6 to an 1103 word; for example, ABC 75 would be coded as follows: 30 23160 47262

Therefore, NX consecutive coded words must be provided beginning at location A.

Z = 1 octal digit which determines the use of deck numbers, card numbers, and line space codes. A 1 in the first binary position causes the routine to punch a deck number, a 1 in the second binary position produces a card number, and a 1 in the third binary position produces a line space code. A 0 in any of the 3 positions causes the routine to bypass that portion of the routine controlled by that particular bit.

Thus the values of Z produce the following results:

Z = 0; no deck number, no card number, no line space code
1; no deck number, no card number, a line space code
2; no deck number, a card number, no line space code
3; no deck number, a card number, a line space code
4; a deck number, no card number, no line space code
5; a deck number, no card number, a line space code
6; a deck number, a card number, no line space code
7; a deck number, a card number, a line space code

If Z = 0, a Type B Parameter word should not be used. If Z = 0 and a Type B parameter is used, it will be used by the routine as the NI.

If Z ≠ 0, a secondary parameter word must be used.

If Z ≠ 0, and no secondary parameter word is used, the 1103 word following the primary parameter word will be used as the secondary parameter word.

D = address of Deck number which may be either drum or MC address. The deck number is a 3-digit alpha-numeric number coded in the same manner as the regular coded words and stored in the last 6 octal positions of an 1103 word. The same deck number will be placed on all cards punched under control of a single parameter word. The address of this word is D in the parameter word.

C = address of Card number which may be drum or MC address.

The card number must be coded in octal and will be punched as a decimal number in columns 76 - 79 of the card. The 1103 word containing the card number is made up of two parts, as follows:

HH 00000 BBBBB

where B is the octal equivalent of the number to be punched in the first card and H is the increment by which the card numbers are to be advanced. For example, if C contained

02 00000 00144

the card numbers would be 100, 102, 104, 106 . . .

The decimal value of the card number can never exceed 4 digits.

K = the line space code (i.e., octal equivalent of card row in which the column 80 punch is desired).

Card Row (Line space code)	Value of K (octal)
12	14
11	13
0	0
9	11
8	10
7	7
6	6
5	5
4	4
3	3
2	2
1	1

The ORO IBM 407 has been wired to space on this code as follows:

	Single Spacing	Suppress Spacing	Double Spacing	Quadruple Spacing
Numeric data cards	no punch	3	6	9
Alpha-numeric cards (Print wheels 1-72) (from card col. 1-72)	1	4	7	11
Alpha-numeric cards (Print wheels 73-120) (from card col. 1-48)	2	5	8	12

Since the spacing is done before printing the card, the line space code must be placed in the card on which the desired space or space suppression is wanted. Thus to achieve quadruple spacing after a heading, a 9 punch would be placed in the first data card. To print more than 1 card per line, a space suppress code will be placed in all but the first of the cards to be printed on that line.

Alarm Conditions

Code 1: A value greater than 14b has been placed on K.

Code 2: Decimal equivalent of card number exceeds 4 digits.

After typing out the appropriate code, the machine will halt on a zero stop. At this time the secondary parameter word upon which the routine is attempting to operate will appear in Q and the address of this secondary parameter word will appear in A_R. The parameter word in Q will have the normal secondary parameter word format: KK DDDDD CCCCC. Starting the machine again will cause the subroutine to exit either to the NI or return to process the next parameter word, whichever is applicable.

PX 71900-8-118

Timing

Several timed runs made on the ORO 1103 produced the following results: With the coded information stored in the magnetic core memory, times varied from a minimum of 60 cards per minute with 11 coded words per card, deck number, and line space code to a maximum of 79 cards per minute with only 1 coded word per card and no deck number, card number, or line space code. With the coded words stored in the drum, the corresponding figures were a minimum of 40 cards per minute and a maximum of 75 cards per minute.

Modification

Since this routine was assembled with the Ramo-Wooldridge one-pass assembly routine (RW-72), it can be modified by changing the directory cards. If this is done, the main program (the A00 regions) should follow in consecutive order. If desired the card image (C100 region) may be assembled elsewhere in the rapid-access memory. If it is desired to modify the octal version of the program, only 254b words should be modified.

Explanation of Dictionaries

Figure I (page 11) shows a binary breakdown of the card dictionaries. Dictionary I is used for all Flex codes < 45 and Dictionary II is used for all Flex codes > 45. It can be seen that if the appropriate dictionary were shifted row by row, the number of places left corresponding to the octal Flex code, the resulting configuration would be such that the extreme right bits of the 12 words would correspond to the IBM code of the letter, number, or symbol whose Flex code was used. In the routine, to avoid destroying the dictionary, each row is shifted in the Accumulator and handled independently. All unused or illegal codes have an entire column of 1's as a signal to the routine to ignore that code.

Listing the Cards

Several basic assumptions were made in wiring the 407 control panel for listing these cards. These were:

1. The cards will be in order when fed into the 407. Either they were punched consecutively by the 1103 or they were sorted into the correct order by use of the card and deck number. If the first card to be printed is an alpha-numeric card, it should be preceded by a blank card.

2. The carriagetape is punched to space a fixed amount at the top and bottom of each page.

3. Spacing in excess of 3 lines between printed lines will be taken care of by the use of cards which are blank in columns 1 - 72.

4. Wiring for listing the numeric data cards will be done separately for each table. The wiring for the alpha-numeric cards is the same for all tables and is permanently wired into the control panel.

5. Zero print control will be wired only for numeric data cards. It is this assumption which makes the dash and the ampersand fail to print on alpha-numeric cards.

For purposes of explaining the ORO control panel used to list these cards, one further assumption is made. This assumption is that the reader is familiar with the operation of the IBM 407 tabulator and understands how to wire it for listing. The basic principles for this can be found in pages 5 - 16, 22 - 27, and 54 of IBM Accounting Machine Type 407 Principles of Operation, Fifth Revision, International Business Machines, New York, 1953.

Complete format flexibility is obtained by each person doing the wiring necessary to list his specific numeric cards. This consists of wiring from the second read outlet for impulses from each card column to be printed into the normal print entry for the type wheel in which the column is to print. If only numbers are to be printed with no zeros to the left of the most significant digit, zero print control for all but the type wheel for the most significant digit of the number should be jack plugged. Zero print control for cases involving dollar signs, decimal points, commas, etc. is as shown in the 407 Principles of Operation Manual. When more than 10 filters, which are standard on the machine, are needed, co-selectors 13 and 14 can be used. In this case the information to be filtered can be taken into the C (i.e., common) hub of the selector and out of the corresponding T (i.e., transferred) hub of the selector.

For those who may need to set up a similar 407 control panel two problems must be solved. One of these is finding a way to print in two different type wheels from one card column. The second is to provide the line spacing indicated by the column 80 code shown on page 6.

The first of these problems was solved on the ORO board by the combined use of co-selectors and transfer print. Transfer print was picked up on all the line space codes used by alpha-numeric cards. In addition to this co-selectors 1-10 were picked up for those cards which were to print in type wheels 73-120. This means that card columns 1-48 go into the common of the co-selectors and out of the normal side into type wheels 1-48. When the co-selectors transfer because of a 2, 5, 8 or 12 punch in column 80, columns 1-48 will print in type wheels 73-120.

The wiring used to obtain the spacing is shown in Figure 2. This figure also shows the wiring used to pick up Transfer Print Entry for the printing of the alpha-numeric cards. It will be noted that a set of numeric cards which are blank in column 80 will list with single spacing. If these cards were to be double spaced, a 6 should be gang punched in column 80.

Octal Codes Used for this Subroutine

Letters	Octal Code	Numbers	Octal Code
A	30	0	03
B	23	1	52
C	16	2	74
D	22	3	70
E	20	4	64
F	26	5	62
G	13	6	66
H	05	7	72
I	14	8	60
J	32	9	33
K	36		
L	11		
M	07		
N	06		
O	03		
P	15		
Q	35		
R	12		
S	24		
T	01		
U	34		
V	17		
W	31		
X	27		
Y	25		
Z	21		

Special Symbols	
. (period)	42
, (comma)	46
□	63
$ (dollar sign)	65
* (asterisk)	67
/ (fraction bar)	50
% (per cent)	10
# (number sign)	40
@ (At sign)	41
Space	04

All other 2-digit octal combinations are recognized as illegal and are ignored.

DICTIONARY I

RAWOOP CODING	OCTAL CODING	CARD ROW	T		O	Sp	H	N	M	%	L	R	G	I	P	C	V	E	Z	D	B	S	Y	F	X	A	W	J	9	U	Q	K	0	#	@	.		
OCD00	01262	12	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	1
OCD01	01263	11	0	1	1	0	0	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	1
OCD02	01264	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	0	1	1
OCD03	01265	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1
OCD04	01266	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
OCD05	01267	3	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
OCD06	01270	4	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1
OCD07	01271	5	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
OCD08	01272	6	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1
OCD09	01273	7	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
OCD10	01274	8	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1
OCD11	01275	9	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

DICTIONARY II

				,		/		1						8		5	□	4	$	6	*	3		7		2												
OCD12	01276	12	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD13	01277	11	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD14	01300	0	1	1	1	1	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD15	01301	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD16	01302	2	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
OCD17	01303	3	1	1	1	0	1	0	1	1	1	1	1	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD18	01304	4	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD19	01305	5	1	0	1	0	1	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD20	01306	6	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD21	01307	7	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD22	01310	8	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
OCD23	01311	9	1	0	1	0	1	0	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

FIGURE I

CARD TITLE SUBROUTINE

RAWOOP CODING		OCTAL CODING	EXPLANATION
D	01A00 00512	01000 00 00000 00000	DISSECTION OF PARAMETER WORDS
D	02A00 00544	01040 00 00000 00000	FORMULATION OF CARD IMAGE
D	03A00 00600	01130 00 00000 00000	FORMULATION OF DECK NUMBER
D	04A00 00611	01143 00 00000 00000	FORMULATION OF CARD NUMBER
D	05A00 00629	01165 00 00000 00000	FORMULATION OF SPACE CODE
D	06A00 00642	01202 00 00000 00000	PUNCHING OF CARD
D	0AP00 00659	01223 00 00000 00000	ALARM PRINT ROUTINE
D	00K00 00667	01233 00 00000 00000	CONSTANTS
D	0CD00 00690	01262 00 00000 00000	CARD DICTIONARY
D	0PT00 00712	01310 00 00000 00000	POWERS OF TEN
D	0C100 00718	01316 00 00000 00000	CARD IMAGE
D	00T00 00008	00010 00 00000 00000	TEMPORARY STORAGE

01A00 00 00000 00000	01000 00 00000 00000	NOT USED
01A01 MJ 00000 [00000]	01001 45 00000 [00000]	NORMAL EXIT
01A02 SP 01A01 00015	01002 31 01001 00017	ENTRANCE
01A03 TU A 01A04	01003 15 20000 01004	PRIMARY PARAMETER WORD
01A04 TP [B] 00T00	01004 11 [30000] 00010	⟶10b
01A05 SP 00T00 00015	01005 31 00010 00017	STORE ADDRESS OF FIRST
01A06 TU A 00K14	01006 15 20000 01251	CODED WORD
01A07 LQ 00T00 10012	01007 55 00010 10014	
01A08 QT 00K00 A	01010 51 01233 20000	X - 1⟶12b
01A09 ST 0PT05 00T02	01011 36 01315 00012	

PX 71900-8-118

01A10	LQ	00T00	10018	01012	55	00010	10022		
01A11	QT	00K00	A	01013	51	01233	20000	N - 1 → 13b	
01A12	ST	0PT05	00T03	01014	36	01315	00013		
01A13	QJ	01A16	01A14	01015	44	01020	01016	DECK NUMBER?	
01A14	QJ	01A16	01A15	01016	44	01020	01017	CARD NUMBER?	TEST Z
01A15	QJ	01A16	01A31	01017	44	01020	01037	SPACE CODE?	
01A16	TP	0PT05	A	01020	11	01315	20000		
01A17	SA	01A01	00015	01021	32	01001	00017	ADDRESS OF	
01A18	TU	A	01A21	01022	15	20000	01025	SECONDARY PARAMETER	
01A19	TU	A	01A22	01023	15	20000	01026	WORD DETERMINED	
01A20	TU	A	01A28	01024	15	20000	01034		
01A21	TU	[B]	06A00	01025	15	[30000]	01202	ADDRESS OF DECK NUMBER STORED	
01A22	SP	[B]	00015	01026	31	[30000]	00017	ADDRESS OF CARD	
01A23	TU	A	01A24	01027	15	20000	01030	NUMBER DETERMINED	
01A24	SP	[B]	00042	01030	31	[30000]	00052	H → 11b	
01A25	TP	A	00T01	01031	11	20000	00011		
01A26	SS	A	00030	01032	34	20000	00036	CARD NUMBER → 14b	
01A27	TP	A	00T04	01033	11	20000	00014		
01A28	LQ	[B]	10006	01034	55	[30000]	10006	K → 32b	
01A29	QT	00K00	00T18	01035	51	01233	00032		
01A30	RA	01A01	0PT05	01036	21	01001	01315	INCREASE EXIT	
01A31	RA	01A01	0PT05	01037	21	01001	01315		
02A00	TP	0PT05	00T05	01040	11	01315	00015	1 → 15b	
02A01	RP	30036	02A03	01041	75	30044	01043	CLEAR CARD IMAGE	
02A02	RS	0C100	0C100	01042	23	01316	01316		

PX 71900-8-118

02A03	TV	02A02	02A34	01043	16	01042	01102	
02A04	TU	00K14	02A10	01044	15	01251	01052	
02A05	TU	02A03	02A12	01045	15	01043	01054	SETUP FOR FIELD I
02A06	TU	02A02	02A21	01046	15	01042	01065	
02A07	TU	02A02	02A36	01047	15	01042	01104	
02A08	TP	00T03	00T06	01050	11	00013	00016	N - 1→16b
02A09	RS	00T07	00T07	01051	23	00017	00017	CLEAR 17b
02A10	TP	B	00T08	01052	11	30000	00020	CODED WORD→20b
02A11	TP	00K03	00T09	01053	11	01236	00021	5→21b
02A12	TV	02A02	02A25	01054	16	01042	01071	SETUP v OF 1071b
02A13	LQ	00T08	00006	01055	55	00020	00006	MASK LETTER INTO A
02A14	QT	00K00	A	01056	51	01233	20000	
02A15	TJ	00K04	02A17	01057	42	01237	01061	TEST FOR APPROPRIATE DICTIONARY
02A16	RA	A	00K05	01060	21	20000	01240	SETUP SHIFT
02A17	AT	00K06	02A23	01061	35	01241	01067	DICTIONARY COMMAND
02A18	TP	00K07	00T10	01062	11	01242	00022	13→22b
02A19	RS	00T11	00T11	01063	23	00023	00023	CLEAR BIT COUNTER
02A20	RP	20012	02A22	01064	75	20014	01066	SHIFT CARD IMAGE
02A21	LQ	0C100	00001	01065	55	01316	00001	1 PLACE LEFT
02A22	RA	00T07	0PT05	01066	21	00017	01315	INCREASE COLUMN COUNTER
02A23	LQ	0CD00	A	01067	55	01262	20000	SHIFT APPROPRIATE DICTIONARY AND LEAVE IN A
02A24	TP	0PT05	Q	01070	11	01315	10000	1 BIT MASK→Q
02A25	QS	A	0C100	01071	53	20000	01316	MASK FINAL BIT OF ROW OF DICTIONARY INTO ROW OF IMAGE
02A26	QT	A	A	01072	51	20000	20000	ISOLATE FINAL BIT OF IMAGE ROW
02A27	AT	00T11	00T11	01073	35	00023	00023	SUM BITS OF COLUMN
02A28	RA	02A23	00K02	01074	21	01067	01235	INCREASE DICTIONARY ROW

PX 71900-8-118

02A29 RA 02A25 OPT05	01075 21 01071 01315	INCREASE IMAGE ROW
02A30 IJ 00T10 02A23	01076 41 00022 01067	12 ROWS?
02A31 TP 00T11 A	01077 11 00023 20000	TEST FOR ILLEGAL CODE,
02A32 TJ 00K03 02A38	01100 42 01236 01106	IF LEGAL, GO TO 1106b
02A33 RP 10012 02A35	01101 75 10014 01103	MASK OUT ILLEGAL BITS
02A34 QS 00K02 [0C100]	01102 53 01235 [01316]	
02A35 RP 20012 02A37	01103 75 20014 01105	MOVE IMAGE 1 PLACE RIGHT
02A36 LQ [0C100] 00035	01104 55 [01316] 00043	
02A37 RS 00T07 OPT05	01105 23 00017 01315	DECREASE COLUMN COUNTER
02A38 TP 00T07 A	01106 11 00017 20000	TEST FOR COMPLETION
02A39 RJ 00K18 02A46	01107 42 01255 01116	OF FIELD
02A40 TU 02A41 02A12	01110 15 01111 01054	SETUP FOR FIELD II
02A41 TU 00K17 02A21	01111 15 01254 01065	
02A42 TU 00K17 02A36	01112 15 01254 01104	
02A43 TV 00K17 02A34	01113 16 01254 01102	
02A44 RS 00T07 00T07	01114 23 00017 00017	
02A45 RS 00T05 00T05	01115 23 00015 00015	
02A46 IJ 00T09 02A12	01116 41 00021 01054	CODED WORD COMPLETED?
02A47 RJ 02A47 02A48	01117 37 01117 01120	OPTIONAL EXIT
02A48 RA 02A10 00K02	01120 21 01052 01235	INCREASE ADDRESS OF CODED WORD
02A49 IJ 00T06 02A10	01121 41 00016 01052	N WORDS COMPLETED?
02A50 IJ 00T05 02A52	01122 41 00015 00124	ONE OR TWO FIELDS USED?
02A51 TP 00K05 A	01123 11 01240 20000	SETUP SHIFT COMMAND
02A52 AT 00K09 A	01124 35 01244 20000	
02A53 ST 00T07 02A55	01125 36 00017 01127	
02A54 RP 20012 03A00	01126 75 20014 01130	MOVE CARD IMAGE TO LEFT
02A55 LQ [0C100] 0000	01127 55 [01316] 00000	

03A00	LQ	00T00	10018	01130	55	00010	10022	DECK NUMBER USED?
03A01	QJ	03A02	04A00	01131	44	01132	01143	IF NOT, GO TO 1143b
03A02	TU	03A03	02A12	01132	15	01133	01054	
03A03	TU	00K16	02A21	01133	15	01253	01065	
03A04	TU	00K16	02A36	01134	15	01253	01104	SETUP FOR FIELD III
03A05	TU	02A10	00K14	01135	15	01052	01251	
03A06	TU	06A00	02A10	01136	15	01202	01052	
03A07	TV	00K16	02A34	01137	16	01253	01102	
03A08	RJ	02A47	02A09	01140	37	01117	01051	FORM DECK NUMBER IN IMAGE
03A09	RP	20012	04A00	01141	75	20014	01143	POSITION DECK NUMBER
03A10	LQ	0C124	00005	01142	55	01346	00005	IN FIELD III
04A00	LQ	00T00	10019	01143	55	00010	10023	CARD NUMBER USED?
04A01	QJ	04A02	05A00	01144	44	01145	01165	IF NOT, GO TO 1165b
04A02	TP	00T04	A	01145	11	00014	20000	CARD NUMBER→A
04A03	TJ	00K01	04A06	01146	42	01245	01151	
04A04	SP	00K21	00042	01147	31	01260	00052	ALARM IF CARD NUMBER 4 DIGITS
04A05	MJ	00000	0AP00	01150	45	00000	01223	
04A06	RP	30004	04A08	01151	75	30004	01153	CONVERT TO DECIMAL AND
04A07	DV	0PT02	00T12	01152	73	01312	00024	STORE DIGITS IN 24b - 27b
04A08	TV	04A07	04A12	01153	16	01152	01157	SETUP v OF 1157b
04A09	TP	00K08	00T05	01154	11	01243	00015	3→15b
04A10	LQ	0PT05	10005	01155	55	01315	10005	FLOATING 1→Q
04A11	TV	00K01	04A14	01156	16	01234	01161	SETUP v OF 1161b
04A12	RA	04A14	00T12	01157	21	01161	00024	
04A13	LQ	Q	35	01160	55	10000	00043	SHIFT FLOATING 1
04A14	QS	Q	0C126	01161	53	10000	01350	MASK BIT INTO IMAGE
04A15	RA	04A12	0PT05	01162	21	01157	01315	
04A16	IJ	00T05	04A11	01163	41	00015	01156	4 DIGITS FORMED?
04A17	RA	00T04	00T01	01164	21	00014	00011	INCREASE CARD NUMBER

05A00 LQ 00T00 10020	01165 55 00010 10024	CODE PUNCH USED?
05A01 QJ 05A02 06A00	01166 44 01167 01202	IF NOT, GO TO 1202b
05A02 TP 00T18 A	01167 11 00032 20000	K→A
05A03 TJ 00K19 05A06	01170 42 01256 01173	
05A04 SP 00K20 00042	01171 31 01257 00052	ALARM IF K→14b
05A05 MJ 00000 0AP00	01172 45 00000 01223	
05A06 TJ 00K22 05A09	01173 42 01261 01176	
05A07 ST 00K22 A	01174 36 01261 20000	DETERMINE ROW OF CARD
05A08 TN A A	01175 13 20000 20000	IMAGE IN WHICH CODE
05A09 RA A 00K01	01176 21 20000 01234	PUNCH IS TO BE PLACED
05A10 TV A 05A12	01177 16 20000 01201	
05A11 TP 0PT05 Q	01200 11 01315 10000	1 BIT MASK→Q
05A12 QS 0PT05 [0C126]	01201 53 01315 [01350]	MASK BIT INTO ROW OF IMAGE
06A00 EF 00000 00K11	01202 17 00000 01246	CYCLE REPRODUCER
06A01 TP 00K07 00T01	01203 11 01242 00022	ROW COUNTER = 12
06A02 RP 30003 06A04	01204 75 30003 01206	SETUP v OF EW ORDERS
06A03 TV 00K12 06A04	01205 16 01247 01206	
06A04 EW 00000 [0C135]	01206 77 00000 [01361]	
06A05 EW 10000 [0C111]	01207 77 10000 [01331]	EW ORDERS
06A06 EW 10000 [0C123]	01210 77 10000 [01345]	
06A07 RP 20003 06A09	01211 75 20003 01213	DECREASE v OF EW ORDERS
06A08 RS 06A04 0PT05	01212 23 01206 01315	
06A09 IJ 00T10 06A04	01213 41 00022 01206	12 ROWS?
06A10 IJ 00T02 02A00	01214 41 00012 01040	X CARDS COMPLETED?
06A11 SP 01A01 00015	01215 31 01001 00017	CONTENTS OF EXIT
06A12 TU A 06A13	01216 15 20000 01217	ADDRESS→Q
06A13 LQ [B] 10006	01217 55 [30000] 10006	

FX 71900-8-118

```
06A14 QT 00K00 A          01220 51 01233 20000   OPERATION PORTION OF EXIT->A

06A15 TJ 00K15 01A02      01221 42 01252 01002   TEST FOR NI OR PARAMETER
06A16 MJ 00000 01A01      01222 45 00000 01001   WORD.  IF PARAMETER WORD,
                                                 GO TO 1002b.  IF NI, GO TO
                                                 EXIT AT 1001b

0AP00 PR 00000 A          01223 61 00000 20000
0AP01 SS A     00006      01224 34 20000 00006
0AP02 ZJ 0AP00 0AP03      01225 47 01223 01226
0AP03 TN OPT05 A          01226 13 01315 20000
0AP04 SA 01A01 00015      01227 32 01001 00017   ALARM PRINT ROUTINE
0AP05 TU A     0AP06      01230 15 20000 01231
0AP06 TP B     Q          01231 11 30000 10000
0AP07 MS 00000 06A11      01232 56 00000 01215

00K00 00 00000 00077 B    01233 00 00000 00077
00K01 00 00000 0C126      01234 00 00000 01350
00K02 00 00001 00000 B    01235 00 00001 00000
00K03 00 00000 00005 B    01236 00 00000 00005
00K04 00 00000 00045 B    01237 00 00000 00045
00K05 00 00014 00000 B    01240 00 00014 00000
00K06 LQ 0CD00 A          01241 55 01262 20000   CONSTANTS
00K07 00 00000 00013 B    01242 00 00000 00013
00K08 00 00000 00003 B    01243 00 00000 00003
00K09 LQ 0C100 00036      01244 55 01316 00044
00K10 00 00000 23417 B    01245 00 00000 23417
00K11 00 00000 00112 B    01246 00 00000 00112
00K12 00 00000 0C135      01247 00 00000 01361
00K13 00 00000 0C111      01250 00 00000 01331
00K14 00 00000 0C123      01251 00 00000 01345
00K15 00 00000 00010 B    01252 00 00000 00010
```

00K16	00	0C124	0C124		01253	00	01346	01346	CONSTANTS
00K17	00	0C112	0C112		01254	00	01332	01332	
00K18	00	00000	00044	B	01255	00	00000	00044	
00K19	00	00000	00015	B	01256	00	00000	00015	
00K20	16	03222	00452	B	01257	16	03222	00452	
00K21	16	03222	00474	B	01260	16	03222	00474	
00K22	00	00000	00012	B	01261	00	00000	00012	
0CD00	22	03254	50007	B	01262	22	03254	50007	DICTIONARY I
0CD01	31	54400	02306	B	01263	31	54400	02306	
0CD02	60	20123	24446	B	01264	60	20123	24446	
0CD03	20	00000	12006	B	01265	20	00000	12006	
0CD04	20	00006	00106	B	01266	20	00006	00106	
0CD05	60	10200	00027	B	01267	60	10200	00027	
0CD06	20	60010	00416	B	01270	20	60010	00416	
0CD07	21	00140	00006	B	01271	21	00140	00006	
0CD08	30	00000	44006	B	01272	30	00000	44006	
0CD09	20	02400	20006	B	01273	20	02400	20006	
0CD10	22	20001	00237	B	01274	22	20001	00237	
0CD11	20	05020	01006	B	01275	20	05020	01006	
0CD12	52	76501	27777	B	01276	52	76501	27777	DICTIONARY II
0CD13	52	76425	27777	B	01277	52	76425	27777	
0CD14	76	76401	27777	B	01300	76	76401	27777	
0CD15	57	76401	37777	B	01301	57	76401	37777	
0CD16	52	76401	37777	B	01302	52	76401	37777	
0CD17	72	76423	27777	B	01303	72	76423	27777	
0CD18	52	76545	27777	B	01304	52	76545	27777	
0CD19	52	76601	27777	B	01305	52	76601	27777	

OCD20	52	76411	27777	B	01306	52	76411	27777	
OCD21	52	76401	67777	B	01307	52	76401	67777	
OCD22	72	77525	27777	B	01310	72	77525	27777	
OCD23	52	76401	27777	B	01311	52	76401	27777	
OPT02	00	00000	01750	B	01312	00	00000	01750	10^3
OPT03	00	00000	00144	B	01313	00	00000	00144	10^2
OPT04	00	00000	00012	B	01314	00	00000	00012	10^1
OPT05	00	00000	00001	B	01315	00	00000	00001	10^0
OC100	00	00000	00000		01316	00	00000	00000	

CARD IMAGE FORMED HERE

1316b - 1361b

OC136	00 00000 00000	01361	00	00000	00000	
START	XXXXX	00000	45	00000	XXXXX	START CARD
		40000	45	00000	XXXXX	
	MEMORY SUM	75202	00	00000	00123	
	MEMORY SUM	75203	77	64233	75420	

PX 7190C-8-118

Figure 2: Wiring for line spacing and picking up Transfer Print Entry on alpha-numeric cards.

1. Numbers beside the pick up hubs of selectors are the digits from the first read of column 80 which are wired into these hubs via digit selector A.

2. These co-selectors are used to print alpha-numeric columns 1-48 from either type wheels 1-48 when the selectors are normal or from type wheels 73-120 when these selectors are picked up.

3. These co-selectors are picked up whenever Transfer Print Entry is not picked up. They, therefore, can be used as 10 extra positions for filtering special symbols in zero print control.

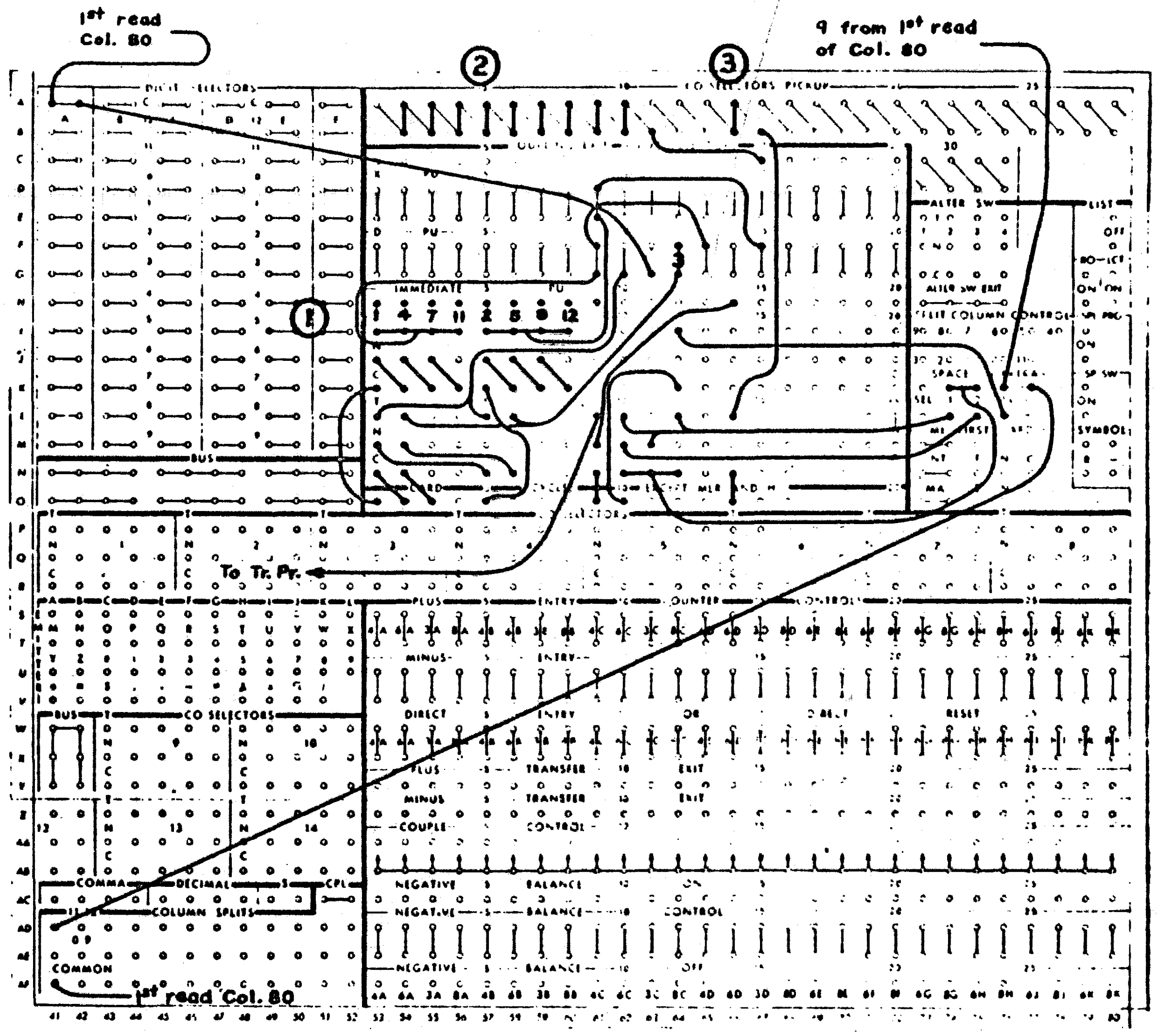


Figure 2: Wiring for line spacing and picking up Transfer Print Entry on alpha-numeric cards.

1. **Numbers beside the pick up hubs of selectors are the digits from the first read of column 80 which are wired into these hubs via digit selector A.**

2. **These co-selectors are used to print alpha-numeric columns 1-48 from either type wheels 1-48 when the selectors are normal or from type wheels 73-120 when these selectors are picked up.**

3. **These co-selectors are picked up whenever Transfer Print Entry is not picked up. They, therefore, can be used as 10 extra positions for filtering special symbols in zero print control.**

WHITE SANDS PROVING GROUND

Computing Branch, FDL

Prepared by K. R. Webster
ERA Engineer

Date 26 Jan 1956

Title Octal Card Read

Use: Read into Computer cards punched in octal form. Cards can have any number of words from 1 to 6. Cards used for this program can also be listed on tabulator as well as read into machine. If no. of words on card are 6, columns 1 thru 72 must have one punch in each column and columns 73 to 77 must have one punch. If a column is missed, the program will come to a 0 stop indicating an error on the card. This does not apply if all the words in a field are 0. Then that field can be blank. Program will stop after one blank card is read. For no. of words less than 6, only the columns needed have to be punched. Program requires 1 blank card at bottom of deck of cards and several blanks on top. Insert for words is punched in columns 73 thru 77 and number of words in column 80.

Initial State:

1. F.1 Switch in 00000 Position

2. Set PAK to 45200

3. If operation a card at a time is desired select MS 1. Words will not be placed in storage until next card is run.

Temporary Storage: 00000 to 00202

Entrance Address: 45200

Restrictions: Will not load into ES from 00000 to 00202.

```
      45200  75 30203 00062   block transfer to ES
      45201  11 45202 00000
00000 45202  45 00000 00062   jump to start
00001 45203  00 00000 00105   EF constants
00002 45204  00 00000 00104
00003 45205  00 00000 00100
00004 45206  00 00000 00000   row 9 constant
00005 45207  00 00000 00000   row 8 constant
00006 45210  70 00000 00000   row 7 constant
00007 45211  60 00000 00000   row 6 constant
00010 45212  50 00000 00000   row 5 constant
00011 45213  40 00000 00000   row 4 constant
00012 45214  30 00000 00000   row 3 constant
00013 45215  20 00000 00000   row 2 constant
00014 45216  10 00000 00000   row 1 constant
00015 45217  00 00000 00000   row 0 constant
00016 45220  00 00000 00004   insert index
00017 45221  00 00000 00013   digit index
00020 45222  00 00000 00012   row index
00021 45223  00 00000 00001   field index
00022 45224  00 00043 00000 } card image constants
00023 45225  00 00044 00000 }
00024 45226  00 00045 00000 }
00025 45227  00 00046 00000   temp storage constant
```

PX 71900-8-119

00026	45230	00 00000 00002	word index
00027	45231	00 00004 00000	start of row constants
00030	45232	00 00001 00000	constant for advancing addresses
00031	45233	00 30000 00000	constant for repeat
00032	45234	00 00000 00371	field III constant
00033	45235	00 30006 00000	constant
00034	45236	00 00000 00000	conversion constant
00035	45237	00 00000 00000	j----n for repeat
00036	45240	00 00000 00000	row index
00037	45241	00 00000 00000	digit index
00040	45242	00 00000 00000	field index
00041	45243	00 00000 00000	word index
00042	45244	00 00000 00000	insert index
00043	45245	00 00000 00000	card image III
00044	45246	00 00000 00000	II
00045	45247	00 00000 00000	II
00046	45250	00 00000 00000	Temp storage for converted words (00046–00053)
00047	45251	00 00000 00000	
00050	45252	00 00000 00000	
00051	45253	00 00000 00000	
00052	45254	00 00000 00000	
00053	45255	00 00000 00000	
00054	45256	00 00000 00000	insert address
00055	45257	00 00000 00000	no. of words on card

PX 71900-8-119

00056	45260	00 00000 00000	field I addition
00057	45261	00 00000 00000	field II addition
00060	45262	00 00000 00000	field III addition
00061	45263	00 00000 00000	blank
00062	45264	17 00000 00002	position cards
00063	45265	17 00000 00002	
00064	45266	17 00000 00001	read
00065	45267	15 00035 00067	set up repeat instruction
00066	45270	16 00054 00070	set up transfer instruction
00067	45271	75 00000 00071	store converted words away
00070	45272	11 00046 00000	
00071	45273	75 10013 00073	clear temporary storage
00072	45274	11 00004 00046	
00073	45275	11 00020 00036	set row index
00074	45276	15 00027 00141	set up row constant field I
00075	45277	15 00027 00152	set up row constant field II
00076	45300	15 00027 00160	set up row constant for 2nd and 3rd words
00077	45301	15 00027 00172	set up for no. of words
00100	45302	37 00133 00125	jump to read subroutine
00101	45303	37 00202 00136	jump to conversion subroutine
00102	45304	45 00000 00100	jump to read subroutine
00103	45305	11 00031 00035	
00104	45306	11 00055 20000	puts no. of words in A
00105	45307	47 00106 00122	tests for 0 words

PX 71900-8-119

00106	45310	54 00055 00017	shift to U portion
00107	45311	21 00035 00055	forms j------n
00110	45312	43 00033 00112	tests for less than 6 words
00111	45313	45 00000 00121	jump to next card
00112	45314	11 00056 20000	field I addition →A
00113	45315	47 00120 00114	Test for 0
00114	45316	11 00057 20000	field II addition → A
00115	45317	47 00120 00116	Test for 0
00116	45320	11 00060 20000	field III addition ⟶ A
00117	45321	43 00032 00121	test for proper addition
00120	45322	56 00000 00062	Stop for error or end of cards
00121	45323	45 00000 00064	back to next card
00122	45324	75 00004 00124	clear read side
00123	45325	17 00000 00003	
00124	45326	45 00000 00120	jump to stop
00125	45327	76 00000 00043	
00126	45330	76 10000 00044	read subroutine
00127	45331	76 10000 00045	
00130	45332	21 00056 00044	field addition I
00131	45333	21 00057 00045	field addition II
00132	45334	21 00060 00043	field addition III
00133	45335	41 00036 30000	row index
00134	45336	56 10000 00103	optional stop after every card
00135	45337	45 00000 00103	superfluous

conversion subroutine

00136	45340	15	00023	00142	puts field I in transfer
00137	45341	15	00025	00146	puts 1st conversion address in place
00140	45342	11	00021	00040	sets field index
00141	45343	11	02000	00034	puts conversion constant in temp storage
00142	45344	11	30000	10000	puts field I in Q
00143	45345	11	00026	00041	sets word counter
00144	45346	11	00017	00037	sets digit counter
00145	45347	44	00146	00147	check for 1
00146	45350	21	02000	00034	adds converted digit to cell
00147	45351	31	00034	00105	positions conversion constant
00150	45352	11	20000	00034	
00151	45353	41	00037	00145	digit index
00152	45354	11	02000	00034	sets conversion cell for 2nd word
00153	45355	21	00146	00030	adds 1 to temp storage address
00154	45356	41	00041	00144	word index
00155	45357	15	00024	00142	puts field II word in place
00156	45360	41	00040	00141	field index
00157	45361	55	00043	00034	shifts field III word
00160	45362	11	02000	00034	conversion word to temp storage
00161	45363	31	00034	00063	positions conversion constant
00162	45364	11	20000	00034	
00163	45365	11	00016	00042	sets insert index
00164	45366	44	00165	00166	check for 1
00165	45367	21	00054	00034	adds converted digit to temp storage

PX 71900-8-119

00166	45370	31	00034	00105	shifts constant and puts it away
00167	45371	11	20000	00034	
00170	45372	41	00042	00164	insert index
00171	45373	55	10000	00002	shift pattern for no. of words
00172	45374	31	00000	00047	positions conversion constant
00173	45375	11	20000	00034	
00174	45376	44	00175	00176	check for 1
00175	45377	21	00055	00034	adds to temp storage
00176	45400	21	00141	00030	
00177	45401	21	00152	00030	adds 1 to change conversion constant
00200	45402	21	00160	00030	
00201	45403	21	00172	00030	
00202	45404	45	00000	30000	jump back to main routine

PX 71900-8-119

WHITE SANDS PROVING GROUND

Computing Branch, FDL

Prepared by L. Graham Date 20 Jan 1956

Checked by ____________

Computer
Checked by L. Graham

Title Octal Card Dump

Use: Dumps 3 octal words on each Field I and Field II, and the insert address for the card is punched in cols. 73-77 inc.

Initial State: Set IA in U of Q, and No. words set in V of Q. PAK at 5Y500 Y=1 or 2

Set not to read row 12.

Range: Will dump Drum from 40000 - 77777.

If Y=1, will dump ES from 00000-00477 and 00710-01777.

If Y=2, will dump ES from 00000-01477 and 01710-01777.

Limitation: Program must operate from ES.

PX 71900-8-120

```
Drum  ES

5Y502 0X500  17 00000 0X603  Pick punch card
5Y503 0X501  17 00000 0X603  Pick punch card
5Y504 0X502  11 10000 0X605  (Q) = (1A)2^15 + N → 0X605
5Y505 0X503  11 0X617 0X606 } clear both cells
5Y506 0X504  11 0X617 0X607 }
5Y507 0X505  16 0X605 0X606  N → V of 0X606
5Y510 0X506  15 0X605 0X607  IA → U of 0X607
5Y511 0X507  11 0X606 20000  N → A
5Y512 0X510  73 0X610 0X612  N/6 = n-1 → 0X615
5Y513 0X511  47 0X512 0X513  Test remainder for zero
5Y514 0X512  11 20000 0X707  No. words on last card
5Y515 0X513  21 0X612 0X614  counter = n for No. of cards.
5Y516 0X514  15 0X605 0X516
5Y517 0X515  75 30006 0X517 }                                    0X600
5Y520 0X516  11[30000]0X631 } Transfer 6 words to be punched  0X514
5Y521 0X517  15 0X516 0X637
5Y522 0X520  17 00000 0X604  Pick punch card, and punch
5Y523 0X521  15 0X627 0X532
5Y524 0X522  15 0X624 0X526
5Y525 0X523  11 0X614 0X640  1 → 0X640.  counter for Fields
5Y526 0X524  75 10041 0X527 }
5Y527 0X525  11 0X617 0X646 } clear card image
5Y530 0X526  21[0X675]0X644  constant for forming card image 0X522
5Y531 0X527  11 0X613 0X645                                      0X546
5Y532 0X530  11 0X622 0X644  2^35 → Floating 1                  0X524
```

PX 71900-8-120

Drum	ES					
5Y533	OX531	11	OX615	OX642	counter for end of word	OX544
5Y534	OX532	11	OX637	OX643		OX521
5Y535	OX533	11	OX526	OX540		OX542
5Y536	OX534	55	OX643	00003	Shift No. to be punched	
5Y537	OX535	51	OX620	20000		
5Y540	OX536	54	20000	00017		
5Y541	OX537	35	OX540	OX540		
5Y542	OX540	21	OX665	OX644		
5Y543	OX541	55	OX644	00043	Shift floating 1	
5Y544	OX542	41	OX642	OX533	if word is not complete, jump	
5Y545	OX543	21	OX532	OX621		
5Y546	OX544	41	OX645	OX531	If ~~a~~ field is completed, proceed	
5Y547	OX545	15	OX625	OX526		
5Y550	OX546	41	OX640	OX527	Jump to OX527 unless fields I and II are complete	
5Y551	OX547	15	OX626	OX526		
5Y552	OX550	11	OX623	OX642	counter = 4 to punch IA	
5Y553	OX551	11	OX622	OX644	Floating 1	
5Y554	OX552	55	OX644	00010	shift Floating 1	
5Y555	OX553	11	OX637	OX643		
5Y556	OX554	55	OX643	00006	$(IA)2^{21}$	
5Y557	OX555	11	OX526	OX562		OX564
5Y560	OX556	55	OX643	00003		
5Y561	OX557	51	OX620	20000		
5Y562	OX560	54	20000	00017		
5Y563	OX561	35	OX562	OX562		

Drum	ES					
5Y564	0X562	21	0X703	0X644		
5Y565	0X563	55	0X644	00043	Shift Floating 1	
5Y566	0X564	41	0X642	0X555	Test for counter of 4	
5Y567	0X565	21	0X703	0X614	Add into 6 row, last column	
5Y570	0X566	11	0X616	0X641	counter = 10 for 11 rows.	
5Y571	0X567	75	30003	0X571	} fix EW'S for 9 row	
5Y572	0X570	16	0X624	0X571		
5Y573	0X571	77	00000	30000		0X576
5Y574	0X572	77	10000	30000		
5Y575	0X573	77	10000	30000		
5Y576	0X574	75	20003	0X576		
5Y577	0X575	23	0X571	0X614		
5Y600	0X576	41	0X641	0X571	Row counter	
5Y601	0X577	21	0X516	0X611	Increase next IA by 00006	
5Y602	0X600	41	0X612	0X515	Jump back to punch another card.	
5Y603	0X601	56	00000	0X500	Stop.	
5Y604	0X602	00	00000	00000		
5Y605	0X603	00	00000	00110		
5Y606	0X604	00	00000	00112		
5Y607	0X605	00	30000	30000	(Q)	0X502
5Y610	0X606	00	00000	00000	Temp. Storage N	
5Y611	0X607	00	00000	00000	Temp. Storage (IA) 2^{15}	
5Y612	0X610	00	00000	00006		
5Y613	0X611	00	00006	00000		
5Y614	0X612	00	00000	00000		
5Y615	0X613	00	00000	00002		
5Y616	0X614	00	00000	00001		

FX 71900-8-120

Drum	ES				
5Y617	0X615	00	00000	00013	
5Y620	0X616	00	00000	00012	
5Y621	0X617	00	00000	00000	
5Y622	0X620	00	00000	00007	Q EXTractor
5Y623	0X621	00	00001	00000	
5Y624	0X622	40	00000	00000	Floating 1
5Y625	0X623	00	00000	00004	
5Y626	0X624	00	0X647	0X706	
5Y627	0X625	00	0X662	0X660	
5Y630	0X626	00	0X675	0X673	
5Y631	0X627	00	0X631	00000	

FX 71900-8-120

WHITE SANDS PROVING GROUND

Computing Branch

Prepared by L. GRAHAM Page 1

Checked by ______ Date 18 January 1956

Computer
Checked by L. GRAHAM

Title Check on High Speed Punch When Punching Bioctal Tapes

1. Read in check tape to 52700.

2. Punch bioctal tape.

3. Set PAK at 52700.
   Set Q at (IA) 2^{15} + n *
   Place new bioctal tape in Reader.
   Start.

4. As soon as New Tape is read in set PAK at 52713.
   Turn printer on.
   Runs until 56 stop.

Results Obtained:

Any error in new punched tape will cause Printer to print correct word (an ES address) and incorrect word.

*If n $>$ 1737, change V of 52710 and U of 52723 to a common MD address.

52700	11	10000	52712	
52701	75	30013	00003	
52702	11	52703	00003	
52703	11	00011	10000	00003
52704	15	00012	00010	00004
52705	54	00012	00017	00005
52706	53	20000	00007	00006
52707	75	30000	45000	00007
52710	11	00000	60000	00010
52711	00	07777	00000	00011
52712	00	70000	01520	00012
52713	75	30030	00001	
52714	11	52715	00001	
52715	11	52712	10000	00001
52716	11	10000	00027	00002
52717	15	10000	00010	00003
52720	23	00030	00030	00004
52721	16	10000	00030	00005
52722	23	00030	00026	00006
52723	11	60000	20000	00007
52724	43	00000	00021	00010
52725	11	00026	00032	00011
52726	15	00007	00032	00012
52727	11	00032	10000	00013

52730	37	50644	50500	00014
52731	11	00026	00032	00015
52732	15	00010	00032	00016
52733	11	00032	10000	00017
52734	37	50644	50500	00020
52735	21	00007	00025	00021
52736	21	00010	00025	00022
52737	41	00030	00007	00023
52740	56	00000	70000	00024
52741	00	00001	00000	00025
52742	00	00000	00001	00026
52743	00	00000	00000	00027

PX 71900-8-121

ANALYSIS
PREPARED BY W. G. Gerkin
CHECKED BY
REVISED BY

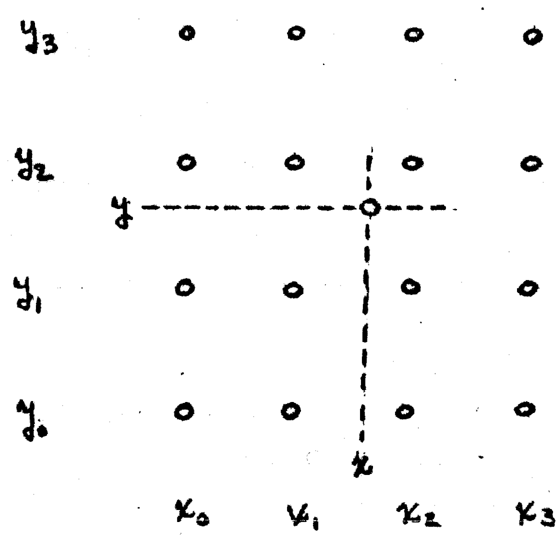
CONVAIR
A DIVISION OF GENERAL DYNAMICS CORPORATION
SAN DIEGO
CV-122

REPORT NO. ZM 491
MODEL All
DATE 3-5-56

4 POINT LAGRANGE INTERPOLATION SUBROUTINE CN 002

This subroutine may be used to interpolate in a tabulated function of one or two variables (one way or two way interpolation, respectively). There are two entrances to this subroutine, one for one way interpolation, the other for two way interpolation.

Third order interpolation is effected by the use of the Lagrange interpolation formula for four points. The four points are found by table lookup and, where the table permits, these points are chosen such that two points lie on either side of the interpolated value. In the case of two way interpolation, 16 tabular values are selected such that the interpolated point (x,y) is as close to the center of the array pictured below as the table permits:



Four interpolations are made in the x direction in order to find $F(x,y_0)$, $F(x,y_1)$, $F(x\ y_2)$, $F(x,y_3)$. Using these four points, a final interpolation is made in the y direction to find $F(x,y)$.

PX 71900-8-122

The four point Lagrange formula,

$$F(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}F(x_0) + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}F(x_1)$$

$$+ \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}F(x_2) + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}F(x_3),$$

is used for two reasons:

1. Uniform interval of tabulation is not necessary-thus, where the function behaves badly the points of tabulation may be taken close together, and conversely, where the function is smooth fewer points need be tabulated.
2. Four points appear to be reasonable compromise between speed of computation and accuracy.

This subroutine requires the following information:

1. Table of function values: $F(x)$ or $F(x,y)$.
2. Table of values of x.
3. Table of values of y (for two way interpolation).

If these numbers are stored on MD it is suggested that they be stored in the following sequence in order to minimize access time.

All y values, all x values, all $F(x,y)$ values.

y values and x values must be stored in <u>ascending</u> order: $x_0, x_1, \ldots x_n$.

$y_0, y_1, \ldots y_n$. F(x,y) stored as follows:

$F(x_0, y_0),\ F(x_1, y_0), \ldots, F(x_n, y_0), F(x_0, y_1), \ldots, F(x_0, y_n), \ldots, F(x_n, y_n)$

If interpolation outside the tabular values (extrapolation) is attempted, this subroutine prints out "xtrap" and exits to the alarm routine with y in (L) and x in (R).

In case of a divide fault, set (P A K) = 0 and start: The subroutine prints out "oflo" and exits to the alarm routine with y in (L) and x in (R). A divide fault will occur if the tabular values of F(x,y) are scaled too large with respect to ratios of differences in the independent variable.

One can estimate an optimum scaling for F(x) by taking into account the order of computation of intermediate results:

$$\left(\frac{\left(\frac{\left(\frac{F(x)\,\Delta x_1}{\Delta x_2}\right)\Delta x_3}{\Delta x_4}\right)\Delta x_5}{\Delta x_6}\right) + \cdots \text{ETC.}$$

If F represents the maximum value of F(x) in some region of four consecutive tabulated points, and $\frac{\Delta x_i}{\Delta x_j}$ represents the maximum ratio of differences in the independent variable in this region, then $F2^s \frac{\Delta x_i}{\Delta x_j} < 2^{35}$ should be a reasonable estimation of the condition for no overflow. The value of s (scale factor of F(x)) is then estimated by the following:

$$s < 35 - \log_2\left(F \frac{\Delta x_i}{\Delta x_j}\right)$$

Rescaling of F(x) and/or retabulation or points would be necessary to eliminate any overflow that does occur.

The subroutine is entered by means of a return jump command which preceeds two parameter words. The values of x and y are placed in (A) and (Q), respectively, before entry. The subroutine exits to the command following the second parameter word. F(x,y) is in (A) upon exit.

1. One way interpolation

Inital: x in (A)

Execute:	c)	37	01001	01002
	c +1)	00	xxxxx	fffff
	c +2)	0N	NN000	00000
	c +3)	Next instruction		

Final: F(x) in (A)

2. Two way interpolation

Inital: x in (A)

y in (Q)

Execute:	c)	37	01000	01001
	c +1)	00	xxxxx	fffff
	c +2)	0N	NNnnn	yyyyy
	c +3)	Next instruction		

Final: F(x,y) in (A)

PARAMETERS

xxxxx address of first x value

fffff address of first function value

NNN number of x values

nnn number of y values

yyyyy address of first y value

This subroutine does not use the constant pool or temporary pool.

Subroutine length--$(207)_8 = (135)_{10}$
(Including constants)

Temporaries required-----------------------------------$(35)_8 = (29)_{10}$

Total working space------------------------------------$(244)_8 = (164)_{10}$

Number of words for assembly modification--------------$(173)_8 = (123)_{10}$

PX -71900-8-122

DETERMINANT EVALUATION PACKAGE-REAL-CA 010

This is a package including Floating Point Real Arithmetic, CA 001 an interpretive routine for the real arithmetic and the real determinant evaluation routine. It is coded in standard form and can be assembly modified. If this package is assembled at 0100, it will evaluate an $N \times N$ determinant for $N = (22)_8 = (18)_{10}$. Each element of the determinant takes two (2) cells, and the entire determinant must be in ES. This routine requires a setting up for particular N, location of determinant in ES, and the location of the result in ES, and can not be used for any other N, determinant location, or result location unless the routine is restored to its original form and set up again. Any number of determinants may be evaluated for a given set-up. The determinant is destroyed during the evaluation. The elimination method is used.

Steps of Elimination:

1. $1 \rightarrow$ (Det. value location).
2. $\frac{a_{1j}}{a_{11}}$ $j = 2, 3, \ldots, N$ (if $a_{11} = 0$ exchange row 1 with a row which has a non-zero first element, and change the sign of (Det. value location).
3. $A_{ij} - A_{i1} \frac{A_{1j}}{A_{11}}$ $j, i = 2, 3, \ldots, N$.
4. (Det. value location) $A_{11} \rightarrow$ (Det. value location)
5. Reduce the order of the resulting matrix by one by removing from consideration row one, and col. 1.
6. Repeat steps 2-5 until order of the matrix is reduced to one.

This routine uses 2 N cells (one row) immediately following the determinant as temporary storage. After it is set up, the last $(46)_{10} = (56)_8$ cells of the routine are no longer used (from 1340-1416). The amount of storage needed for the determinant is given by $2(N^2+N)$. The elements of the matrix must be stored by rows.

The real arithmetic, or the real arithmetic and interpretive routine may be used independent of the determinant routine by choosing the proper entrances.

For instructions on the use of real arithmetic see CA 001. The interpretive routine performs the real arithmetic operations referring to a parameter word for the locations of the mantissas of the operands and result.

REAL ARITHMETIC ENTRANCE

Co 37 01001 01002 Add Ent

Co 37 01001 01003 Mult Ent

Co 37 01001 01004 Div Ent

INTERPRETIVE ROUTINE ENTRANCE

Co 37 01001 PPPPP

C_1 AAAA BBBB CCCC

Where operations performed are:

Add (A)+(B)⟶(C) P=1106

Subt (A)-(B)⟶(C) P=1110

Mult (A)X(B)⟶(C) P=1112

Div (A)÷(B)⟶(C) P=1114

Where AAAA-location of mantissa of 1st operand

BBBB-location of mantissa of 2nd operand

CCCC-location of mantissa of result

DETERMINANT SET-UP ENTRANCE

Co 37 01001 01153

C_1 DDDD RRRR NNNN

Where DDDD-location of Det. in ES

RRRR-location of result in ES

NNNN order of the determinant (octal)

DETERMINANT EVALUATION ENTRANCE

Co 37 01001 01153

Permanent const. used 40, 74, 77, 43, 66, 44.

Temp. storage 00003-00032, (A), (Q)

Commands for assembly modification 407_8 $(263)_{10}$

Drum Address-70723-71340

By: J. N. Ellis
Checked by: Donn B. Parker

CV -123

70723	01000	37	76000	76002	ALARM EXIT
70724	01001	45	00000	30000	EXIT
70725	01002	45	00000	01027	ADD ENTRANCE
70726	01003	45	00000	01073	MULTIPLY ENTRANCE
70727	01004	11	00027	20000	DIVIDE ENTRANCE M (OP 2) → (A)
70730	01005	47	01006	01000	DIVIDING BY 0?
70731	01006	11	00025	20000	NO: NUMERATOR → (A)
70732	01007	47	01014	01010	NUMERATOR = 0?
70733	01010	11	00040	00031	YES: 0
70734	01011	11	00040	00032	→ ANSWER
70735	01012	37	01012	01013	S.R. EXIT
70736	01013	45	00000	01001	JUMP TO EXIT
70737	01014	54	20000	00042	NO: M (NUMERATOR) X 2^{34} → (A)
70740	01015	73	00027	10000	QUOTIENT X 2^{34} → (Q)
70741	01016	11	00040	00025	0 → 25
70742	01017	74	10000	00025	NORMALIZE IF (00025) = 0 OR 71
70743	01020	11	20000	00031	NORMALIZED MANTISSA STORED
70744	01021	23	00026	00030	DIFF OF EXP. → (00026)
70745	01022	11	00025	20000	H → A
70746	01023	47	01025	01024	H = 0?
70747	01024	21	00026	00074	H = 0 CORRECTED
70750	01025	11	00026	00032	H = 71 EXP → 00032
70751	01026	45	00000	01012	JUMP TO EXIT
70752	01027	11	00025	20000	ADDITION M (OPERAND 1) → (A)
70753	01030	47	01031	01051	M (OPERAND 1) = 0?
70754	01031	11	00027	20000	NO: M (OPERAND 2) → (A)
70755	01032	47	01036	01033	M (OPERAND 2) = 0?
70756	01033	11	00025	00031	YES: M (OPERAND 1) → M (ANSWER)
70757	01034	11	00026	00032	E (OPERAND 1) → E (ANSWER)
70760	01035	45	00000	01012	JUMP TO EXIT
70761	01036	11	00026	20000	NO: E (OPERAND 1) → (A)

PX 71900-8-123

By: J. N. Ellis
Checked by: D. B. Parker

CV -123

```
70762  01037  36 00030 00032   E(OP 1)  MINUS E(OP2) = K -> (00032)
70763  01040  46 01047 01041   K > 0?
70764  01041  11 00025 20000   YES:  EXCHANGE
70765  01042  11 00027 00025         OPERAND
70766  01043  11 20000 00027          ONE
70767  01044  11 00026 20000           AND
70770  01045  11 00030 00260
70771  01046  11 20000 00030              TWO
70772  01047  12 00032 20000   NO:  |K| -> (A)
70773  01050  42 01072 01054   |K| < 35?
70774  01051  11 00027 00031   |K| >= 35 OR M(OP) 1 = 0
70775  01052  11 00030 00032        (OP 2) -> ANSWER
70776  01053  45 00000 01012   JUMP TO EXIT
70777  01054  16 20000 01055   |K| < 35?  SET SHIFT OF 5 BITS
71000  01055  54 00027 30000   M (OP2) X 2^|K| -> (A)
71001  01056  35 00025 20000   M(OP2) X 2^K + M(OP1) -> (A)
71002  01057  11 00040 00032   0 -> (00032)
71003  01060  74 20000 00032   NORMALIZE M (ANS) = H -> (00032)
71004  01061  11 20000 00031   M (ANS) -> (00031)
71005  01062  47 01064 01063   M (ANS) = 0?
71006  01063  13 00032 00026   YES:  -H -> (00026)
71007  01064  11 00032 20000   NO:    H -> (A)
71010  01065  42 01071 01067   H < 38?
71011  01066  23 00026 00077   NO: E (OP1)-32 -> (00026)
71012  01067  21 00032 00026   YES: E (OP1) PLUS H  (OR H-72)
71013  01070  45 00000 01012   JUMP TO EXIT
71014  01071  00 00000 00*46   DEC 38
71015  01072  00 00000 00*43   DEC 35
71016  01073  71 00025 00027   MULT M (OP1) X M (OP 2) -> (A)
71017  01074  11 00040 00025   0 -> (00025)
```

PX 71900-8-123

FORM NO. E. T. - I 35 F

CONVAIR — DIVISION OF GENERAL DYNAMICS CORP.
SAN DIEGO, CALIFORNIA

By: J. N. Ellis
Checked by: D. B. Parker

CV-123

71020	01075	74	20000	00025	NORMALIZE M(ANS)=34 OR 35 → H
71021	01076	11	20000	00031	M(ANS) → (00031)
71022	01077	47	01102	01100	M(ANS) = 0?
71023	01100	11	00040	00032	YES: 0 = E(ANS)
71024	01101	45	00000	01012	JUMP TO EXIT
71025	01102	23	00025	01072	NO: H-35 → (00025) = 0 OR -1
71026	01103	21	00026	00030	E(OP1) PLUS E(OP 2) → (A) → (00026)
71027	01104	35	00025	00032	E (ANS) → (00032)
71030	01105	45	00000	01012	JUMP TO EXIT
71031	01106	16	01147	01134	ADD
71032	01107	45	00000	01115	ENTRANCE
71033	01110	16	01151	01134	SUBTRACT
71034	01111	45	00000	01115	ENTRANCE
71035	01112	16	01150	01134	MULTIPLY
71036	01113	45	00000	01115	ENTRANCE
71037	01114	16	01146	01134	DIVIDE ENTRANCE
71040	01115	31	01001	00017	SET
71041	01116	15	20000	01117	PARAMETER
71042	01117	11	00000	01152	LOCATION
71043	01120	55	01144	10003	SET
71044	01121	53	01152	01136	L (Z)
71045	01122	55	01152	00033	SET
71046	01123	55	01145	10003	L (X)
71047	01124	53	01152	01131	
71050	01125	54	01152	00014	SET
71051	01126	53	01152	01133	L(Y)
71052	01127	21	01001	00074	UP EXIT LOC.
71053	01130	75	30002	01132	X →
71054	01131	11	00000	00025	(OP 1)
71055	01132	75	30002	01134	Y →

PX 71900-8-123

CONVAIR — DIVISION OF GENERAL DYNAMICS CORP.
SAN DIEGO, CALIFORNIA

FORM NO. E.T. 1 9 F

By: J. N. Ellis
Checked by: D. B. Parker

PAGE CA-010-7
REPORT ZM 491
MODEL
DATE 2-27-56

CV-123

71056	01133	11	00000	00027	(OP 2)
71057	01134	37	01012	00000	REAL ARITH. JUMP
71060	01135	75	30002	01137	ANSWER
71061	01136	11	00031	00000	→ (Z)
71062	01137	37	01137	01140	S.R. EXIT
71063	01140	45	00000	01001	JUMP TO EXIT
71064	01141	13	00027	00027	- (00027) → (00027)
71065	01142	37	01012	01002	JUMP TO SUB
71066	01143	45	00000	01135	JUMP TO STORE (ANS) + EXIT
71067	01144	70	00000	00777	MASK
71070	01145	00	00777	70000	MASK
71071	01146	00	00000	01004	
71072	01147	00	00000	01002	
71073	01150	00	00000	01003	
71074	01151	00	00000	01141	
71075	01152	00	00000		
71076	01153	45	00000	01303	SET UP S. R. ENTRY
71077	01154	15	01405	01115	SET PARAMETER LOC
71100	01155	15	01323	01174	SET L (A11) X 2^{15} → (01174)
71101	01156	15	01323	01177	SET L (A111)·2^{15} → (01177)
71102	01157	11	01322	01225	SET L (A12 A11 A12)
71103	01160	11	01306	01242	SET L (A12 A21 T2)
71104	01161	11	01307	01244	SET L (A22 T2 A22)
71105	01162	11	01321	01255	SET L (T2 A11 A22)
71106	01163	11	00066	00003	1 →
71107	01164	11	00074	00004	(T1)
71110	01165	11	01302	01337	STORE $2 \times 2^{24} + 4$
71111	01166	11	01314	01336	STORE $2(N-1) \cdot 2^{24}$
71112	01167	75	30006	01171	STORE COPYS
71113	01170	11	01302	01325	2N X 2^{15}

By: J. N. Ellis
Checked by: D. B. Parker

CV-123

71114	01171	21	01325	01311	N-1
71115	01172	11	01305	01333	INDEX = N-2
71116	01173	11	01327	01334	INDEX = N-1
71117	01174	11	00000	20000	(A11) → A
71120	01175	47	01217	01166	A = 0?
71121	01176	75	30000	01200	YES: ROW 1
71122	01177	11	00000	00000	→ (TEMP ROW)
71123	01200	15	01177	01205	STORE
71124	01201	21	01205	01326	L (ROW 2)
71125	01202	55	01177	20025	STORE
71126	01203	16	10000	01205	L (ROW1)
71127	01204	75	30000	01206	ROW 2
71130	01205	11	00000	00000	→ ROW 1
71131	01206	55	01205	20025	STORE
71132	01207	16	10000	01211	L (ROW 2)
71133	01210	75	30000	01212	ROW 1 (TEMP ROW)
71134	01211	11	00000	00000	→ ROW 2
71135	01212	13	00003	00003	-|T1| → (T1)
71136	01213	21	01326	01303	$2NX2^{15} + 2NX2^{0}$
71137	01214	41	01334	01174	TEST NEW A11
71140	01215	75	10002	01277	COL 1 = 0
71141	01216	11	00040	00003	0 → |DET| JUMP TO EXIT
71142	01217	11	01303	01326	COL1 = 0 RESTORE CONSTANT
71143	01220	23	01327	00074	INDEX - 1
71144	01221	21	01177	01303	SET FOR
71145	01222	21	01174	01312	NEXT ROW
71146	01223	11	01330	01334	SET INDEX = N-1
71147	01224	37	01137	01114	
71150	01225	00	00000		A12/A11 → (A)
71151	01226	21	01225	01302	SET FOR A13

PX 71900-8-123

FORM NO E.T.1 ...

CONVAIR — DIVISION OF GENERAL DYNAMICS CORP.
SAN DIEGO, CALIFORNIA

By: J. N. Ellis
Checked by: D. B. Parker

C_v-123

71152	01227	41	01334	01224	R/A11
71153	01230	11	01330	01334	SET INDEX=N -2
71154	01231	11	01330	01335	SET INDEX=N-2
71155	01232	31	01242	00003	SET
71156	01233	11	01324	10000	L (A21)
71157	01234	53	20000	01235	
71160	01235	11	00000	20000	(A21)→(A)
71161	01236	47	01241	01237	(A21) = 0?
71162	01237	21	01244	01317	YES; SET FOR NEXT ROW
71163	01240	45	00000	01252	JUMP TO SKIP ROW
71164	01241	37	01137	01112	NO: X
71165	01242	00	00000	00 *	A12 X A21→ T
71166	01243	37	01137	01110	
71167	01244	00	00000	00 *	A22-(A22 A21)→ A22
71170	01245	21	01242	01301	SET FOR A13
71171	01246	21	01244	01302	SET FOR A23
71172	01247	41	01335	01241	0 → 1 COL
71173	01250	23	01242	01336	
71174	01251	21	01244	01337	
71175	01252	21	01242	01310	
71176	01253	41	01334	01231	1ST COL 1, 0,0
71177	01254	37	01137	01112	X
71200	01255	00	00000	00000	(T) (A11)→ T
71201	01256	23	01330	00074	INDEX -1
71202	01257	23	01336	01301	$4(N+1)\times 2^{24} + 4 \times 2^{12}$
71203	01260	21	01337	01302	$4 \cdot 2^{24}$ $4 \cdot 2^{12} + 4$
71204	01261	21	01255	01311	SET FOR A22
71205	01262	21	01331	01315	SET
71206	01263	11	01331	01242	PARAMETERS
71207	01264	21	01332	01320	

By: J. N. ELLIS
Checked by: D. B. Parker

CV-123

71210	01265	11	01332	01244	
71211	01266	21	01325	01302	
71212	01267	21	01225	01325	
71213	01270	41	01333	01173	DET REDUCED ?
71214	01271	11	01255	01273	YES:
71215	01272	37	01137	01112	X
71216	01273	00	00000		T → |DET|
71217	01274	15	01406	01115	RESET INT. S.R.
71220	01275	15	01406	01127	
71221	01276	75	30002	01001	STORE
71222	01277	11	00003	00000	|DET| & EXIT
71223	01300	00	00000	20000	
71224	01301	00	02000	00000	
71225	01302	00	02000	00002	
71226	01303	31	01001	00017	S R SET UP ENT
71227	01304	15	20000	01305	SET (L) PARAMETER
71230	01305	11	00000	01416	PARAMETER → (T)
71231	01306	21	01001	00074	EXIT PLUS 1
71232	01307	11	01410	10000	MASK → (O)
71233	01310	51	01416	01153	STORE N
71234	01311	11	01153	01304	STORE
71235	01312	23	01304	00074	N-1
71236	01313	36	00074	01305	STORE N-2
71237	01314	71	01300	01153	STORE
71240	01315	11	20000	01310	$2N \cdot 2^{12}$
71241	01316	55	01310	10003	STORE
71242	01317	11	10000	01303	$2N \cdot 2^{15}$
71243	01320	35	01300	01311	STORE $2(N+1) \cdot 2^{12}$
71244	01321	54	20000	00003	STORE
71245	01322	11	20000	01312	$2(N+1) \cdot 2^{15}$

PX 71900-8-123

By: J. N. Ellis
Checked by: D. B. Parker

CV-123

71246	01323	11	01153	01313	STORE
71247	01324	21	01313	00074	N + 1
71250	01325	71	01301	01153	STORE
71251	01326	11	20000	01314	$2N \times 2^{24}$
71252	01327	15	01405	01127	SET S R
71253	01330	35	01301	01315	STORE $2\ (N+1)\cdot 2^{24}$
71254	01331	11	01311	01316	STORE
71255	01332	21	01316	01302	$2 \times 2^{24} + 2(N+1)\cdot 2^{12} + 2$
71256	01333	21	01315	01311	STORE $2(N+1)\,2^{24} + 2(N+1)\cdot 2^{12}$
71257	01334	71	00041	01153	STORE
71260	01335	35	01314	01317	$2N \times 2^{24} + 2N$
71261	01336	35	01302	01320	STORE $2(N+1)\,2^{24} + 2(N+1)$
71262	01337	23	01314	01301	STORE $2(N-1)\cdot 2^{24}$
71263	01340	11	01412	10000	MASK → (Q)
71264	01341	11	01414	01306	STORE L (T2)
71265	01342	11	01414	01307	STORE L (T2)
71266	01343	53	01416	01306	STORE
71267	01344	53	01416	01307	A11
71270	01345	55	01416	00030	
71271	01346	11	01411	10000	MASK → Q
71272	01347	53	01416	01306	STORE L (A11) $\times 2^{12}$
71273	01350	11	01415	01321	STORE L $(T1)\cdot 2^{24}$ + L (T1)
71274	01351	53	01416	01321	STORE L $(A11)\cdot 2^{12}$
71275	01352	11	01410	10000	MASK → (Q)
71276	01353	53	01416	01277	STORE L (|DET|)
71277	01354	55	01416	00030	
71300	01355	11	01410	10000	MASK → (Q)
71301	01356	11	01306	01322	
71302	01357	53	01416	01322	STORE
71303	01360	53	01416	01307	L (A11)

71304	01361	55	01416	00017	
71305	01362	11	01407	10000	MASK ⟶ (Q)
71306	01363	51	01416	01323	STORE L (A11) · 2^{15}
71307	01364	21	01322	01302	STORE L (A12 A11 A12)
71310	01365	21	01306	01310	
71311	01366	21	01306	01301	STORE L (A12 A21 T2)
71312	01367	21	01307	01320	STORE L (A22 T2 A22)
71313	01370	11	01413	10000	MASK ⟶ (Q)
71314	01371	53	01303	01176	SET
71315	01372	53	01303	01204	REPEAT
71316	01373	53	01303	01210	COMMANDS
71317	01374	15	01323	01211	STORE L (A11)
71320	01375	71	01303	01153	$2N^2$ ⟶ A
71321	01376	35	01211	01211	STORE L (T ROW) · 2^{15}
71322	01377	55	01211	20025	
71323	01400	16	10000	01177	
71324	01401	11	01404	01153	SET ENT JUMP
71325	01402	11	01407	01324	STORE MASK
71326	01403	45	00000	01001	JUMP TO EXIT
71327	01404	45	00000	01154	
71330	01405	00	01137	00000	
71331	01406	00	01001	00000	
71332	01407	00	07777	00000	
71333	01410	00	00000	07777	
71334	01411	00	00777	70000	
71335	01412	77	77000	00000	
71336	01413	00	00777	00000	
71337	01414	00	00000	70007	
71340	01415	00	03000	00003	

P_x 71900-8-123

Notes on Timing of the Controlled Reproducer

The controlled Reproducer punches or reads cards at a maximum rate of 120 cards per minute or 500 milliseconds per card. When the Reproducer is started, an electromagnetic clutch is energized and several shafts are turned through one revolution. Each revolution operates cams and mechanisms which execute an 18 point sequence of operations called the "card cycle". During the first five points of a cycle a card is moved through the channel and positioned for reading or punching. The next 12 points process 12 rows of card information. Following the last row there is a final point before starting the next card cycle.

The theoretical execution time for one point of this 18 point cycle is 27.8 or $\frac{500}{18}$ milliseconds. Because of the mechanical nature of the equipment the theoretical time of 27.8 m.s. may not be realized for every point of a 500 ms. cycle.

A study of the computing times actually available during various portions of a card cycle has been made. Tests were run on four Reproducers. The time available for computation was measured by determining the number of times an Index Jump instruction could be performed without an IO fault occuring. Each of the Reproducers was adjusted to pass cards at the rate of 120/min.

<u>Theoretical Computation Times</u> Consider a sequence for reading and punching consecutive cards:

EF - V	Select Reproducer	
Other instructions	(interval C)	
EW - OV	Write Field III	Row 9
EW - IV	Write Field I	
EW - IV	Write Field II	
ER - OV	Read Field III	
ER - IV	Read Field I	
ER - IV	Read Field II	
Other instructions	(interval A)	
————————		Row 8
- - - - - - - - - -		
————————		Row 12
Other instructions	(interval B)	
EF - V or EW, Field III	Select Reproducer for next card if "single card" mode or process Row 9 if "free run" mode	

Three intervals are available for computation:

(A) The time between card rows (or clutch selection)
(B) The time from row 12 to the next External Function Instruction
(C) The time from the External Function instruction to row 9 (or clutch selection)

The numbers used to designate the 18 points of the card cycle are not sequential. At the start of a cycle when the cards are at rest the Reproducer is at point 14. Points 14 through 18 occur as the card moves through the channel to the next station. During each of the next 12 points a row of information is processed; these points are numbered, corresponding to the card rows, 9, 8, 7, 6, 5, 4, 3, 2, 0, 11, 12. The final point of the cycle is 13. See Figure 1 for a diagram of the card cycle.

(A) At the beginning of each of the 12 cycle points during which row information is processed, a row ENABLE occurs. This activates a timing device. In the case of punching, if information is not loaded into IOB within 15 ms. a NO INFORMATION fault results. In the case of reading, if information is not read out of IOB within 10 ms. a NO INFORMATION fault results.

The NO INFORMATION fault will result only from a failure to process information for Field I in time. There is no fault to indicate a failure to process information for either Field III or Field II. Hence all External Write and External Read instructions for a given row should be programmed consecutively. (There is in fact a period of about 16 ms. following a cycle point during which information may be transferred between the computer and the Reproducer; thus there is time to process Field II following Field I.)

The time from one ROW ENABLE to the next is theoretically 27.8 ms. Thus for programs which execute a punch and read, a punch, or a "normal read", there is theoretically available between rows about 27.6 ms. By a "normal read" is meant an External Read executed at the beginning of a cycle point. It is also possible to execute a "delayed read". After information has been loaded in the IOA or IOB registers it is possible to delay reading it out for 10 ms. after the cycle point. It is thus possible to program around 36 ms. of computation between reading two successive rows. However, the interval available for computation following such a delayed read is at most 27.6. Should the interval following a delayed read be shortened to 17.6 ms. it is then possible to execute another delayed read 36 ms. later. See Figure 2. In performing a punch and read operation it is mandatory that the External Reads follow immediately the External Writes since there is no NO INFORMATION fault protection on other than the External Write, Field I. Thus a delayed read should be used only in a reading operation and not in a reading and punching operation. If a delayed read is executed on row 12 the interval from row 12 to the clutch selection (EF instruction) must be decreased by 10 ms; if a delayed read is used on row 9 the interval from the clutch selection to row 9 may be increased by 10 ms.

(B) The time from the execution of the External Function instruction which selects the Controlled Reproducer to the clutch latch up is known as the clutch access time. Since there are two points at which the clutch may be latched, the access time may vary from 0 to 250 ms.* In case consecutive card cycles are programmed, if the clutch selection is made within 17.6 ms. of the beginning of point 12 there will be no delay in latching the clutch. Otherwise the clutch may not latch up for as much as a half revolution or 250 ms. If the Reproducer is programmed in the "single card" mode, one may of course include any amount of computation between row 12 and the External Function instruction for the next card cycle. However, if more than 10 ms. of computation takes place a delay of from 0 to 250 ms. will be incurred in latching the clutch.

(C) The time from the clutch selection to point 9 would appear to be 177 milliseconds or 10 ms. of point 12 plus 6 X 27.8 ms. for points 13 through 18. This interval is in fact longer because of a small amount of time required for the clutch to actually latch up.

Actual Computation Times Because of the complex mechanical nature of the Controlled Reproducer, it is to be expected that there will be considerable variation from the theoretical times stated above. Consideration of the tabulated results of the timing trials has led to the following recommendations:
(A) that no more than 24 ms. be programmed between card rows;
(B) that no more than 170 ms. be programmed between the External Function instruction and the first External Read or External Write for row 9; and
(C) that if it is desired not to lose time while the clutch is latched up for the next cycle no more than 10 ms. be programmed between row 12 and the External Function instruction for the following card cycle.

A program whose timing is within these limits is guaranteed to run.

Single Field Reading and Writing It is possible, but not advisable, to execute a single External Read and/or a single External Write to process one field of information from a card row. In the case of reading, Field I only may be read provided nothing is punched in Field II. With the Reproducer set for two-field operation, the first External Read will read from IOB information punched in Field I of row 9. The information from Field II will then be placed in IOB. If this information is not read out during the row 9 point, when row 8 is enabled the information from Field I of row 8 will be placed in IOB. The contents of IOB are now (Field II, row 9) $\oplus$ (Field I, row 8) which equals (Field I, row 8) provided nothing is punched in Field II. In the case of writing, execution of a single External Write per row will punch in Field I the information loaded in IOB and punch in Field II the zeros resulting from clearing IOB. To program the Reproducer for either reading or writing with a single External Read or External Write instruction per row is exceedingly dangerous. For if the interval between External Read or External Write instructions is not sufficiently long the information for a row may be punched in Field II of the preceeding row or read from Field II of the preceding row. Programmers should always code at least two External Read or External Write instruction per row and dump irrelevant information into a one register garbage pit. It is well worth squandering the small additional storage to retain the fault protection.

FX 71900-8 App. A

*Some Reproducers have a clutch which may be latched at six points; in this case the access time is 0 to 84 ms.

<u>Programming Rules</u> The following rules are set forth to define normal programming of the Controlled Reproducer. Although methods of abnormally programming the Reproducer have been mentioned, programmers are urged not to code in these fashions. It must be emphasized that violation of any of the rules is dangerous. At the very least a programmer must thoroughly understand the operation of the Reproducer. And it must not be assumed that the operation is adequately described in these notes.

A complete discussion of normal programming of the Controlled Reproducer is included in the publications (A) The ERA 1103 Controlled Reproducer, PX71778'A and (B) The ERA 1103 Computer System, Section 6: Programming, PX 71209. The following list of rules for programming the Reproducer is not complete. It is designed to indicate the instances in which the coding is most likely to be faulty.

Some Rules for Programming the Controlled Reproducer

(1) All External Read and External Write instructions processing information for a given row should be consecutive. The External Write instructions should precede the External Read instructions.

(2) At most 24 milliseconds of computation should be programmed between card rows. If the Reproducer is FREE RUNNING at most 180 milliseconds should be programmed between cards. At most 170 milliseconds should be programmed between the External Function instruction which selects the Reproducer and the External Read and/or External Write instructions for row 9.

(3) Either 3 or 2 External Read and/or External Write instructions should be programmed per card row, depending on whether Field III is being used or not.

PX 71900-8 App. A

CARD CYCLE 500 M-Secs.

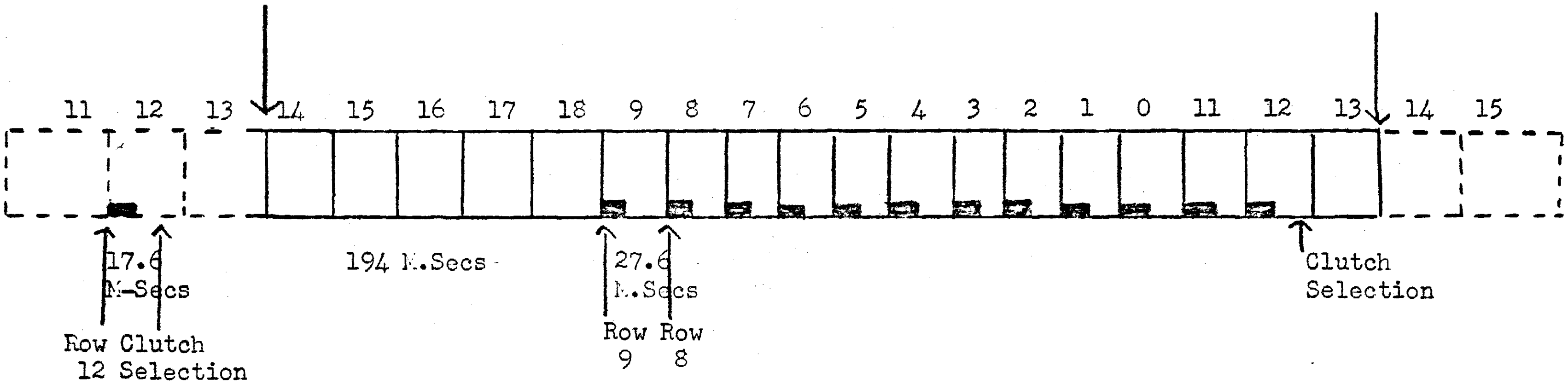
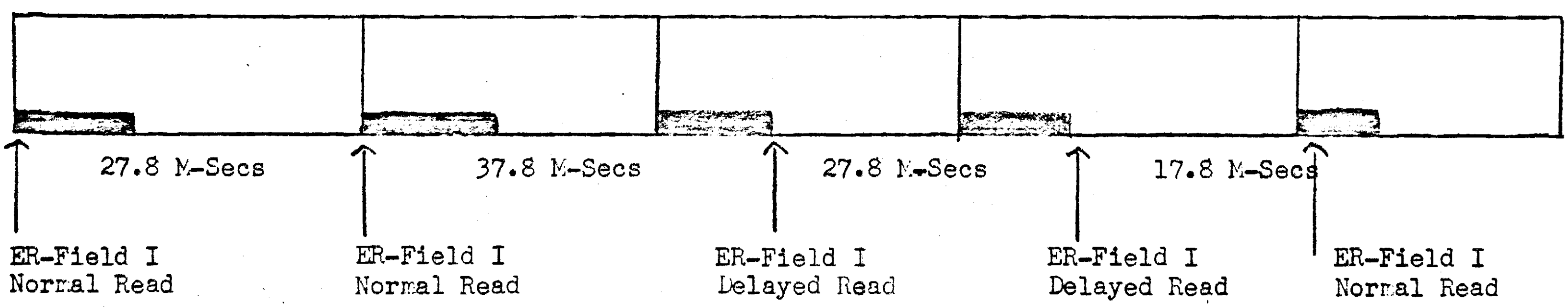


Figure 1

8:21

DELAYED READ



8-263

Figure 2

Formation of USE - a Cooperative Organization of 1103A Users

16 February 1956

In December of 1955 the desire was expressed by several 1103A purchasers to form a cooperative organization of 1103A users. Accordingly, a meeting was held at the Ramo-Wooldridge Corporation on December 19 and 20 to form such an organization. Attending were representatives of Boeing Airplane Company, Holloman Air Force Base, Lockheed Missile Systems Division, Ramo-Wooldridge Corporation and Remington Rand Univac Division.

The name USE - Univac Scientific Exchange - was selected for the organization. A number of objectives for the group were listed.

1. Exchange of programming techniques and ideas.
2. Exchange of programs and subroutines.
3. Exchange of information on computing organizations, operating procedures, etc.
4. Adoption of a common programming language for exchanged programs.
5. Adoption of a standard format for program write-ups.
6. Adoption of standard subroutine conventions.
7. Setting up of a cooperative manpower effort.
8. Cooperation at the program planning stage.
9. Achievement of a uniform general purpose system for the operation of all 1103A's.

It was pointed out that Remington Rand is continuing the Central Exchange for 1103 and 1103A information. However, material in the Central Exchange is unsolicited and unedited. The philosophy has been to require no special language or format for Central Exchange material; this makes it easy to contribute material and to distribute it quickly.

Membership in USE is open to any organization which is renting or has purchased or has a firm order for one or more Model 1103A computers. USE publications will be available to 1103 and 1103A users only. These publications will be distributed to all 1103 and 1103 A installations.

A structure of working committees was established. On January 9 and 10 the committees met as guests of the Boeing Airplane Company in Seattle, Washington.

Specifications for a common language for the exchange of library programs were discussed. It was emphasized that a particular installation would in no sense be bound to use this common language internally. The common language is designed to be sufficiently general to include most other languages. A minimum assembly program for translating common-language routines to octal programs was described. Specifications were also proposed for subroutine format and standard program write-ups.

Plans were made for immediate cooperation in achieving routines for the 1103A. Investigations of existing routines were initiated to determine the value of adapting such routines to the 1103A. In particular, common function routines from the Central Exchange and general matrix routines were to be surveyed. Assignments were made for the framing of specifications for minimum and ultimate data input and output routines.

Discussions of a common compiling routine were begun. The goals of such a compiler were listed as:

- Translation: Symbolic to octal
- Subroutine referencing
- Preparation for input and output formats
- Algebraic coding
- Storage assignment
- Automatic identification
- Scaling
- Automatic post-mortem and diagnosis

Arrangements were made for meetings in St. Paul on February 16 and 17 as guests of Remington Rand Univac Division.