# 32/75 System Architecture

## Course No. 340

## Student Workbook

July 1981

Publication Order Number: 359-000361-000

GOULD
Electronics

32/75 SYSTEM ARCHITECTURE
Course No. 340/2 Weeks

PREREQUISITES: A knowledge of electronic fundamentals, number systems, digital logic, and data processing techniques. Two years of maintaining software programmable equipment is required.

DESCRIPTION: Course No. 340 provides the student with the knowledge needed to efficiently operate, test, adjust, and troubleshoot the SYSTEMS 32/75 to the board level. The course familiarizes the student with the physical and functional features of all the units in the SYSTEMS 32/75. Topics discussed include: Computer organization and characteristics, system integration, physical layout and packaging, instruction set, diagnostics, and block diagram analysis of each system and function level module. During laboratory sessions, the student develops his troubleshooting skills by diagnosing simulated malfunctions placed in the equipment by the instructor.

## STUDENT INFORMATION

Welcome to the SYSTEMS Training Center. We hope your stay with us is both rewarding and pleasant.

## STUDENT LOUNGE

We have provided a Student Lounge for your use. You may make your choice of coffee, tea or hot chocolate, and donuts will be furnished each morning. There is no charge for any of these; however, if we are to have a constant supply of coffee, please make a new pot-full when you empty one. Please insure that only Brim is made in the pot marked Brim.

## CLASS HOURS

Our class hours are normally 8:30 A.M. to 4:30 P.M. with an hour for lunch, but in some cases classes will be required to meet in the evenings. Also, your instructor may slightly modify this schedule from time to time to allow proper flow of the course material. Break times will be at the instructor's discretion.

## KEEPING THE FACILITY CLEAN

We solicit your cooperation for helping us keep our area clean. This will not be a problem, even though we have a large number of people here, if each of you gives us a hand by tidying your work area when you leave it. Please do not take coffee into the lab.

## ASSISTANCE WITH PROBLEMS

If you run into a problem that your instructor cannot solve for you, please do not hesitate to seek the assistance of the administrative staff or the Training Manager.

## TELEPHONE CALLS AND CORRESPONDENCE

Your incoming telephone calls may be placed to 1-305-587-2900 X3042. The receptionist will take a message (if you are in class) and post it on the bulletin board in the student lounge. If you have a genuine emergency call, incoming or outgoing, we will handle it immediately. For outgoing emergency calls, contact the Training Manager or the administrative staff.

Any incoming mail may be addressed to:

SYSTEMS ENGINEERING LABORATORIES, INC.
6901 W. Sunrise Blvd.
Ft. Lauderdale, Fl. 33313
Attn: (Your Name) c/o Training Department

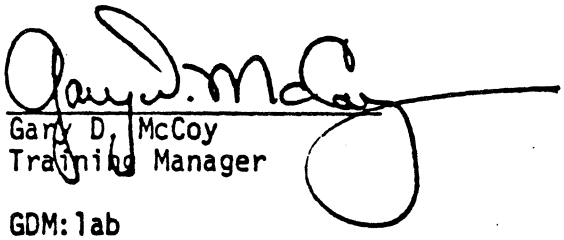
## STUDENT BADGES

While you are with us, you are asked to wear a student badge. We appreciate your cooperation. Your access is to the Training area only. Please do not forget to return the badge to your instructor before you leave.

## SYSTEMS TRAINING PHILOSOPHY

Finally a word about our training philosophy. Our courses are very laboratory intensive. If you expect to gain maximum benefit from the course, you must do the lab exercises. The instructor will be available to assist you in lab (but will not do the project for you). We like to conduct our classroom sessions to interact with the instructor and your classmates. This makes the sessions more interesting, and more learning effective.

Enjoy your stay!!

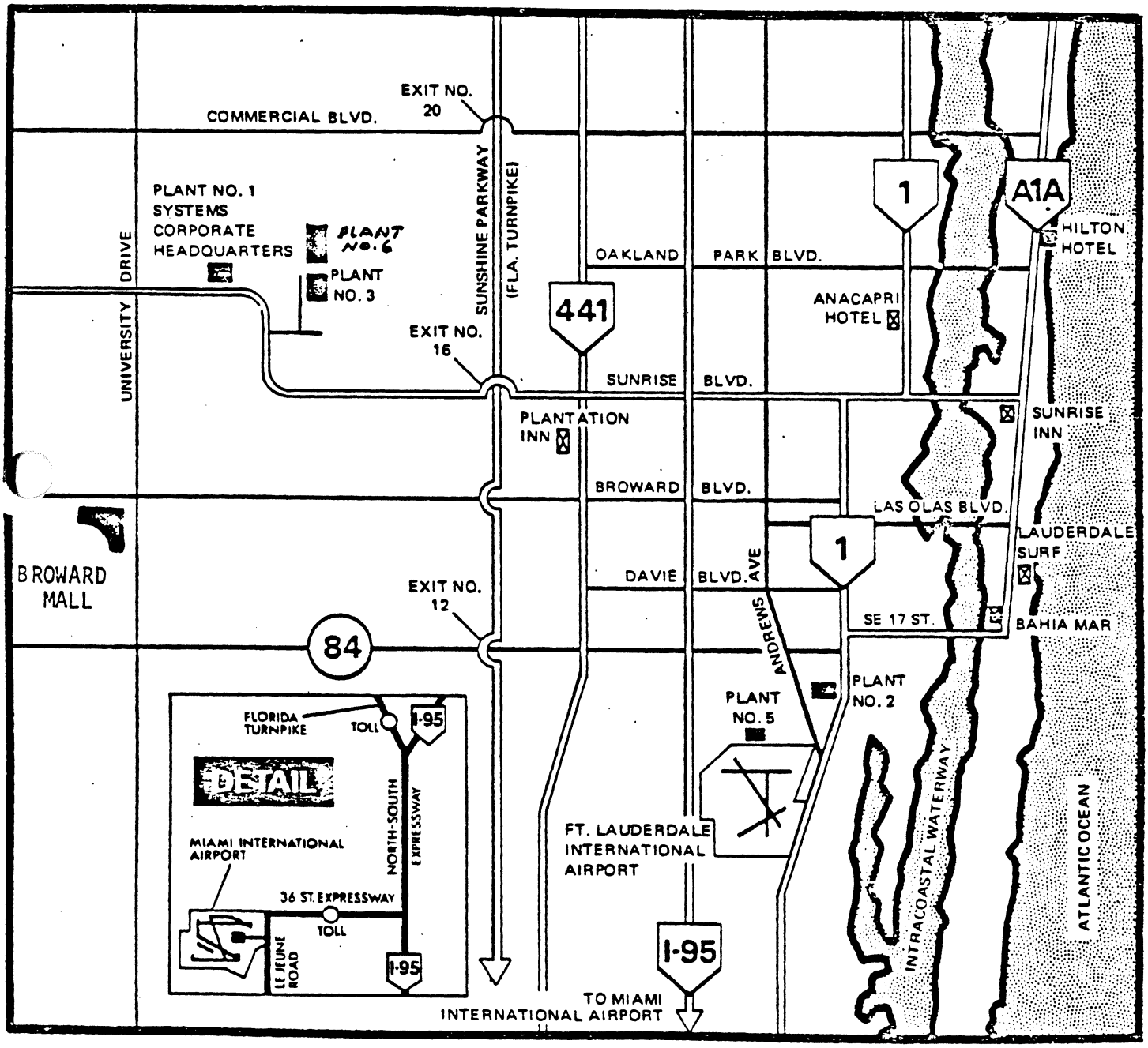


Gary D. McCoy
Training Manager

GDM:lab

# SYSTEMS facilities tour map

ENGINEERING LABORATORIES



SYSTEMS TRAINING DEPARTMENT

END OF COURSE CRITIQUE

STUDENT_____ COURSE_____

COMPANY _____ DATE_____

INSTRUCTOR(S) _____

PLEASE CIRCLE LETTER OF YOUR CHOICE

1. How much experience do you have in the computer industry?

   A. 0-1 Year
   B. 1-3 Years
   C. 3-5 Years
   D. Over 5 Years

2. How appropriate was the length of this course?

   A. Much too long, I could have easily learned the subject in 25% less time
   B. A little too long, I could have learned the subject in 10% less time
   C. Just right
   D. A little too short, I could have used about 10% more time
   E. Much too short, I could have used about 25% more time

3. What is your reaction to the amount of lab given in this course?

   A. Could have used much less lab time (20% less)
   B. Could have used less lab time (10% less)
   C. Lab time was sufficient
   D. Would have liked more lab time (10% more)
   E. Would have liked much more lab time (20% more)

4. The objectives for a unit of study were presented in such a way that:

   A. I always knew what I was expected to learn
   B. I usually knew what I was expected to learn
   C. I occasionally had an idea of what should be learned
   D. I seldom knew what I was expected to learn
   E. I never had an idea of what should be learned

5. The content of the course was presented in a sequence that was:

   A. Well organized and very easy to follow throughout
   B. Usually well organized and easy to follow
   C. Fairly organized; at times difficult to follow
   D. Disorganized and difficult to follow
   E. Disorganized and very difficult to follow

6. How would you rate the visual aids (such as illustrations in the student materials, slides, transparencies, etc.) in terms of helping you to learn the materials?

   A. Good quality and sufficient numbers of visual aids
   B. Fair quality and adequate numbers of visual aids
   C. Good quality but not enough visual aids
   D. Poor quality but sufficient numbers of visual aids
   E. Poor quality and not enough visual aids

7. How would you evaluate your ability to use the documentation (manuals, listings, etc.) in your work as a result of this course?

   Comments

   A. Excellent
   B. Good
   C. Fair
   D. Poor
   E. Not applicable

8. In regard to assistance you received during conference or lab, mark one of the following.

   A. I did not require assistance
   B. The assistance I received was excellent
   C. I could have used assistance about one to three times more each day
   D. I could have used assistance about four to six times more each day
   E. The quality of assistance I received was inadequate

9. When going into lab, I felt that the conference sessions had prepared me:

   A. Excellently
   B. Very well
   C. Marginally
   D. Poorly
   E. No lab in this course

10. The directions for lab projects were:

A. Always clear and easy to understand
B. Usually clear and easy to understand
C. Often unclear
D. Very unclear and required additional instructions to understand
E. No lab in this course

11. How well would you evaluate the subject matter expertise of the instructor?

Comments

A. Excellent
B. Good
C. Fair
D. Poor

12. How would you rate the instructor's ability to present the material?

Comments

A. Excellent
B. Good
C. Fair
D. Poor

13. How would you rate the instructor's ability to relate to the students?

A. Very helpful, very cooperative, very responsive
B. Cooperative, responsive
C. Cooperative but sometimes abrupt
D. Antagonistic and degrading to the student

14. What is your overall opinion of the training department, the staff and its facilities?

Comments

A. Excellent
B. Good
C. Fair
D. Poor

Please use the reverse side for any comment you wish to make. Some topics you may wish to address are: publications, instructor ability, additional courses, weak or strong points in this course, technical level of course, etc.

## Section II - Documentation

1. What is your opinion of SYSTEMS standard publications?

2. Do you like the format?

3. Are the manuals reasonably free of typographical errors?

4. Are they technically accurate (please be specific)?

5. What changes in the documentation would you suggest?

## Section III - Product Evaluation

1. What is your opinion of the products studied in this course?

2. What is your application of these products?

3. What would make these products more attractive to your application?

4. Do you think you are getting good product support from SYSTEMS (please be specific)?

5. Would you recommend that your company continue to use SYSTEMS products?

# COURSE SCHEDULE

| | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY |
|---|---|---|---|---|---|
| 1 | INTRO TO COURSE | INSTRUCTION SET | I/O W/O INTERRUPTS | I/O WITH INTERRUPTS | SCRATCHPAD |
| 2 | 1. CPU BLOCK DIAGRAM<br>2. SYSTEM BLOCK DIAGRAM | 1. OPERATING MODES<br>2. INTRODUCTION TO INSTRUCTION SET | 1. INTRODUCTION TO I/O (CLASS 0,1, 2,3,D,E,F)<br>2. I/O INSTRUCTIONS<br>A) CLASS 3<br>B) CLASS 0,1,2<br>C) CLASS E | 1. I/O INTERRUPT PHILOSOPHY<br>2. I/O PROGRAMMING WITH INTERRUPTS (CLASS 0,1,2,E) | 1. INTRO TO SCRATCHPAD<br>2. PSD INT/TRAP PROCESSING |
| 3 | 3. SYSTEM CONFIGURATION | 3. INSTRUCTION GROUPS | | | |
| 4 | 4. POWER DISTRIBUTION AND CLOCK DISTRIBUTION | | | | |
| 5 | 1. LAB FAMILIARIZATION | 1. INSTRUCTION SET THUMB-INS | 1. I/O PROGRAMMING THUMB-INS | 1. I/O PROGRAMMING THUMB-INS WITH INTERRUPTS | 1. ICL LOADING AND SCRATCHPAD CHECKOUT |
| 6 | 2. MODULE IDENTIFICATION | | | 2. MODIFIED I/O PROGRAMS TO CHECK STATUS | 2. INT/TRAP THUMB-INS |
| 7 | 3. CONTROL PANEL FAMILIARIZATION | | | | |
| 8 | LAB | LAB | LAB | LAB | LAB |

COURSE 340 PREPARED BY M. Courville & Hal Levitt

WEEK 1 of 2 DATE 7/31/81

# COURSE SCHEDULE

| | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY |
|---|---|---|---|---|---|
| 1 | RTOM | CPU | MEMORY | I/O | LARGE SYS. |
| 2 | 1. INTRO TO RTOM<br>A. PHY DESC<br>B. INSTALLATION<br>C. CHECKOUT | 1. INTRO TO CPU<br>A. PHY DESC<br>B. INSTALLA-TION<br>C. CHECKOUT | 1. INTRO TO MBC<br>2. MOS & CORE MEM | 1. INTRO TO IOM<br>2. IOM INSTALLA-TION AND CHECKOUT | 1. INTRO TO CLASS 'F' |
| 3 | 2. CONNECT EXTERNAL INTERRUPT | | 3. MEMORY INSTALL AND CHECKOUT | 3. NON-STANDARD IOM INSTALLA-TION AND CHECKOUT | |
| 4 | | 2. SEL BUS THEORY | 4. INTRO TO MAPPING | | |
| 5 | 1. INSTALLATION AND CHECKOUT | 1. CPU INSTALLA-TION AND CHECKOUT | 1. MEMORY INSTALLATION AND CHECKOUT | 1. IOM INSTALLATION AND CHECKOUT | REVIEW<br>TEST AND<br>CRITIQUE |
| 6 | 2. CONNECT EXT INTERRUPT AND CHECKOUT | | 2. MAPPING CHECKOUT | | |
| 7 | 3. TROUBLESHOOT | 2. TROUBLESHOOT | 3. TROUBLESHOOT | 2. TROUBLESHOOT | |
| 8 | | | | | |
| | LAB | LAB | LAB | LAB | |

COURSE 340 PREPARED BY M. Courville & Hal Levitt

WEEK 2 of 2 DATE 7/31/81

SYSTEMS 32/75 ARCHITECTURE (2 Weeks)

COURSE #340

COURSE OUTLINE

DAY 1

1. CPU BLOCK DIAGRAM

   A. BASIC CPU ELEMENTS
   B. BASIC MICROPROGRAM FLOW
   C. MAJOR COMPUTER ELEMENTS

2. SYSTEM BLOCK DIAGRAM

   A. SELBUS
   B. SYSTEM CONTROL PANEL
   C. MEMORY - (MBC'S, MEMORY BUS, MEMORY MODULES)
   D. RTOM
   E. I/O

3. SYSTEM CONFIGURATION

   A. SYSTEM LAYOUT DRAWINGS
      1) IDENTIFY HARDWARE MODULES(MODEL #'S)
      2) IDENTIFY CABINET DRAWINGS(104'S)
   B. SYSTEM KIT DRAWINGS (118'S)
      1) IDENTIFY CPU KIT
      2) SELBUS TERMINATOR KIT
      3) MEMORY BUS TERMINATOR KIT
      4) TLC CONTROLLER
      5) OTHER I/O KITS

4. POWER SUPPLIES AND A/C DISTRIBUTION

   A. A/C POWER DISTRIBUTION BLOCK DIAGRAM
   B. POWER FAIL DETECT SYSTEMS
   C. CLOCK DISTRIBUTION
   D. D/C POWER DISTRIBUTION
      1) POWER SUPPLY IDENTIFICATION
      2) POWER SUPPLY ADJUSTMENTS

5. SYSTEM CONTROL PANEL FAMILIARIZATION

DAY 2

1. OPERATING MODES

   A. PSW/PSD
   B. PRIVILEGED/NONPRIVILEGED

2. ADDRESSING MODES

   A. 512 KB
   B. 512 KB EXTENDED
   C. 512 KB MAPPED
   D. MAPPED EXTENDED

3. INTRODUCTION TO THE INSTRUCTION SET

   A. GENERAL PURPOSE REGISTERS
   B. INFORMATION BOUNDARIES IN MEMORY
   C. DATA BOUNDARIES
   D. INDEXING
   E. INDIRECT ADDRESSING
   F. PSW FORMAT
   G. PSD FORMAT

4. INSTRUCTION GROUPS

   A. MEMORY REFERENCE
   B. COMPARE
   C. BRANCH
   D. BIT MANIPULATION
   E. CONTROL
   F. ARITHMETIC

5. INSTRUCTION SET WORKSESSION

DAY 3

1. I/O CLASSES

   A. DATA TRANSFER TYPES
   B. RECORD LENGTH CAPABILITIES
   C. ADDRESSING CAPABILITIES AND LIMITATIONS
   D. DEVICE TYPES

2. I/O PROGRAMMING WITHOUT INTERRUPTS

   A. I/O INSTRUCTIONS
      1) COMMAND DEVICE
      2) TEST DEVICE
   B. TCW FORMATS
      1) DEDICATED MEMORY LOCATIONS
   C. SAMPLE PROGRAMS WITHOUT INTERRUPTS
   D. I/O PROGRAMMING WORKSESSION

DAY 4

1. I/O INTERRUPT PHILOSOPHY

   A. DEDICATED INTERRUPT LEVELS FOR I/O CONTROLLERS
   B. INTERRUPT INSTRUCTIONS AND STATES
   C. IVL DEDICATED LOCATIONS
   D. ICB FORMATS
      1) CLASS 0, 1, 2, AND E
      2) CLASS F

2. INTERRUPT FLOW

   A. SUBROUTINE ENTRY AND EXIT PROCEDURES
   B. SUBROUTINE CONTENT

3. INTERRUPT AND I/O PROGRAMMING

   A. INTERPRETING ASSEMBLY LISTINGS
   B. I/O PROGRAMS WITH INTERRUPTS
      1) CLASS 0, 1, 2
      2) CLASS E

4. MODIFY I/O PROGRAMS

   A. EXERCISE OTHER PERIPHERAL DEVICES
   B. CHECK STATUS OF DEVICE IN SUBROUTINE

5. I/O PROGRAMMING WORKSESSION

DAY 5

1. INTRODUCTION TO SCRATCHPAD

   A. FUNCTION AND LOCATION
   B. ICL DECK
      1) DEVICE ENTRY FORMAT
         A) DEVICE INTERRUPT ENTRY FORMAT
      2) INTERRUPT ENTRY FORMAT
   C. LOADING SCRATCHPAD
   D. DEVICE AND DEVICE INTERRUPT ENTRY LOCATION
   E. TRAPS/INTERRUPT ENTRY LOCATIONS
   F. SCRATCHPAD ROLLOUT AREA
   G. SCRATCHPAD INSTRUCTIONS

2. INTERRUPT AND TRAP PROCESSING

   A. CPU TRAPS - RTOM INTERRUPTS
   B. IVL DEDICATED LOCATIONS
   C. ICB FORMATS
      1) TRAP
      2) INTERRUPT
   D. AUTOMATIC TRAP HALT IMPLEMENTATION
   E. OPERATING SEQUENCE

3. SCRATCHPAD WORKSESSION

4. TRAP/INTERRUPT WORKSESSSION

DAY 6 1. INTRODUCTION TO THE RTOM

A. PHYSICAL DESCRIPTION/MODEL #'S
B. FUNCTIONAL DESCRIPTION
   1) EXTERNAL INTERRUPTS
   2) INTERVAL TIMER
   3) REAL TIME CLOCK
C. RTOM BLOCK DIAGRAM
D. OPERATING SEQUENCE
   1) LOAD RAM
   2) SOFTWARE REQUEST INTERRUPT
   3) HARDWARE EXTERNAL REQUEST INTERRUPT
   4) INTERRUPT POLLING
E. BOARD JUMPERING
F. CONNECT EXTERNAL INTERRUPT
   1) PHYSICAL CONNECTIONS
   2) SCRATCHPAD MODIFICATION
   3) PROGRAM CHECKOUT
G. INTERRUPT DIAGNOSTIC DESCRIPTION
H. RTOM WORKSESSION

DAY 7

1. INTRODUCTION TO THE CPU

   A. PHYSICAL DESCRIPTION/MODEL #'S
   B. BLOCK DIAGRAM DESCRIPTION (FUNCTIONAL)
   C. BOARD JUMPERING
   D. CABLING
   E. CHECKOUT (DIAGNOSTICS)

2. SELBUS THEORY

   A. DESCRIPTION OF LINES
   B. SELBUS TRANSFERS
   C. CD EMULATION SEQUENCE

3. CPU WORKSESSION

4. SELBUS WORKSESSION

DAY 8

1. MEMORY SUBSYSTEM

   A. PHYSICAL DESCRIPTION/MODEL #'S
   B. BLOCK DIAGRAM DESCRIPTION (FUNCTIONAL)
      1) WRITE OPERATION
      2) READ OPERATION
      3) OVERLAPPED OPERATION
      4) INTERLEAVING
   C. BOARD JUMPERING
   D. CABLING
   E. CHECKOUT (DIAGNOSTICS)

2. MAPPING

   A. MAP REGISTER DESCRIPTION
   B. FUNCTIONAL DESCRIPTION
   C. MAPPING MANAGEMENT INSTRUCTIONS
   E. DIAGNOSTICS

3. MEMORY/MAPPING WORKSESSION

DAY 9

1. INTRODUCTION TO THE IOM

   A. PHYSICAL DESCRIPTION/MODEL #'S
      1) TLC
      2) MHD
      3) MTC
   B. FUNCTIONAL DESCRIPTION
   C. BLOCK DIAGRAM DESCRIPTION (STANDARD IOM)
   D. DEVICE TYPES
   E. JUMPERING
   F. CABLING
   G. CHECKOUT (DIAGNOSTICS)
   H. NON-STANDARD IOM CONFIGURATION
      1) INSTALLATION
      2) CHECKOUT (THUMB-IN)

2. IOM WORKSESSION

DAY 10

1. INTRODUCTION TO CLASS 'F'

   A. DEDICATED MEMORY
   B. INTERRUPT CONTEXT BLOCK
   C. COMMAND FORMATS
   D. CLASS 'F" SIO SEQUENCE

2. REVIEW/EXAM

   CRITIQUE/GRADUATION

SYSTEMS 32/75 ARCHITECTURE COURSE OBJECTIVES

THE STUDENT WILL BE ABLE TO:

1. IDENTIFY AND LOCATE MAJOR COMPONENTS AND CABLING OF THE SEL COMPUTER.

2. OPERATE THE CONTROL PANEL WHILE ENTERING SHORT PROGRAMS INTO MEMORY AND EXECUTING THEM.

3. PERFORM PREVENTIVE MAINTENANCE ON THE SEL COMPUTER.

4. DECODE HEXADECIMAL INSTRUCTIONS ON PAPER IN ORDER TO DETERMINE WHAT THE MACHINE WILL BE DOING WHILE PERFORMING THAT INSTRUCTION.

5. LOAD AND EXECUTE DIAGNOSTIC PROGRAMS.

6. INTERPRET ERROR PRINTOUT MESSAGES FROM THE DIAGNOSTICS.

7. ANALYZE PROGRAM LISTINGS.

8. WRITE SHORT PROGRAMS FOR TROUBLESHOOTING PURPOSES.

9. RECOGNIZE NORMAL COMPUTER OPERATION.

10. ISOLATE A SYSTEM PROBLEM TO THE BOARD LEVEL.

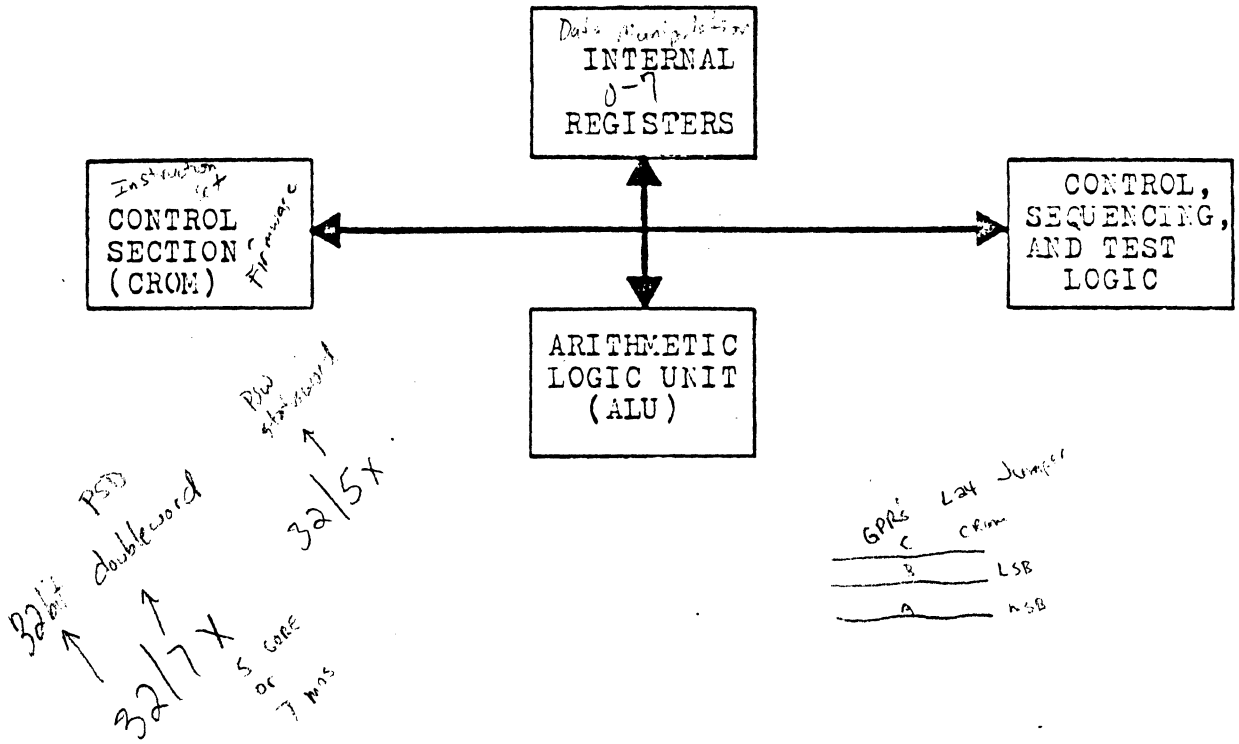
TABLE OF CONTENTS

# DAY 1

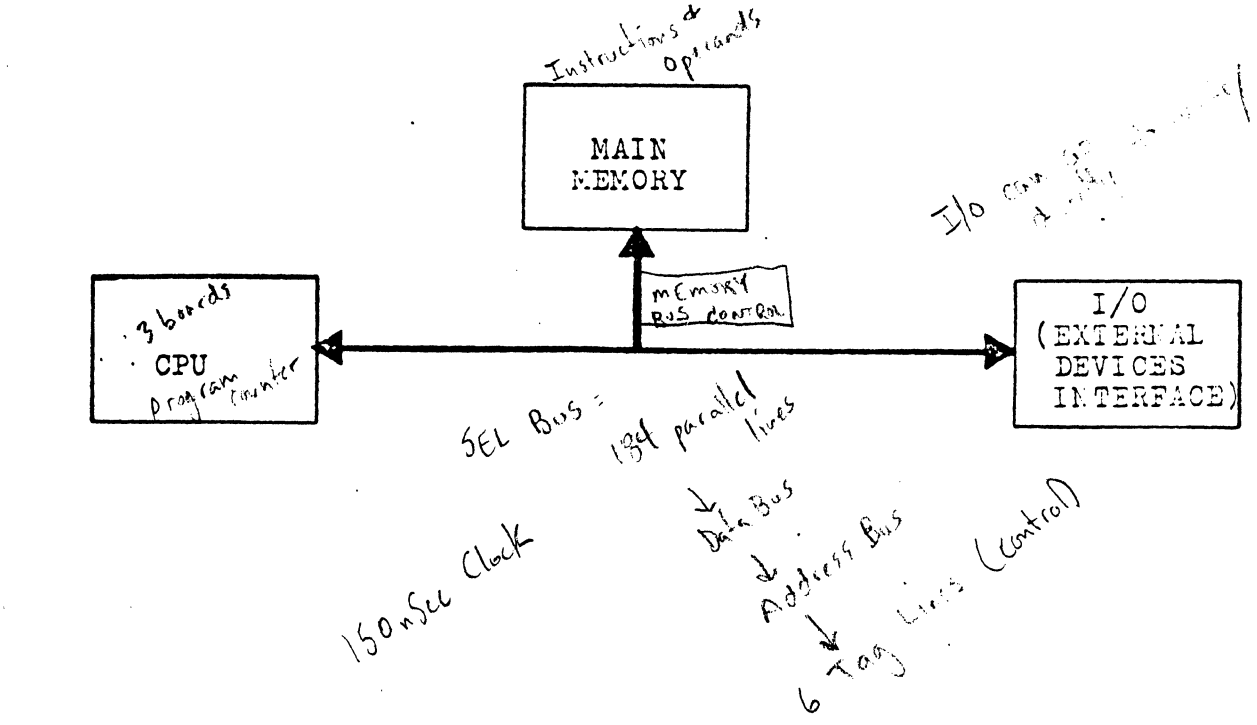
# SECTION 1

# SYSTEM CHARACTERISTICS

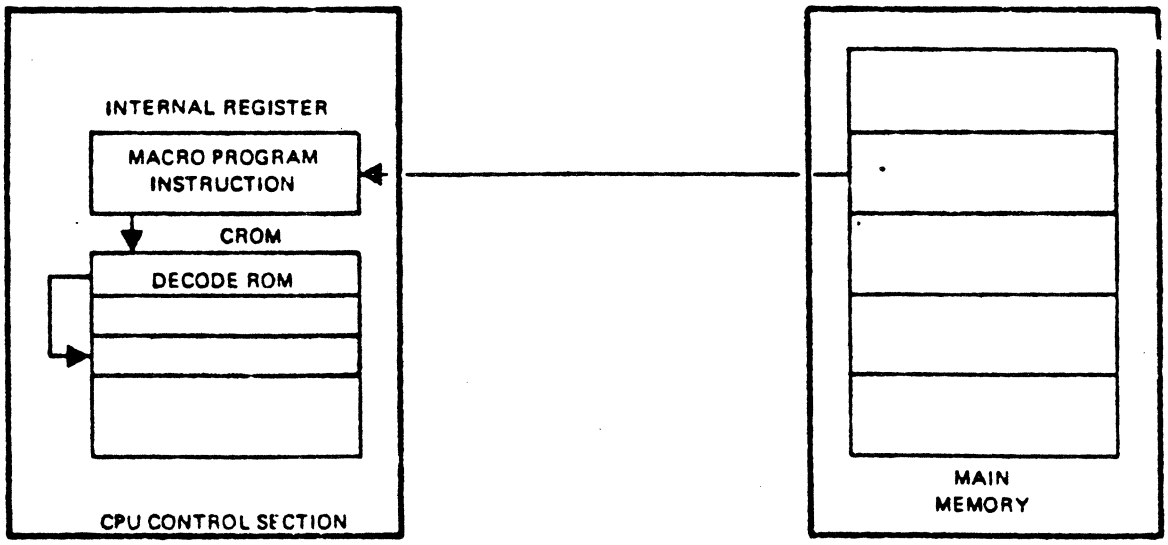
# CPU BLOCK DIAGRAM




# COMPUTER BLOCK DIAGRAM



# MICROPROGRAM IMPLEMENTATION



## MODEL NUMBER CHART

### CPU OPTIONS

| MODEL # | DESCRIPTION |
|---|---|
| 2000 | 32/55 CPU (WIRE/WRAP) |
| 2003 | 32/70 SERIES CPU (COPPER) |
| 2005 | 32/70 SERIES CPU/IPU (COPPER) |
| 2117 | RTOM (WIRE/WRAP) |
| 2118 | RTOM (WIRE/WRAP) |
| 2345 | RTOM (COPPER) |
| 2140 | TURNKEY PANEL |
| 2142/2346 | SYSTEM CONTROL PANEL (PARALLEL) WITH SCPI |
| 2145 | HEX DISPLAY |
| 2146 | CONTROL PANEL (SERIAL) WITH HEX DISPLAY |
| 2341 | HIGH SPEED FLOATING POINT UNIT (MULTI-WIRE) |
| 2342 | HIGH SPEED FLOATING POINT UNIT (WIRE/WRAP) |
| 2343 | SCIENTIFIC ACCELERATOR (WCS) |
| 2344 | WRITABLE CONTROL STORAGE (WCS) |
| 2347 | SCIENTIFIC ACCELERATOR (PCS) |
| 2181 | LOGIC CHASSIS |

## MEMORY OPTIONS

| MODEL # | DESCRIPTION |
|---|---|
| 2150 | CORE MEMORY BUS CONTROLLER (OLD WIRE/WRAP) |
| 2162 | CORE MEMORY BUS CONTROLLER (COPPER) 4 PORT, 600/900N |
| 2164 | CORE MEMORY BUS CONTROLLER (WIRE/WRAP) 2 PORT, 600 N |
| 2168 | CORE MEMORY BUS CONTROLLER (900 NSEC) |
| 2152 | 8KW CORE MEMORY MODULE (600 NSEC) |
| 2153 | 16KW CORE MEMORY MODULE (900 NSEC) |
| 2377 | MOS MEMORY BUS CONTROLLER (32KW/900 NSEC) |
| 2382 | MOS MEMORY BUS CONTROLLER |
| 2158 | 32KW MOS MEMORY MODULE (600 NSEC/SEL BUS) |
| 2160 | 64KW MOS MEMORY MODULE (600 NSEC/SEL BUS) |
| 2378 | 32KW MOS MEMORY MODULE (600 NSEC) |
| 2376 | 32KW MOS MEMORY MODULE (900 NSEC) |
| 2379 | 64KW MOS MEMORY MODULE (600 NSEC) |
| 2381 | 64KW MOS MEMORY MODULE (900 NSEC) |
| 2178 | MEMORY BUS ADAPTER (MBA) |
| 2179 | MEMORY INTERFACE ADAPTER (MIA) |
| 2374 | BATTERY BACKUP UNIT |
| 2182 | MEMORY CHASSIS (32/55) |

## CLOCKS/POWER FAIL SAFE/POWER SUPPLIES

| MODEL # | DESCRIPTION |
|---|---|
| 2190 | LOGIC POWER SUPPLY (500W) |
| 2191 | MEMORY POWER SUPPLY (500W) |
| 2192 | GPDC POWER SUPPLY |
| 2195 | AC DISTRIBUTION |
| 2128 | MULTIPROCESSOR CENTRAL TIMING UNIT |
| 2129 | MULTIPROCESSOR CENTRAL TIMING DECODER |
| 2134 | POWER FAIL W/O DC SENSE (32/50 & 32/70 SERIES) |
| 2136 | POWER FAIL WITH DC SENSE (32/50 & 32/70 SERIES) |
| 2138 | MULTIPROCESSOR CENTRAL TIMING SOURCE |
| 2139 | MULTIPROCESSOR CENTRAL TIMING SOURCE CABLE |

## PROCESSING UNITS

| MODEL # | DESCRIPTION |
|---|---|
| 8000 | INPUT/OUTPUT PROCESSOR (IOP) - SUPPORTS UP TO 16 IOP CONTROLLERS ON AN MP BUS - ANY CPU. |
| 8030 | LP/FDD CONTROLLER |
| 8032 | IOP DISC CONTROLLER |
| 9144 | REGIONAL PROCESSING UNIT (RPU) - USER PROGRAMMABLE |
| 9145 | REGIONAL PROCESSING UNIT (RPU WITH RAM) |

## DISC SUBSYSTEM

| MODEL # | DESCRIPTION |
|---|---|
| 9010 | MOVING HEAD DISC CONTROLLER (WIRE/WRAP) |
| 9024 | DISC PROCESSOR |
| 8055 | DISC PROCESSOR II (16 MB ADDRESSING) |
| 9008 | CARTRIDGE DISC CONTROLLER (WIRE/WRAP) |
| 9009 | CARTRIDGE DISC CONTROLLER (COPPER) |
| 9014 | FIXED HEAD DISC CONTROLLER |

## TAPE SUBSYSTEM

| MODEL # | DESCRIPTION |
|---|---|
| 9012 | MAG TAPE CONTROLLER (WIRE/WRAP) |
| 9013 | MAG TAPE CONTROLLER (COPPER) |
| 8050 | HIGH SPEED TAPE PROCESSOR (HSTP) - HANDLES 75/125 IPS DUAL/TRI DENSITY TAPE DRIVES. |
| 9020 | LOW SPEED TAPE PROCESSOR (LSTP) - HANDLES 45/75 IPS, 800/1600 BPI, NRZ/PE TAPE DRIVES. |

## INTERFACES

| MODEL # | DESCRIPTION |
|---|---|
| 9004 | TLC CONTROLLER (WIRE/WRAP) TTY,LP,CR |
| 9005 | TLC CONTROLLER (COPPER) TTY,LP,CR |
| 9102 | GENERAL PURPOSE I/O CONTROLLER (GPIO) |
| 9103 | GENERAL PURPOSE MULTIPLEXER CONTROLLER (GPMC 16 MB) |
| 9104 | GENERAL PURPOSE MULTIPLEXER CONTROLLER (GPMC) |
| 9105 | GPDC CHASSIS |
| 9106 | GENERAL PUPOSE DEVICE CONTROLLER (GPDC) |
| 9107 | CARD PUNCH CONTROLLER |
| 9108 | CARD READER/PUNCH CONTROLLER |
| 9112 | PAPER TAPE READER/PUNCH CONTROLLER |
| 9115 | GPDC COUPLER |
| 9120 | GPDC TEST KIT |
| 9131 | HIGH SPEED DATA INTERFACE (HSD II) - COPPER |
| 9132 | HSD (WIRE/WRAP) |
| 9135 | HSD INTER-BUS LINK (IBL II) - COPPER |
| 9136 | HSD IBL (WIRE/WRAP) |
| 9134 | SERIAL DATA INTERFACE |
| 7410 | ANALOG/DIGITAL INTERFACE |

**********
* *
* DATA COMMUNICATIONS *
* *
**********

| MODEL # | DESCRIPTION |
|---|---|
| 9109 | SYNCHRONOUS LINE INTERFACE MODULE (SLIM) |
| 9110 | ASYNCHRONOUS LINE INTERFACE MODULE (ALIM) |
| 9116 | BINARY SYNCHRONOUS LINE INTERFACE MODULE (BLIM) |
| 9122 | ASYNCHRONOUS DATA SET (ADS) |

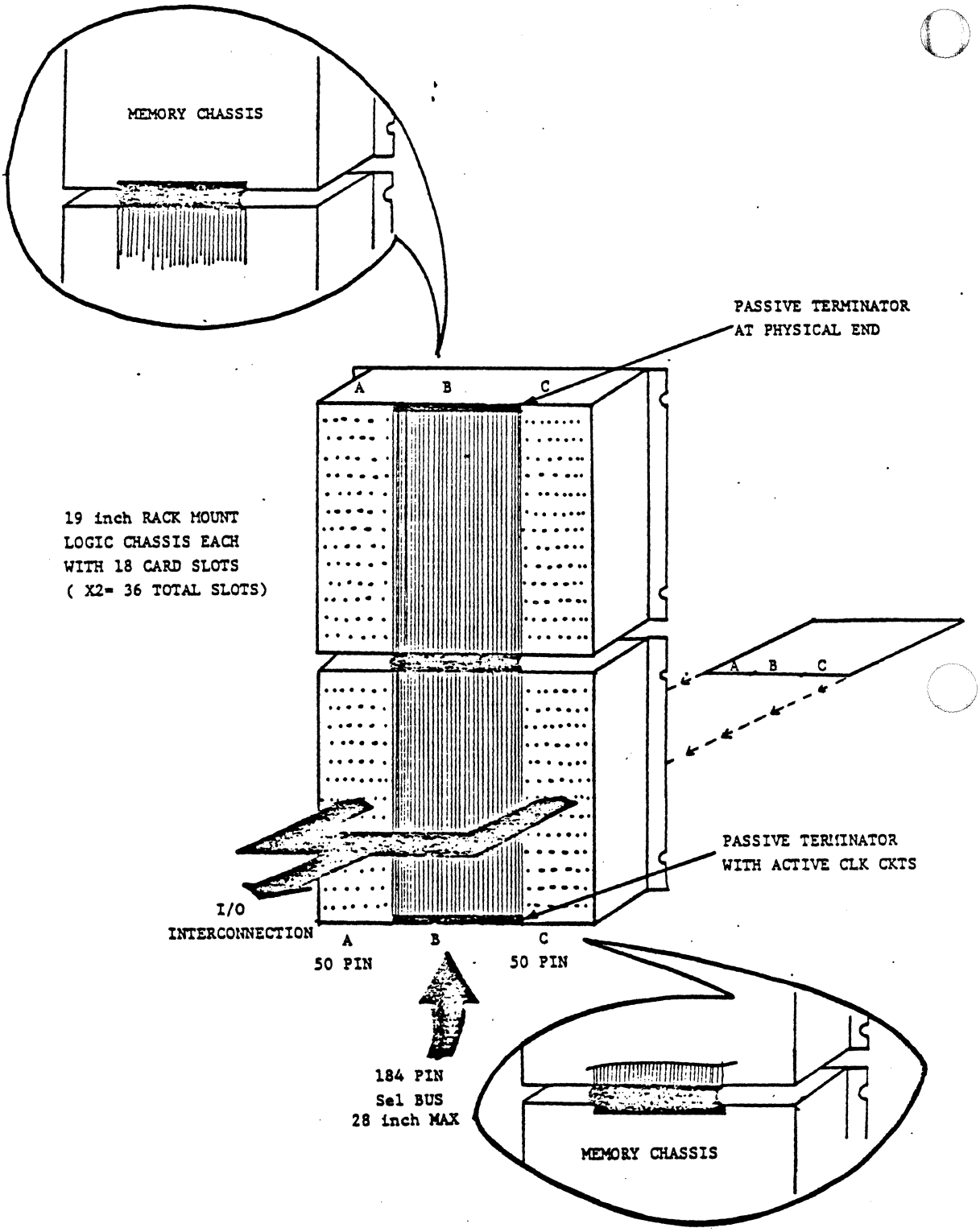
**********
* *
* FAST MUX SYSTEM *
* *
**********

| MODEL # | DESCRIPTION |
|---|---|
| 9180 | FAST MULTIPLEXER SYSTEM (FMS) |
| 9181 | FAST MUX DEVICE CONTROLLER (FDC) |
| 9182 | FAST DEVICE INTERFACE (FDI) |

**********TH-TH-THAT'S ALL FOLKS**********


- 32-BIT BI-DIRECTIONAL DATA PATH
- 24-BIT ADDRESS PATH ( 16Mb ADDRESSING )
- INTERRUPT CONTROL LINES FOR 128 MAX INTERRUPTS 112
- 6.67 MHz BUS CLK WHICH GIVES:
  - 150 nsec BUS TRANFER CYCLE TIME
  - 6.67 MW/SEC (26.67 MB/SEC) BUS THROUGHPUT
- 28 INCH BUS LENGTH WITH:
  - 36 CARD SLOTS MAX
  - NO SLOT DEDICATION
  - HIGH SPEED SCHOTTKY TTL DRIVERS/RECEIVERS
  - PASSIVE TERMINATION
  - I/O CARDS WITH ON-BOARD DMA CONTENTION LOGIC

# CHASSIS LAYOUT



# CENTRAL PROCESSOR UNIT (CPU)

WORD LENGTH IS 32 BITS

- OVERLAPPED INSTRUCTION EXECUTION

  1.2 MICROSECOND INSTRUCTION EXECUTION (TYPICAL)

- OVERLAPPED INSTRUCTION OPERAND FETCH

  .8 MIPS (STRAIGHT LINE CODE) No indexing No indirecting

  545K WHETS

- MICROPROGRAMMED (FIRMWARE)

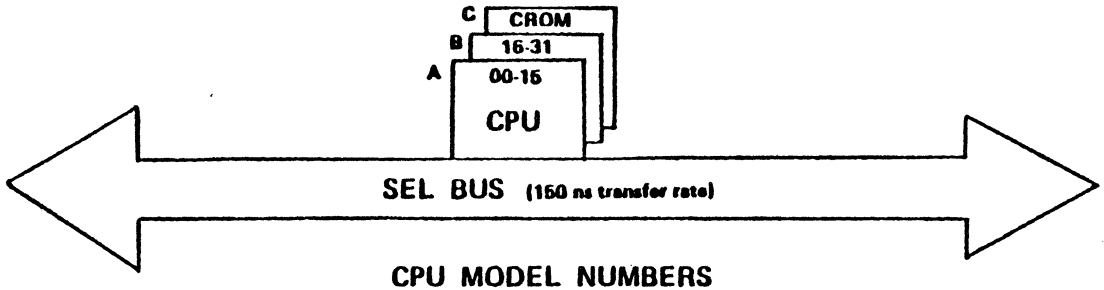
  150 NS PER MICRO-INSTRUCTION

  4K X 64 BIT ROM (000-FFF)

- STANDARD FLOATING POINT (FIRMWARE) → 16 of 64 bit faster execution
- HAS 8 GENERAL PURPOSE REGISTERS 0-7 GPRs sys 4 Priv
- HARDWARE/FIRMWARE MEMORY MANAGEMENT (MAPPING)
- OPERATES IN TWO MODES:

| PSW 3255 | 3277 PSD | |
|---|---|---|
| RTM OPERATING SYSTEM ONLY | RTM OPERATING SYSTEM | MPX OPERATING SYSTEM |
| 161 INSTRUCTIONS macro | 161 + INSTRUCTIONS | 187 INSTRUCTIONS |
| No Traps SYSTEM INTERRUPTS ON RTOM | SYSTEM INTEGRITY TRAPS | SYSTEM INTEGRITY TRAPS |
| DIRECT ADDRESS 128 KW | DIRECT ADDRESS 128 KW | MAPPING (ADDRESS 16 MB) |
| NO CLASS 'F' I/O | CLASS 'F' I/O | CLASS 'F' I/O |
| I/O (ADDRESS 128 KW ONLY) | I/O (ADDRESS 16 MB) | I/O (ADDRESS 16 MB) |

# CENTRAL PROCESSING UNIT



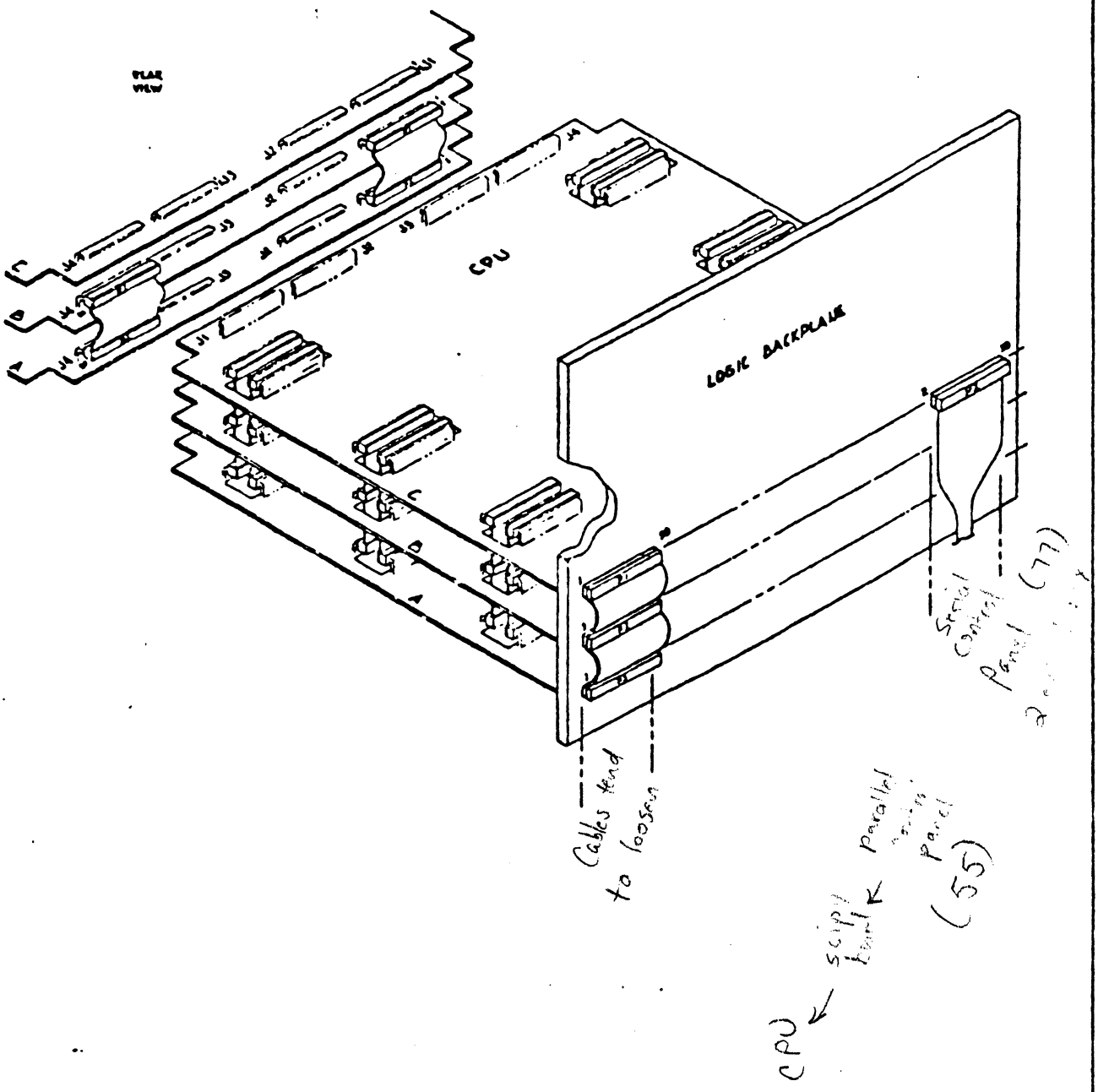

CPU MODEL NUMBERS

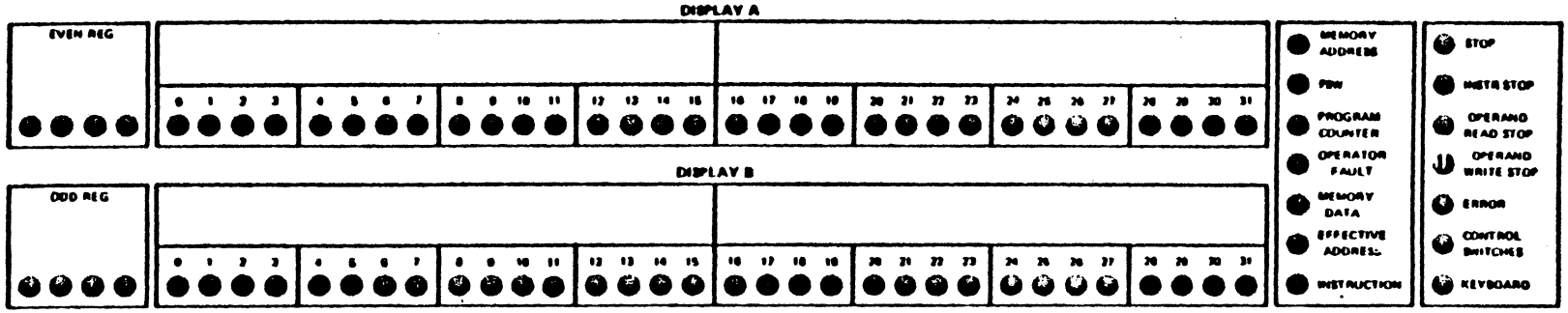
2000 WIRE/WRAP

2003 COPPER

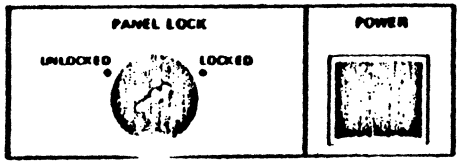
2005 CPU-IPU

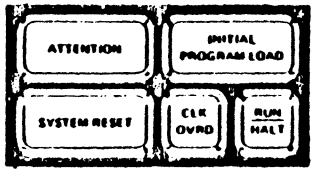
# CPU CIRCUIT CARDS

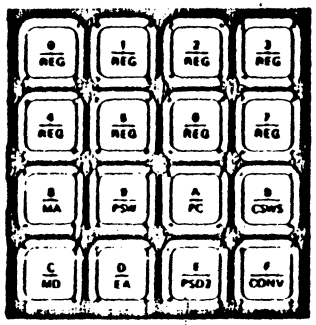


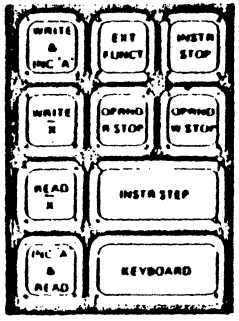



SYSTEMS
ENGINEERING LABORATORIES

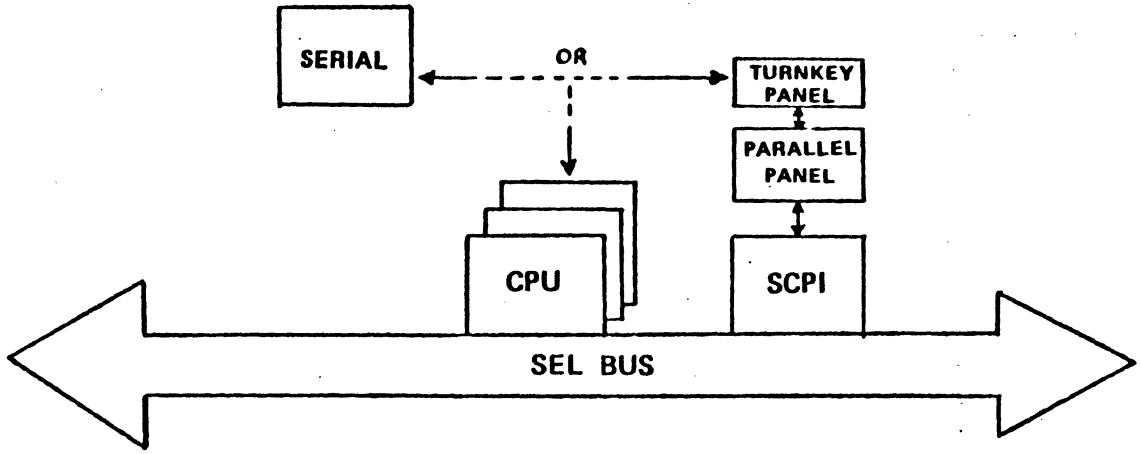




Read Reg
① Read C
② Reg +

Section 7 of Reference Manual

① Keyboard (clear PC)
② 1000
③ write/x
④ B/MA
⑤ Read
⑥ C

# SYSTEM CONTROL PANELS




SERIAL CONTROL PANEL – MODEL 2146

PARALLEL CONTROL PANEL WITH SCPI – MODEL 2346

SCPI – MODEL 2142

# MEMORY SUBSYSTEM

CONTROLLED BY MEMORY BUS CONTROLLER (MBC)

MBC INTERFACES MEMORY MODULE TO SEL BUS
CHECKS AND GENERATES PARITY (CORE) ECC (mos)
MBC CONTROLS UP TO 16 MEMORY MODULES

MBC CAN ADDRESS:

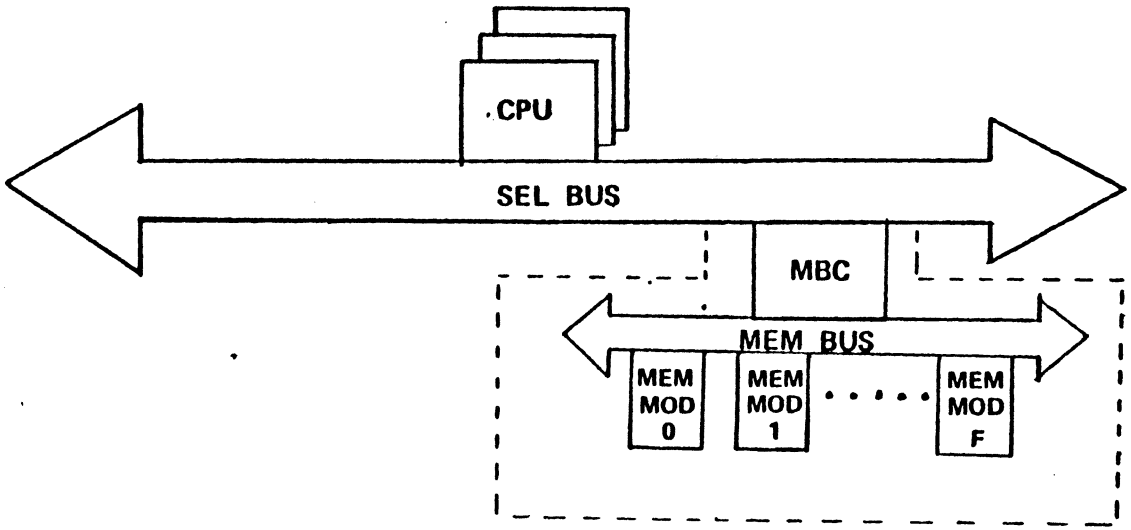
BIT: SMALLEST MEMORY VALUE ADDRESS BY INSTRUCTION Not Bit Addressable
BYTE: 8 BITS OR 2 HEX CHARACTERS
HALFWORD: 16 BITS (LEFT OR RIGHT)
FULLWORD: 32 BITS - 2 HW - 4 BYTES/8 HEX CHARACTERS
DOUBLEWORD: 64 BITS (2 WORD FETCH)

MEMORY MODULES:

8/16 KW (32/64 KB) CORE MODULES, BYTE PARITY (55)

32/64 KW (128/256 KB) MOS MODULES, WORD ECC (77) 2 mos

# MEMORY SUBSYSTEM




## CORE MBC'S

2150 WIRE/WRAP
2162 COPPER - 4 PORT, 600/900NS
2164 WIRE/WRAP - 2 PORT, 600NS

## MOS MBC'S

2377 32KW/900NS
2382 COPPER

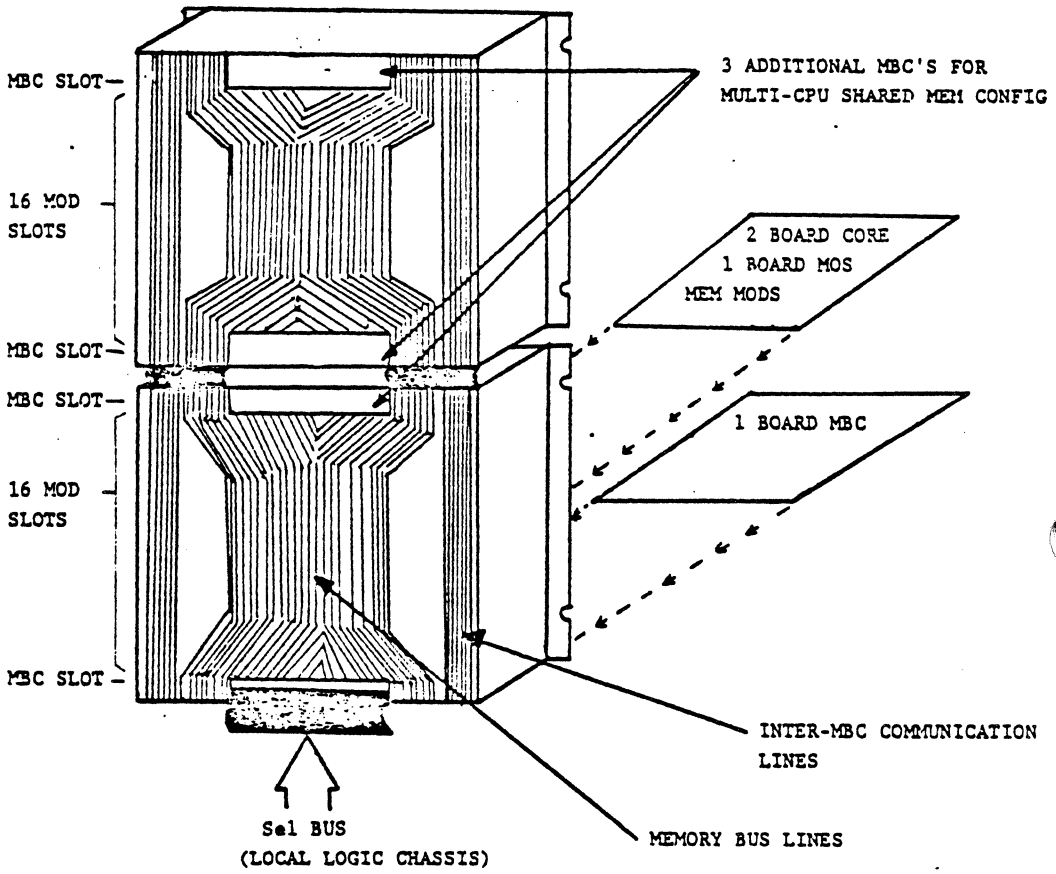
## CORE MEMORY MODULES

2152 8KW/600 NS
2153 16KW/900 NS

## MOS MEMORY MODULES

2378 32KW/600 NS
2376 32KW/900 NS
2379 64KW/600 NS
2381 64KW/900 NS

# MEMORY CHASSIS LAYOUT

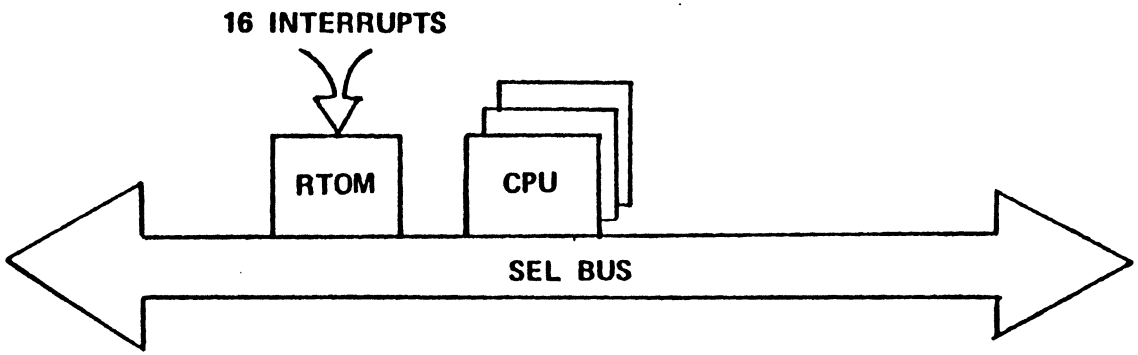



MBC SYS:

CORE - 8 MODS PER CHASSIS X 2 MEM CHASSIS (28 inch MEM BUS)= 16 MODS:
WITH 8KW MODS = 128KW (512 KB)
WITH 16KW MODS = 256KW ( 1 MB )

MOS - 16 MOS MODS IN ONE CHASSIS ONLY:
16 X 64KW = 1024KW ( 4 MB )

# REAL TIME OPTION MODULE




RTOM MODEL 2117 WIRE/WRAP
2118 WIRE/WRAP
2345 COPPER

* SYSTEM REQUIRES AT LEAST ONE AND MAY HAVE 7 MAX
* THE SEL BUS INTERRUPT CONTROL LINES PROVIDE FOR 112 INTERRUPT LEVELS
* EACH RTOM PROVIDES:

  16 INTERRUPTS, 32-BIT INTERVAL TIMER, 60/120 HZ RTC, ATTENTION
* THE INTERRUPT STRUCTURE IS FULLY PROGRAMMABLE BY THE USER TO MANIPULATE THE PRIORITY OF ANY INTERRUPT EVENT.

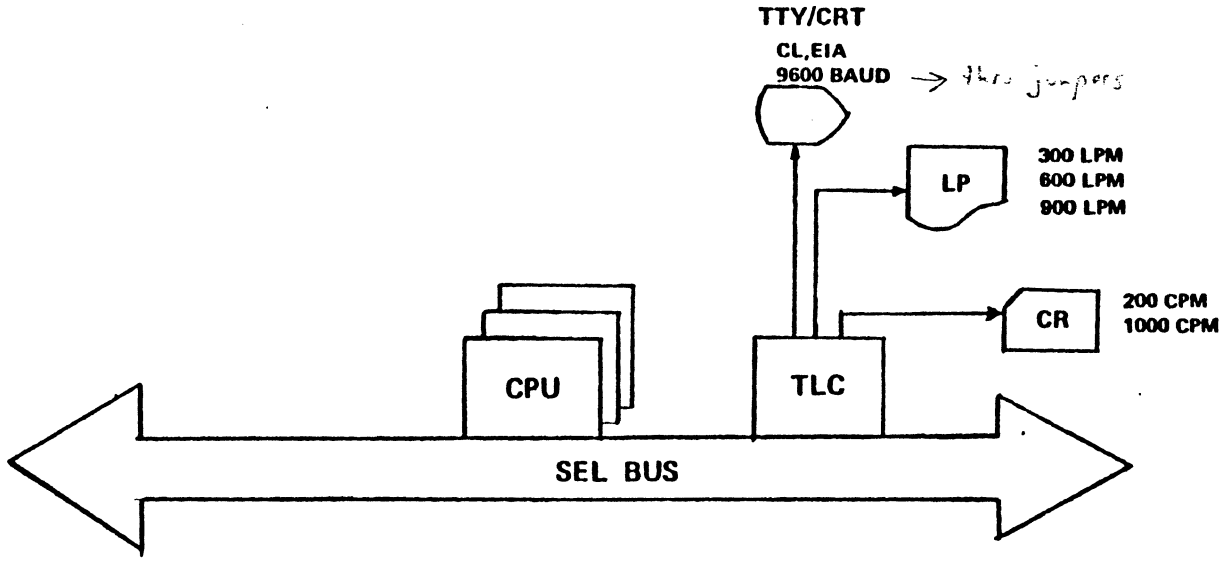
- INPUT/OUTPUT MICROPROGRAMMABLE PROCESSOR (IOM)

  PROVIDES DIRECT COMMUNICATION BETWEEN PERIPHERAL & SEL BUS
  FIRMWARE PROGRAMMABLE I/O CONTROL
  GENERATES I/O SERVICE INTERRUPTS (INTERNAL SI) when I/O terminates
  Mag tape, disk, teletype, controllers

- TLC - TTY/LINE PRINTER/CARD READER CONTROLLER
  Controls 3 devices

  MULTIPLEXER CHANNEL (ALL 3 UNITS ACTIVE)
  DEDICATED I/O CONNECTOR SLOTS
  3 SEPARATE SERVICE INTERRUPTS .

# CONSOLE DEVICES




TLC CONTROLLER
- MODEL 9004 WIRE/WRAP
- MODEL 9005 COPPER

# MAG TAPE SUBSYSTEM

MAG TAPE CONTROLLER - MODEL #9012 (W/W)
#9013 (PC)

* 4 UNITS MAX
* SELECTOR CHANNEL (ONLY 1 UNIT ACTIVE AT A TIME)
* CLASS E - 128 KW ADDRESSING
* 4096 HALFWORD = 8192 BYTE RECORD LENGTH

LOW SPEED TAPE PROCESSOR - MODEL #9020

* 4 UNITS
* ~~MULTIPLEXER~~ SELECTOR CHANNEL (ALL 4 ACTIVE)
* CLASS 'F' - 16 MB ADDRESSING
* INFINITE RECORD SIZE (data chaining)

HIGH SPEED TAPE PROCESSOR - MODEL #8050

* 4 UNITS
* MULTIPLEXER CHANNEL
* CLASS 'F' - 16 MB ADDRESSING
* INFINITE RECORD SIZE

SUPPORTS PERTEC & KENNEDY:

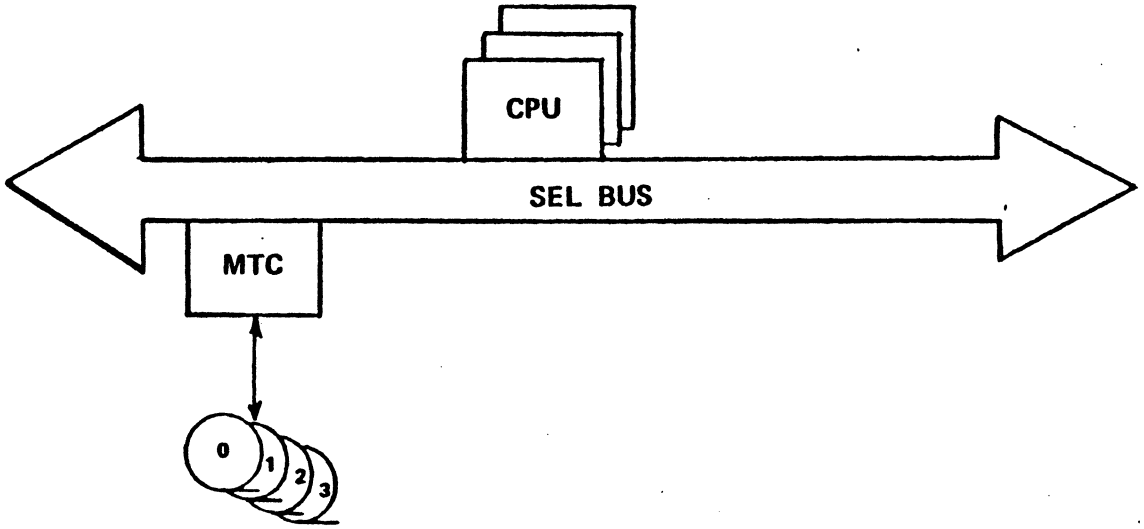
*IMBEDDED FORMATTER
*45 IPS TENSION
*75 IPS VACUUM
*800 BPI (NRZI)
*1600 BPI (PE)

55

DEDICATED TO STC DRIVES:

*75 IPS VACUUM
*125 IPS VACUUM
*800 BPI (NRZI)
*1600 BPI (PE) 1 BIT CORRECT
*6250 BPI (GCR) 2 BIT CORRECT
* AUTO LOAD & UNLOAD
* AUTO & MANUAL DENSITY SELECT

# MAG TAPE SUBSYSTEM




MAG TAPE CONTROLLER - MODEL # 9012 (W/W)
# 9013 (PC)

LOW SPEED TAPE PROCESSOR - MODEL # 9020

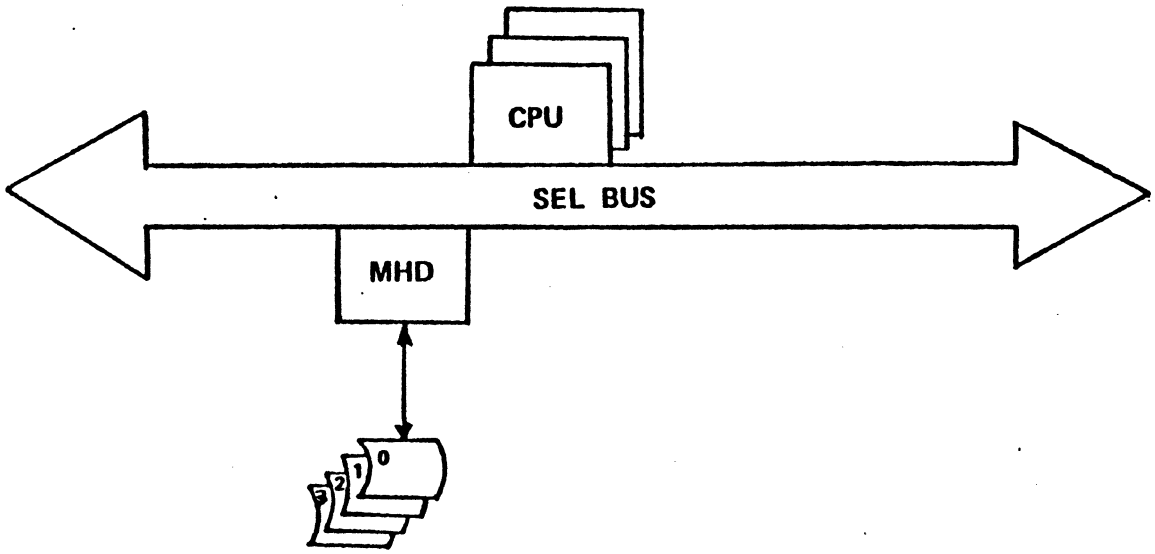
HIGH SPEED TAPE PROCESSOR - MODEL # 8050

# DISC SUBSYSTEM

2 boards 1 bay

| MODEL #: | 9010 (W/W) * | 73-9032 (W/W) | 9024 DISC PROCESSOR | 8055 DP II (Same VDP) |
|---|---|---|---|---|
| CLASS: | E, 128 KW ADDR | E, 16 MB | F, 16 MB | F, 16 MB |
| DEVICE TYPE: | 5 heads 19 heads 40, 80, 150, 300 MB drive size must be the same | 40, 80, 150, 300 MB | 80, 300 MB | 5 MB FHD, 32 MB CMD 80, 300 MHD, 600 FMD |
| CHANNEL OPERATION: | SELECTOR CHAN (4 UNIT MAX) ALL SAME DEVICES | SELECTOR CHAN (4 UNIT) DUAL PORT ALL SAME | MULTIPLEXER (8 UNIT MAX) DUAL PORT (MIXED DEVICES) | MULTIPLEXER (8 UNIT MAX) DUAL PORT (MIXED DEVICES) |
| ERROR CORRECTION: | 9 BIT ECC all disks | 9 BIT ECC | 9 BIT ECC | 9 BIT ECC |

# DISC SUBSYSTEM

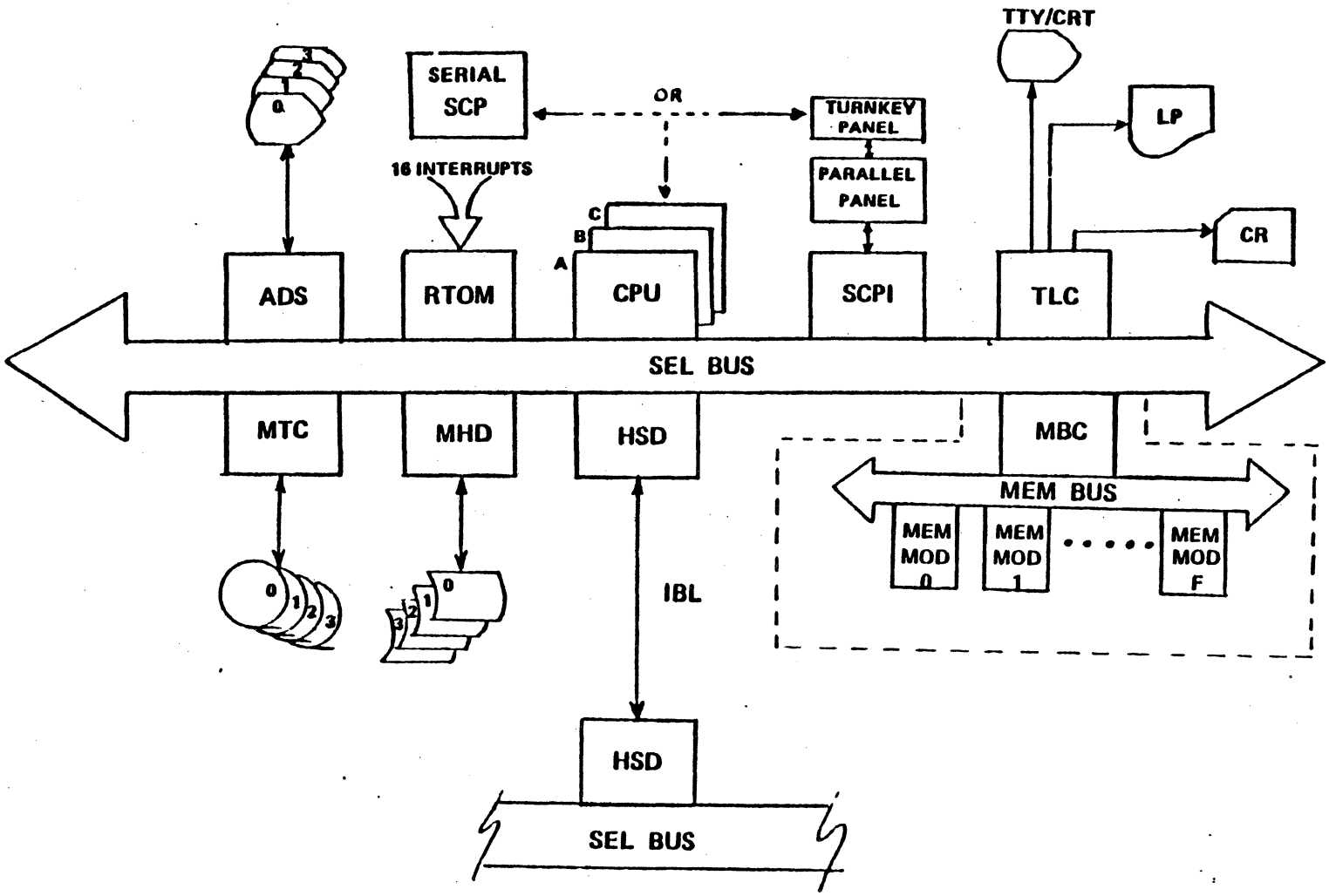



MHD CONTROLLER - MODEL # 9010 (W/W)
# 73-9032 (W/W)

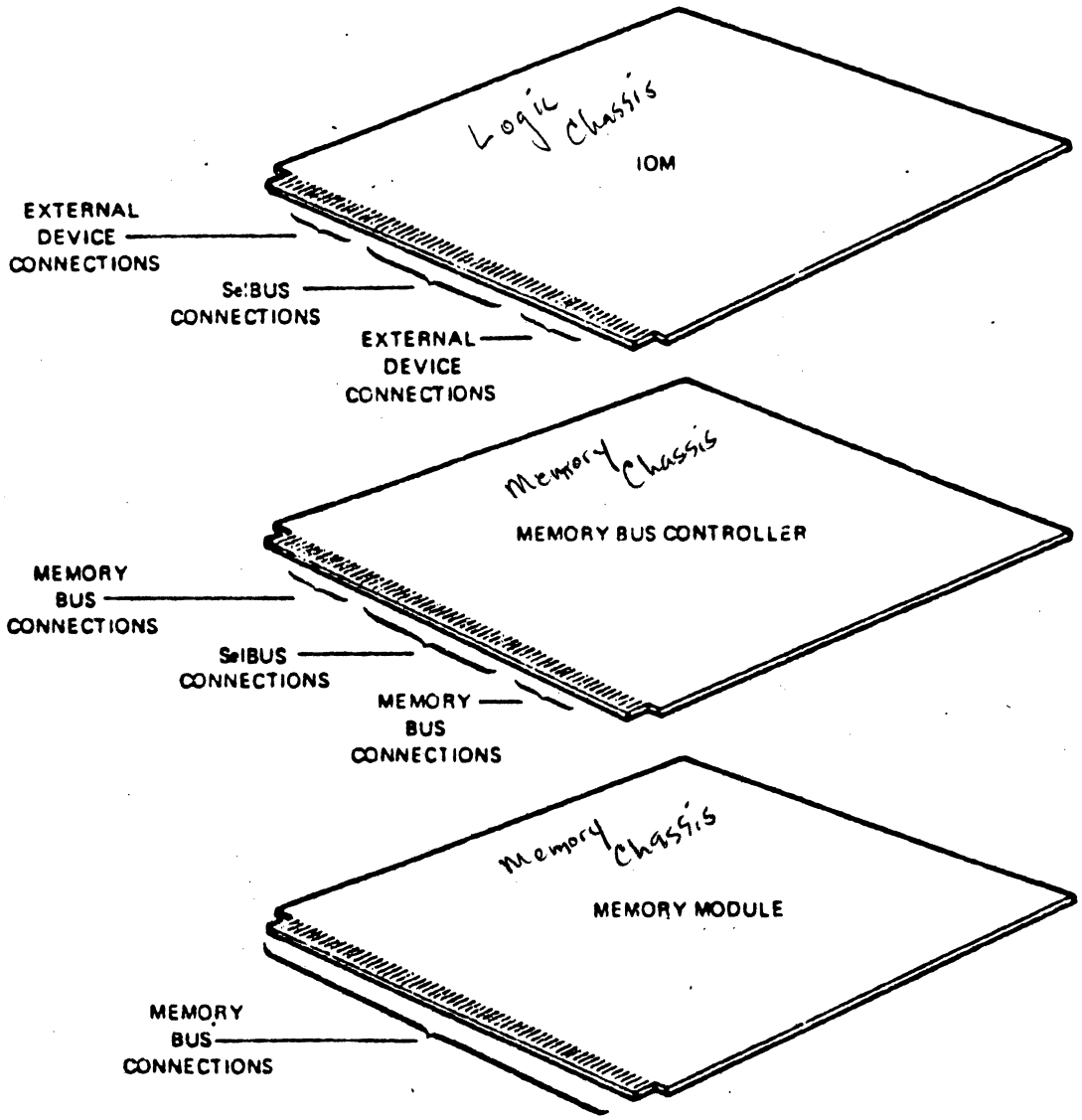
DISC PROCESSOR - MODEL # 9024

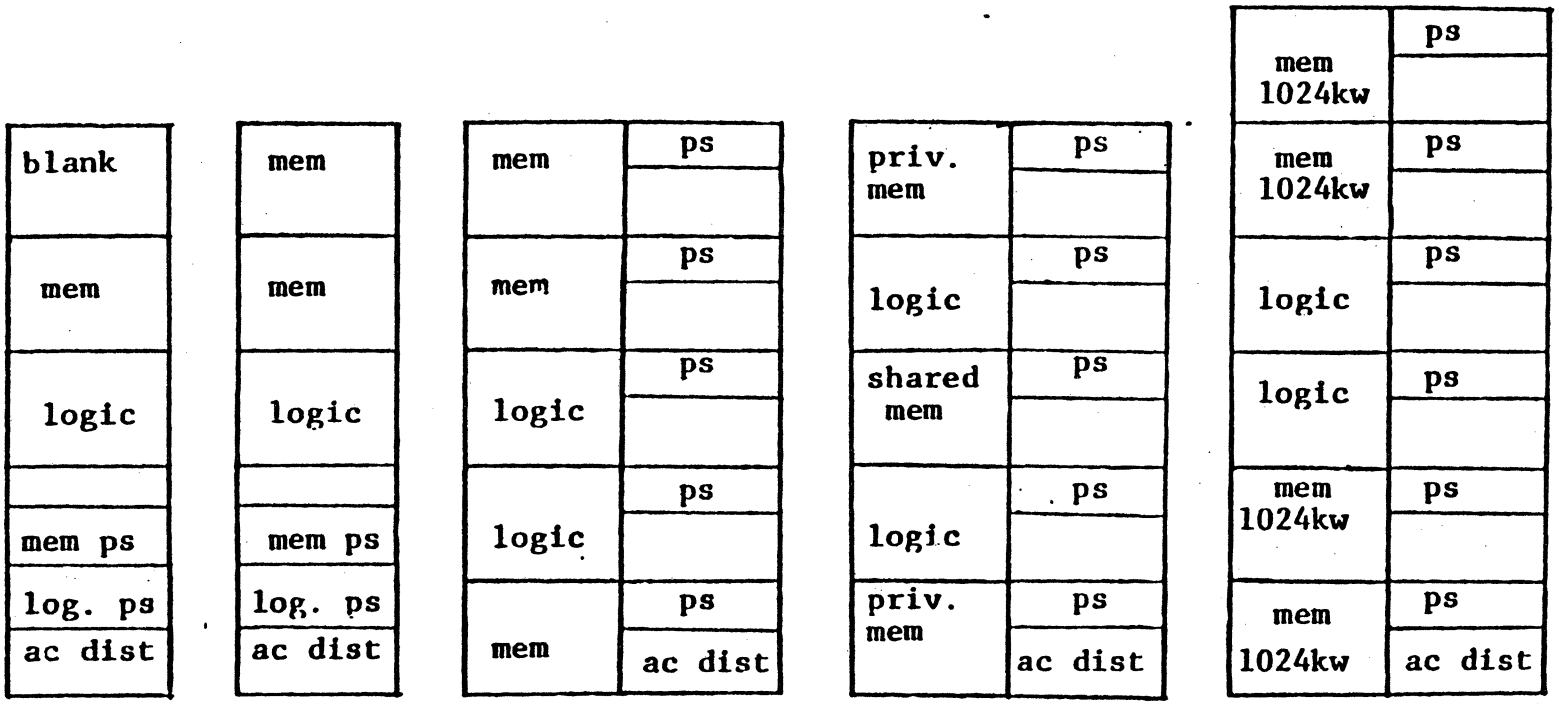
DISC PROCESSOR II - MODEL # 8055

# INTEGRATED SYSTEM



# MODULE PIN ASSIGNMENTS






TYPICAL
SINGLE

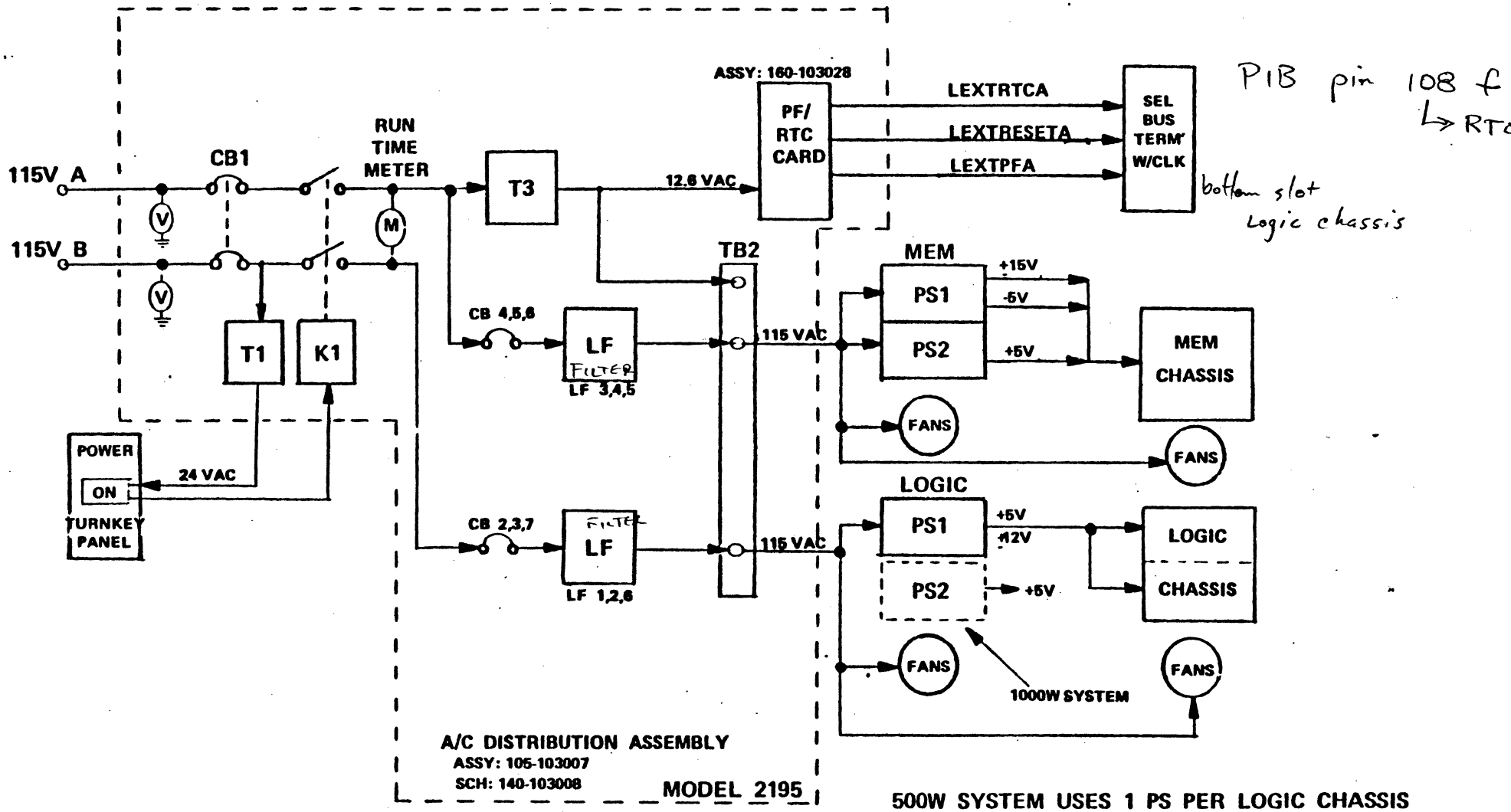
TYPICAL
DOUBLE

TYPICAL
DOUBLE
dual CPU
shared mem

TYPICAL
TALL DOUBLE
single CPU 16 MB mos

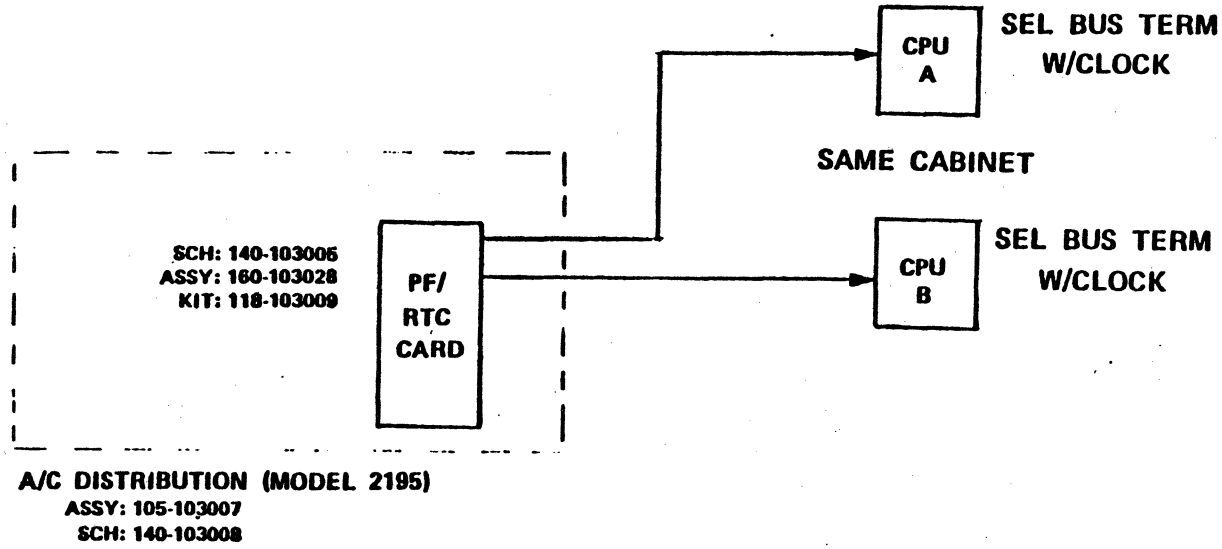
TYPICAL CABINET ORGANIZATION

# A/C POWER SYS BLOCK DIAGRAM

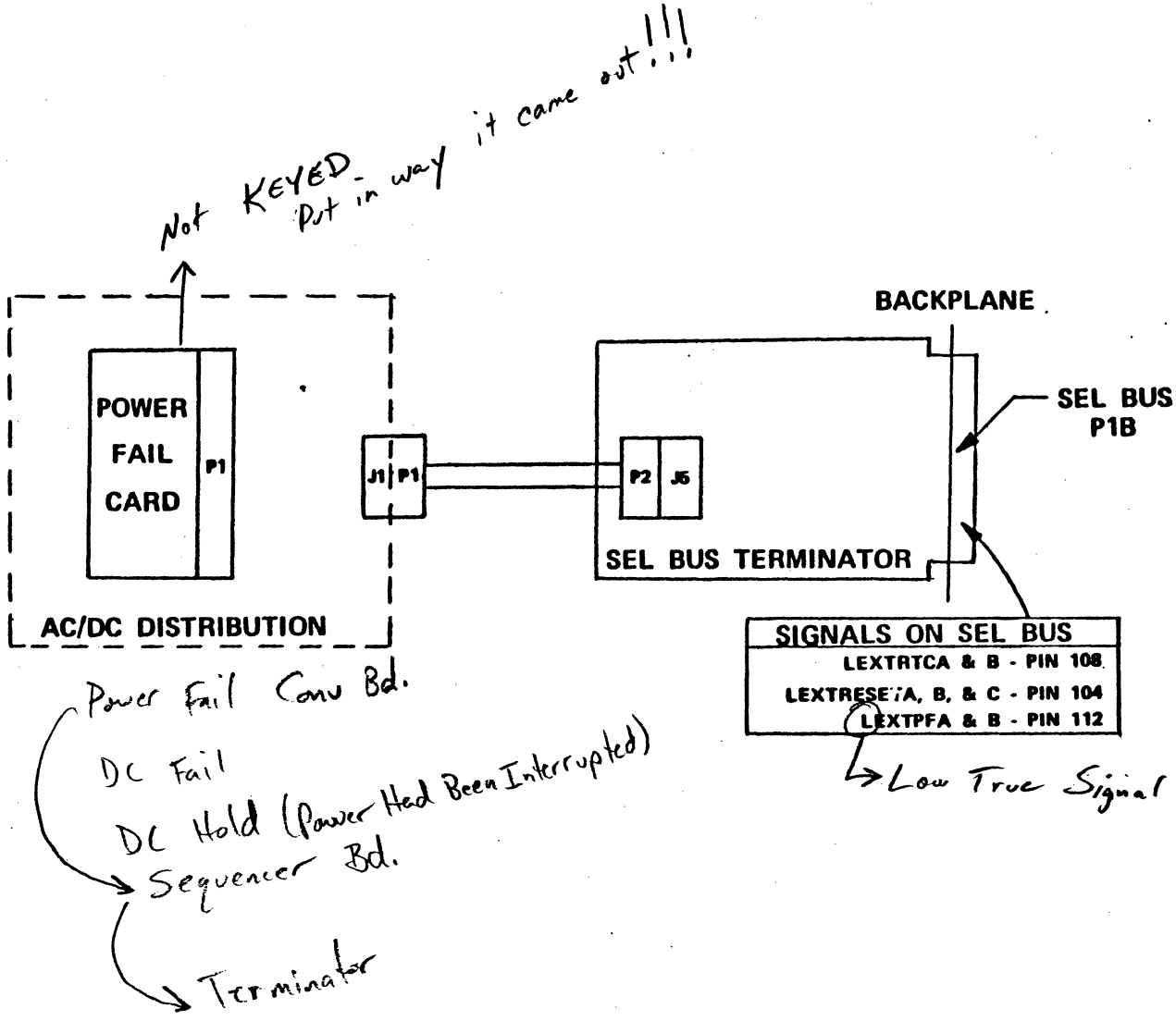



500W SYSTEM USES 1 PS PER LOGIC CHASSIS
1500W & 1000W SYSTEMS USE 2 PS PER LOGIC CHASSIS

# POWER FAIL/RTC DISTRIBUTION
# SINGLE CABINET/DUAL CPU



# POWER FAIL CARD TO SEL BUS



## POWER FAIL DETECTION CARD ADJUSTMENT

THE PURPOSE OF THE POWER FAIL DETECTION CARD, 160-103028-XXX IS TO SENSE THE LOSS OF OR RESTORATION OF AC POWER AND TO GENERATE AN INTERRUPT AT PRIORITY LEVEL 00 TO THE CENTRAL PROCESSING UNIT (CPU) SO THAT THE SOFTWARE/FIRMWARE CAN SAVE THE GENERAL PURPOSE REGISTERS, CPU SCRATCHPAD AND PROGRAM STATUS DOUBLEWORD INTO MAIN MEMORY AND RESTORE SAME UPON RESTORATION OF POWER. THIS CARD ALSO PROVIDES A REFERENCE SIGNAL FOR THE REAL TIME CLOCK ON THE RTOM.

A PRACTICAL METHOD FOR ADJUSTING THE SENSITIVITY OF THE POWER FAIL DETECTOR IS AS FOLLOWS:

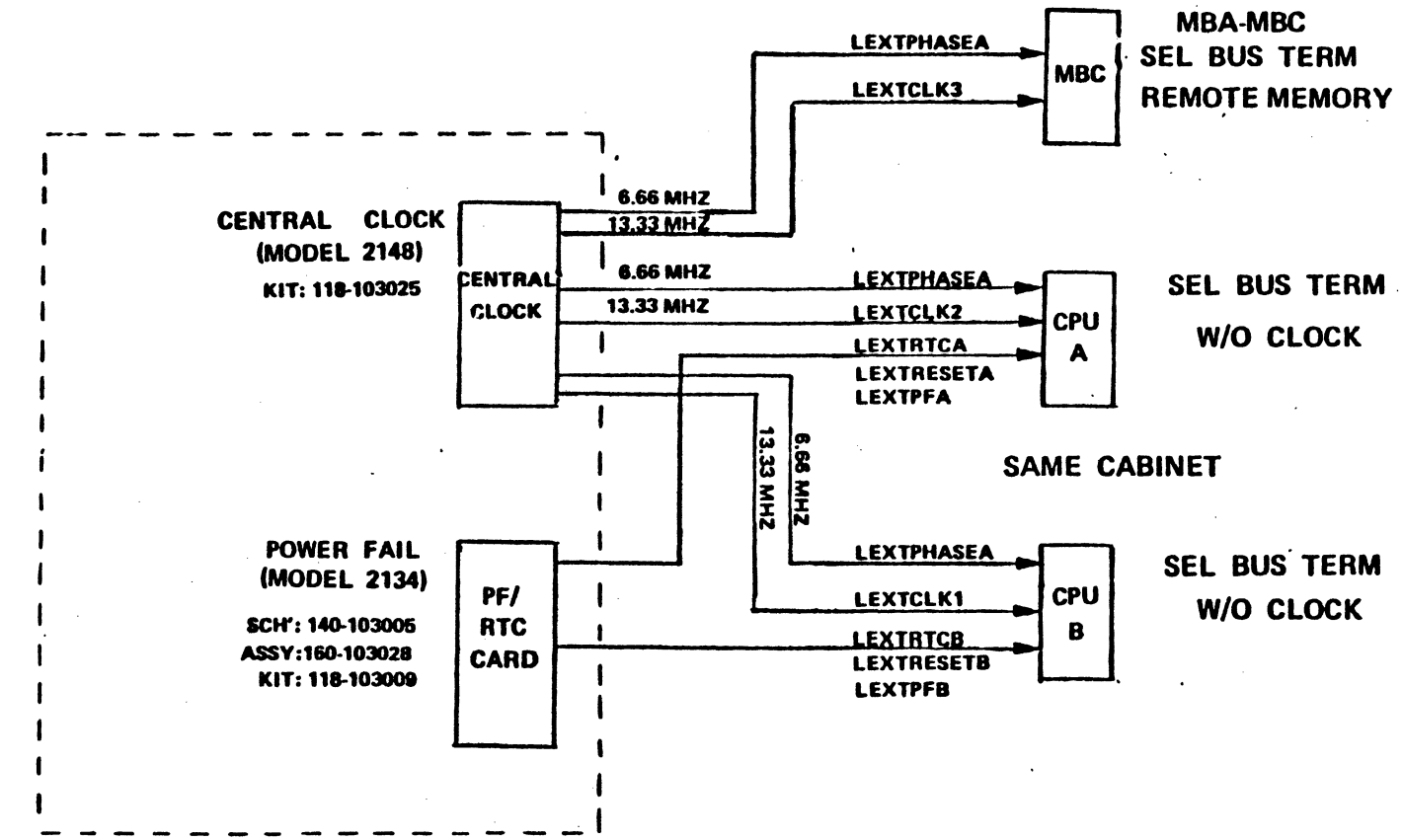
| ENTER: | LOCATION | DATA |
|---|---|---|
| | 0000 | A3881003 |
| | 0004 | EC000000 |

SYSTEM RESET AND RUN. MONITOR LOCATION 1000 (HEX), WITH EXTENDED FUNCTION 1. THE "B" DISPLAY SHOULD BE INCREMENTING. IT WILL PROBABLY LOOK ASYNCHRONOUS BECAUSE IT IS ONLY DISPLAYED EVERY 20 MILLISECONDS.

NOW SLOWLY TURN THE POTENTIOMETER ON THE DETECTOR CARD CLOCKWISE UNTIL THE PROGRAM STOPS. NOW TURN THE POT TWO TURNS COUNTERCLOCKWISE. YOU SHOULD NOW HAVE A SAFE OPERATING MARGIN.

ASSUMING YOUR MACHINE HAS CORE MEMORY OR MOS MEMORY W/BATTERY BACKUP, A GOOD CHECK IS TO POWER THE MACHINE "OFF" AND "ON" AND MAKE SURE THE CPU SCRATCHPAD IS STILL INTACT. IF NOT, ADJUST THE POT SLIGHTLY CLOCKWISE UNTIL THIS TEST PASSES.

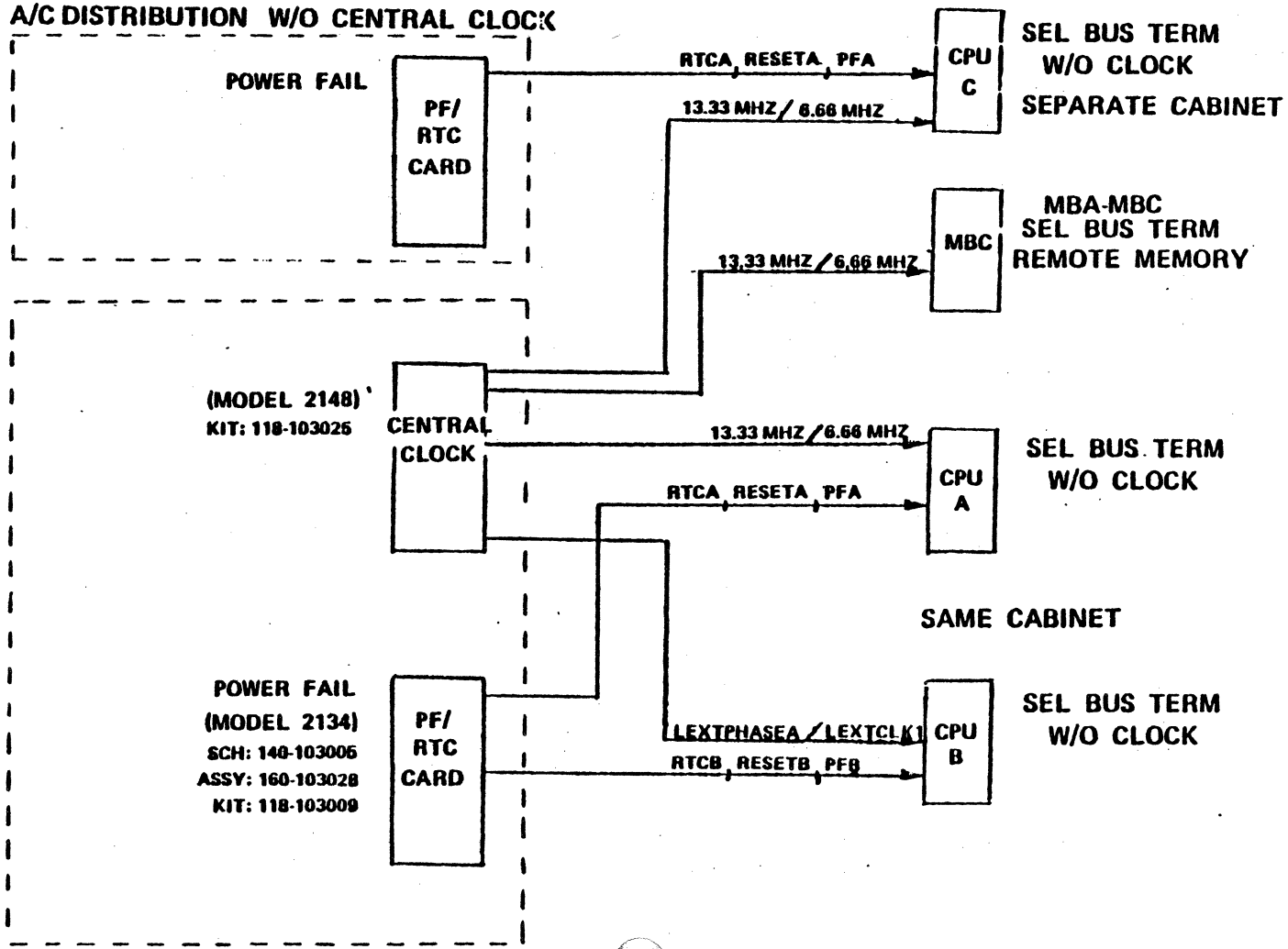
# CENTRAL CLOCK DISTRIBUTION
# 2 CPU'S/REMOTE MEMORY/1 AC DIST.



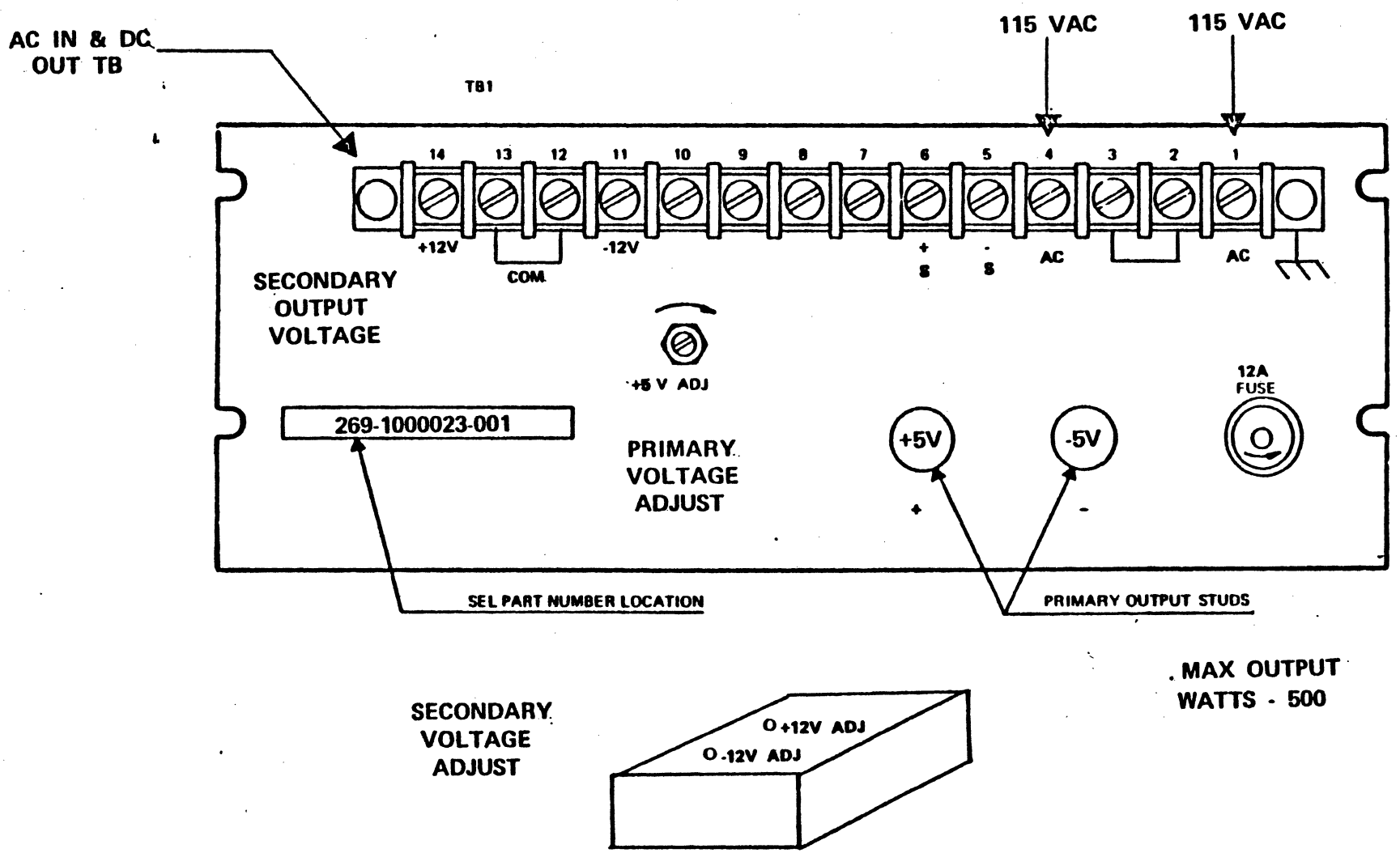

A/C DISTRIBUTION (MODEL 2195) W/CENTRAL CLOCK

ASSY: 105-103007
SCH: 140-103008

# CENTRAL CLOCK DISTRIBUTION
# 3 CPU'S/REMOTE MEMORY/2 AC DIST.

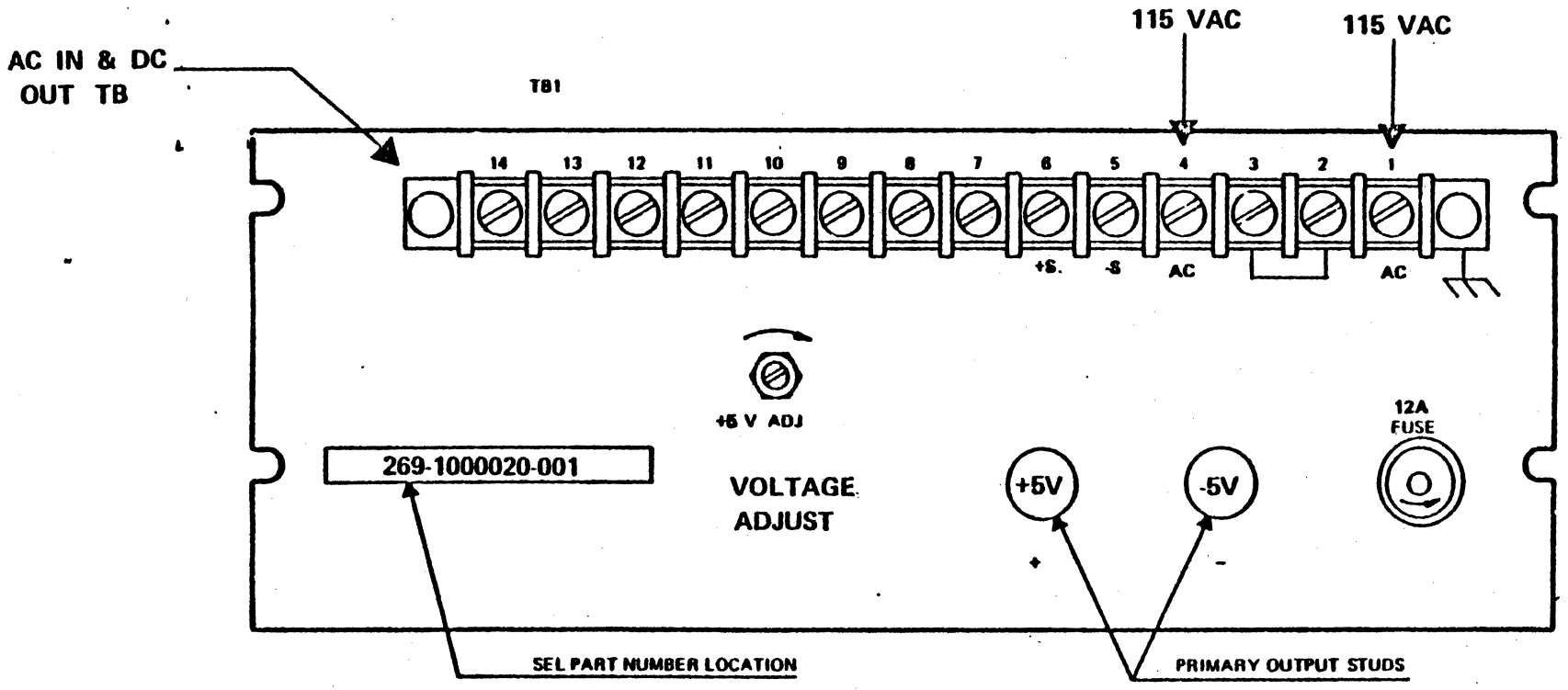


# LOGIC POWER SUPPLY MODEL # 2190




ADJUSTMENT NOTES: ALL VOLTAGES MUST BE ADJUSTED ON MEM/LOGIC CHASSIS 2/10 VOLT GREATER DUE TO CABLE/WIRING LOSSES. I.E.: +5 → +5.2 VDC. TTL IC CIRCUITS MUST HAVE +5V -.05 +.4 VDC. CHECK IC CHIPS ON CIRCUIT CARDS AFTER ADJUSTMENT ON CHASSIS'.

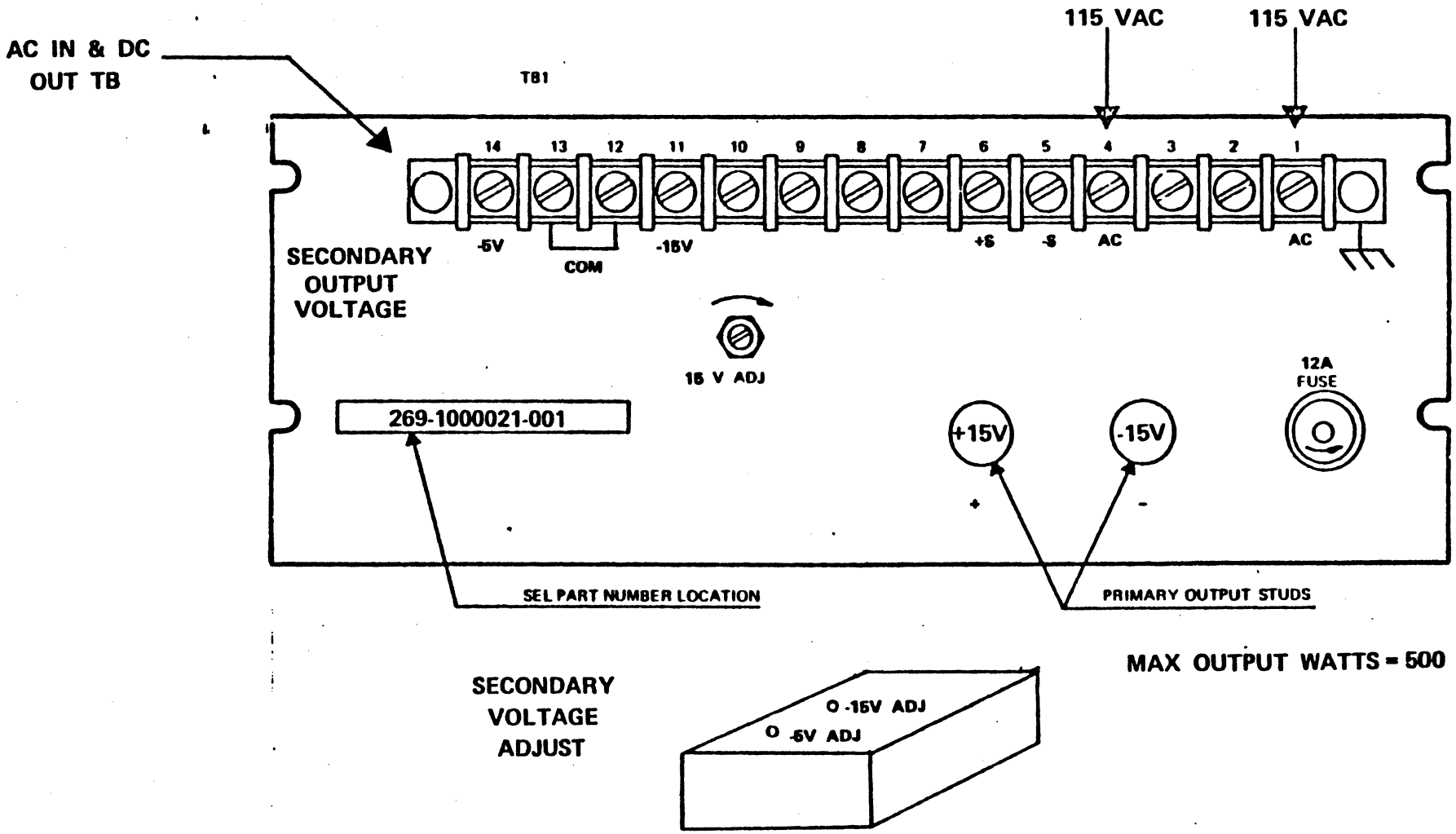
# MEMORY POWER SUPPLY MODEL # 2191




MAX OUTPUT WATTS = 500

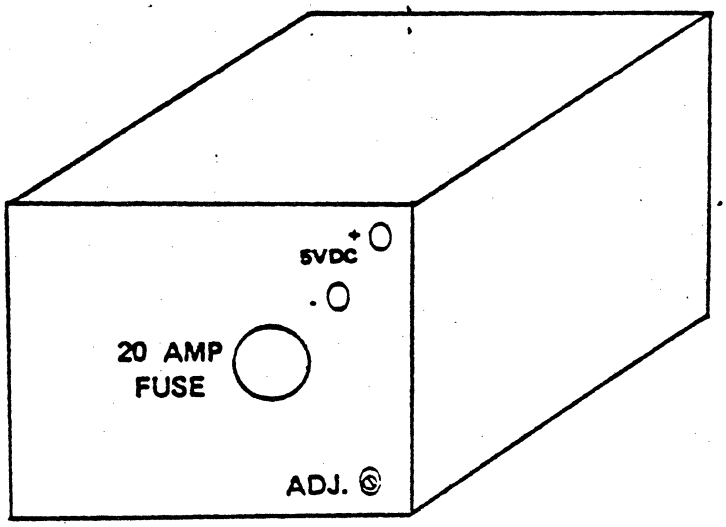
ADJUSTMENT NOTES: ALL VOLTAGES MUST BE ADJUSTED ON MEM/LOGIC CHASSIS 2/10 VOLT GREATER, DUE TO CABLE/WIRING LOSSES. I.E.: +5→+5.2 VDC. TTL IC CIRCUITS MUST HAVE ±5V -.05 +.4 VDC. CHECK IC CHIPS ON CIRCUIT CARDS AFTER ADJUSTMENT ON CHASSIS'.

# MEMORY POWER SUPPLY MODEL # 219



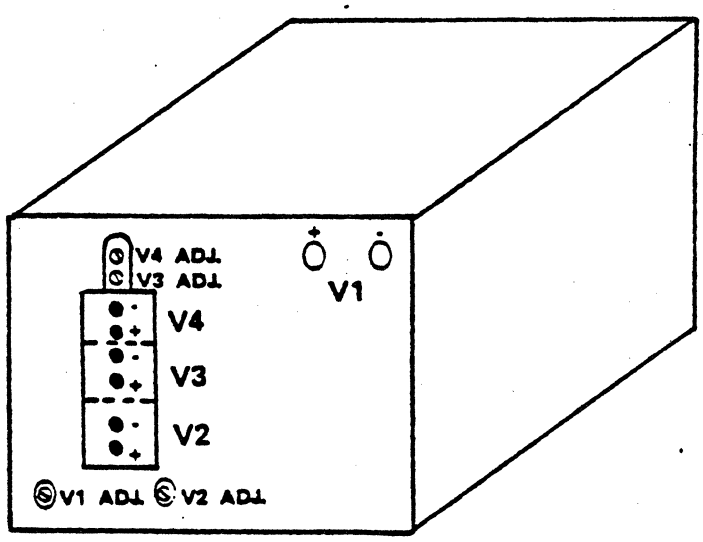

ADJUSTMENT NOTES: ALL VOLTAGES MUST BE ADJUSTED ON MEM/LOGIC CHASSIS 2/10 VOLT GREATER DUE TO CABLE/WIRING LOSSES. I.E.: +15 → +15.2 VDC. TTL IC CIRCUITS MUST HAVE ±5V -.05 +.4 VDC. CHECK IC CHIPS ON CIRCUIT CARDS AFTER ADJUSTMENT ON CHASSIS.

# HIGH PERFORMANCE CHASSIS P.S.




THIS IS THE POWERTEC MODEL 9N5-150-17A. IT SUPPLIES 5 VOLTS @ 150 AMPS TO ONE HALF OF A SIXTEEN SLOT MEMORY CHASSIS. IT HAS AN EXTERNAL FUSE'.

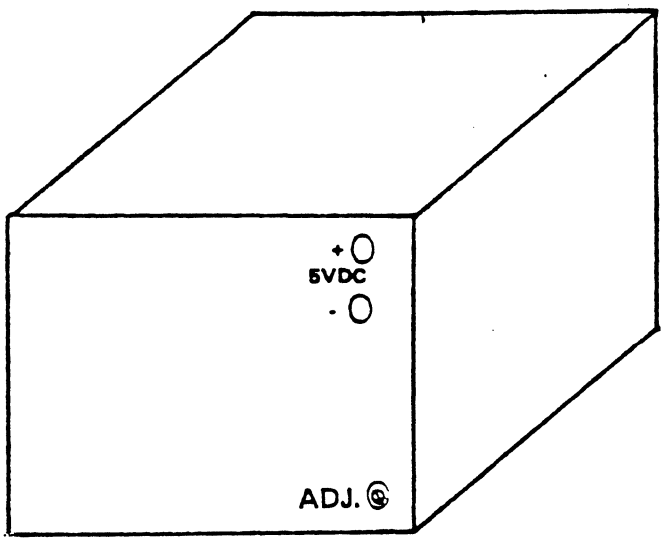
---




V1 — +5VDC @ 150 AMPS
V2 — +15VDC @ 10 AMPS
V3 — -15VDC @ 5 AMPS
V4 — -5VDC @ 5 AMPS
(750 WATTS TOTAL).

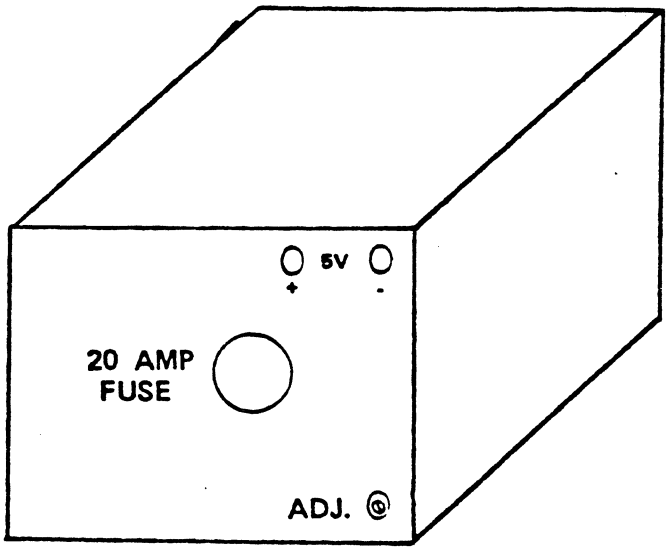
THIS IS THE LH MODEL MM44. IT SUPPLIES MULTI-VOLTAGES AS LISTED ABOVE TO A SIXTEEN SLOT MEMORY CHASSIS'. SUPPLIES +5VDC TO LOWER HALF OF CHASSIS. THIS SUPPLY ALSO FURNISHES + AND -15 VDC TO THE LOGIC CHASSIS(s).

# HIGH PERFORMANCE CHASSIS P.S.




THIS IS THE LH MODEL SM71. IT SUPPLIES 5 VOLTS @ 150 AMPS TO ONE HALF OF A LOGIC CHASSIS OR TO AN EIGHT SLOT MEMORY CHASSIS. NO EXTERNAL FUSE.

NOTE: LH MODEL SM11 LOOKS IDENTICAL BUT HAS 200 AMP CAPABILITY.




THIS IS THE POWER MATE MODEL SWA-5K-P2838. IT SUPPLIES 5 VOLTS @ 150 AMPS TO ONE HALF OF A LOGIC CHASSIS OR TO AN EIGHT SLOT MEMORY CHASSIS. IT HAS AN EXTERNAL FUSE.

# CONTROL PANEL FAMILIARIZATION WORKSESSION

## Control Panel Familiarization Worksession

**Purpose:** The purpose of this lab exercise is to familiarize the student with procedures necessary in loading, executing, and interpreting the results of machine language programs using the System Control Panel.

**References:** SEL 32 Reference Manual: Section 7

**Equipment:** SEL 32 Computer with System Control Panel and 8 K minimum core.

**Introduction:** The architecture of the SEL 32 family of computers is such that failure conditions can often be detected through simple programs written in machine code and entered through the System Control Panel. One can also use the System Control Panel for trouble-shooting using its "Extended Function" feature. Although the objective of this lab exercise is familiarization of the Control Panel, the procedure used below might be used in the field for troubleshooting.

**Instructions:** Follow the procedure, answering questions as asked. References will be supplied and the instructor is available to answer your questions.

### System Reset Check

System Reset causes all operations of the computer to cease and clears all registers, etc. to zero. System Reset is also sent to the IOM's to clear them and terminate all activity. Upon receiving System Reset, the TLC issues a "carriage return" character to the Teletype and the System Control Panel accesses and displays memory location zero.

Is System in halt mode?

Procedure: DEPRESS AND RELEASE SYSTEM RESET.

1. Did the teletype "carriage return"? _____
2. Are the System Control Panel PSW and instruction indicators lit? _____

If you answer NO to any of the above questions, System Reset is failing.

### Keyboard Mode

The operator must signal the SCPI that he wants to communicate with the system. He does so by depressing and releasing KEYBOARD. The SCPI is testing for this signal and responds by illuminating the KEYBOARD indicator.

Procedure:
1. Depress and release KEYBOARD.
2. Observe KEYBOARD indicator lit.

The SCPI is now ready to receive hex data into the 'B' display or receive a function key.

## Control Panel Read/Write

In order to enter machine code programs into the SEL 32 memory, you must be able to Read and Write to memory from the SCP.

This procedure is relatively simple. In this exercise, we will read location 1000, write to location 1000 and re-read location 1000 to verify the data has changed.

Procedure:

1. System Reset
2. Press and release Keyboard.
3. Observe Keyboard indicator lit.
4. Press and release Hex Key #1
5. Observe a hex 1 in 'B' display.
6. Press and release Hex Key #0 three (3) times.
7. Observe hex 1000 in 'B' display. (The hex characters are shifted in from the right.)
8. Press and release WRITE/X.
9. Press and release HEX KEY 8/MA.
10. Observe hex 1000 in 'A' display.
11. Observe Memory address indicator lit.

Note: You have just selected Memory address 1000. Now we want to read that location.

12. Press and release Read/X.
13. Press and release hex key C/MD.
14. Observe 'B' display. Observe memory data indicator lit.

Note: The B display now contains the data in memory location 1000. What is the value of the data? _____

We will now modify location 1000.

15. Press/release: Keyboard
16. Enter X'12345678' in B display.
17. Press/release: Write/MD

Note: You have just written X'12345678' in location 1000.

Repeat steps 1 through 14 to verify X'12345678' is loaded in location 1000. What is now in location 1000? _____

Let's summarize the procedure for reading memory:

1. Keyboard
2. ENTER HEX LOCATION IN 'B' display.
3. WRITE-MEMORY ADDRESS, READ-MEMORY data.

Let's summarize the procedure for writing to a memory location.

1. Keyboard
2. Enter location in 'B' display.
3. Write-memory address.
4. Keyboard
5. Enter new data in 'B' display.
6. Write-memory data.

Try these a few times to"get the hang of it."

## Entering Machine Language Programs

Entering programs into memory is the same as entering data as we did in the Control Panel Read/Write lab. As a matter of fact, you can use the same procedures! But there is a better way. Programs usually occupy sequential locations in memory. To take advantage of this, there are two special Keys on the Panel: INCREMENT'A' AND READ and WRITE AND INCREMENT 'A'. Here is how to use them.

Procedure: Let's put the following sequential data in memory:

| Locations | Data |
|---|---|
| 00000 | 11111111 |
| 00004 | 22222222 |
| 00008 | 33333333 |
| 0000C | 44444444 |

Note: Observe 'A' & 'B' displays after each operation.

1. SYSTEM RESET
2. KB - enter X'11111111'
3. WRT & INC 'A'

Observe PSW = 4. This is the next location in memory.

4. KB - enter X'22222222'
5. WRT & INC 'A'

Observe PSW = 8

6. KB - enter X'33333333'
7. WRT & INC 'A'

Observe PSW = C

8. KB - enter X'44444444'
9. WRT & INC 'A'

Observe PSW = 10

You have just entered the data in memory starting at location zero. Now let's read it back.

1. SYSTEM RESET

Observe PSW = 0, INSTRUCTION = X'11111111'

2. INC 'A' & RD

Observe PSW = 4, INSTRUCTION = X'22222222'

3. INC 'A' & RD

Observe PSW = 8, INSTRUCTION = X'33333333'

4. INC 'A' & RD

Observe PSW = C, INSTRUCTION = X'44444444'

QUESTION: What is the advantage of using WRT & INC'A' and INC 'A' & RD keys?

TO SAVE TIME

## Executing Programs

There are two ways of executing a program. One way is to execute the instructions one by one. This may be done using INSTRUCTION STEP. The other way is to execute the instructions until a HALT occurs either by program control or operator control. Program Control is done by the HALT instruction. Operator Control is done by the HALT key. Let's see how this works.

Procedure: Enter the following program.

| Location | Hex | Instruction |
|---|---|---|
| 0 | '00020002' | NOP-NOP |
| 4 | 'AC000100' | LW R0,100 |
| 8 | 'D4000200' | STW R0,200 |
| C | 'AC000300' | LW R0,300 |
| 10 | 'D4000400' | STW R0,400 |
| 14 | 'AC000500' | LW R0,500 |
| 18 | 'D4000600' | STW R0,600 |
| 1C | 'EC000000' | BU 0 |

### Procedure for Using Instruction Step

1. SYSTEM RESET
2. Press and release INSTRUCTION STEP.

Repeat #2 each time observing the PSW and INSTRUCTION. Note that the HALT indicator is lit.

### Procedure for Executing the Program at Computer Speed

1. System Reset
2. Press/release RUN

Observe RUN indicator lit. Observe 'A' and 'B' displays go [redacted]. This is normal. The program is running.

3. Press/release HALT.

Observe HALT indicator lit. Observe 'A' display and 'B' display shows next instruction to be executed.

## ERROR INDICATOR (Parallel Panel)

B Display Error Code Definition

1 Hex Keyboard or Function Keyboard Change Indicator did not reset
2 No Hex or Function Keyboard Key detected
3 No response from memory
4 Non-present memory
5 Parity error in memory
6 Write/Read Compare error in memory
7 SEL Bus Communication Error

## OPERATOR FAULT INDICATOR (Parallel Panel)

B Display Operator Fault Definitions

1 Operator Sequence Error
2 Operation Not Allowed - Run on Lock Restrictions
3 Invalid Operand Source or Destination
4 A Display Not Valid for Operation to be performed
5 Invalid Extended Function
6 Special Extended Function Not Enabled
7 Multiple Hex or Function Keyboard Keys detected

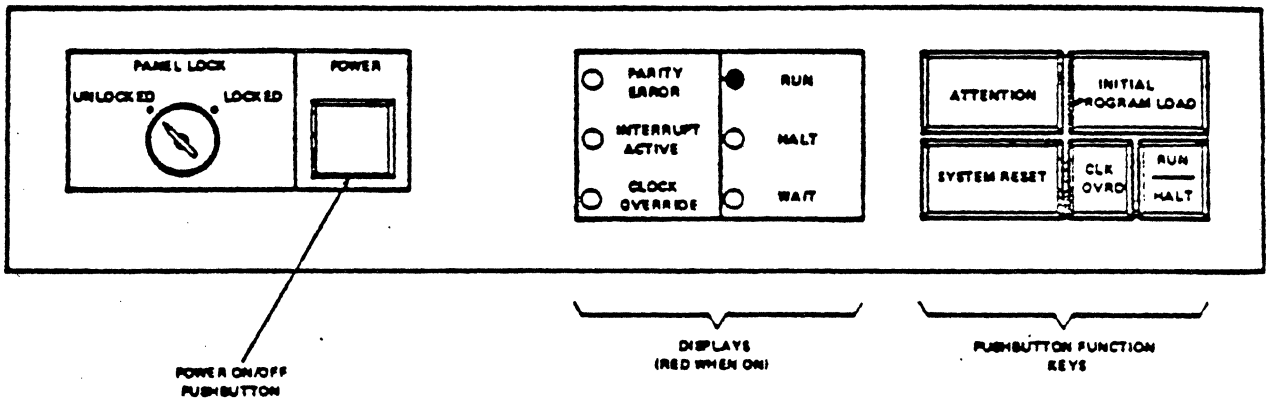



Figure 7-1. Turnkey Panel Diagram

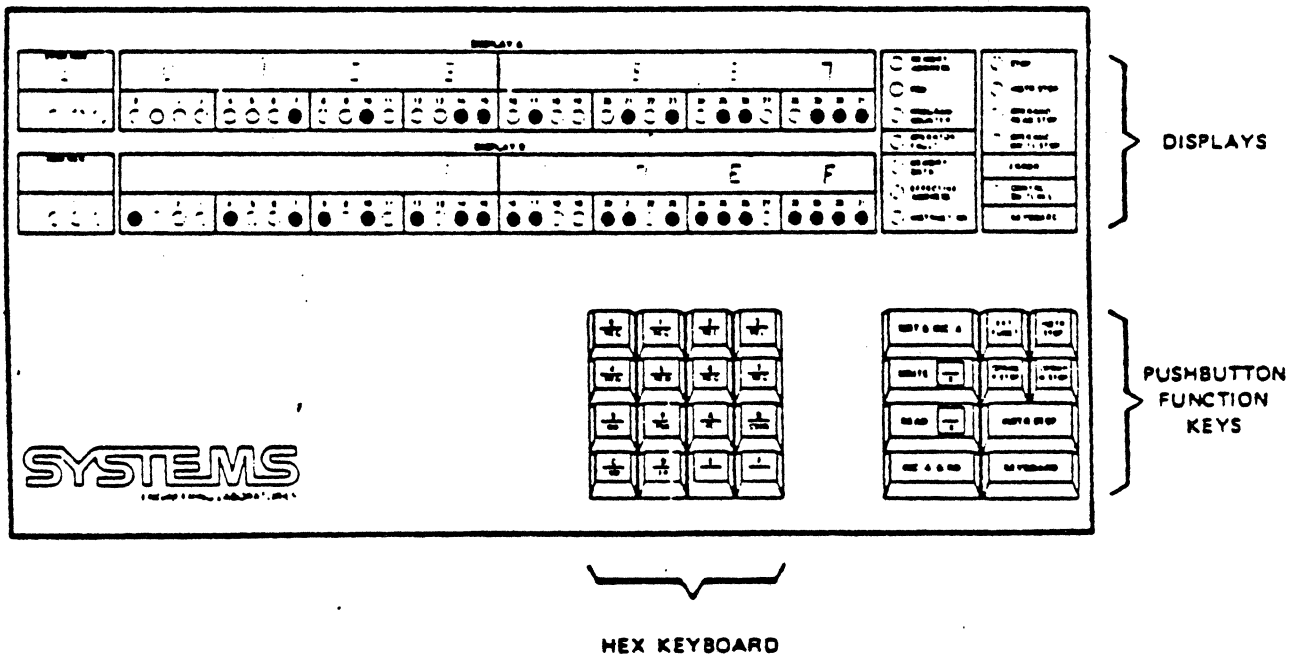
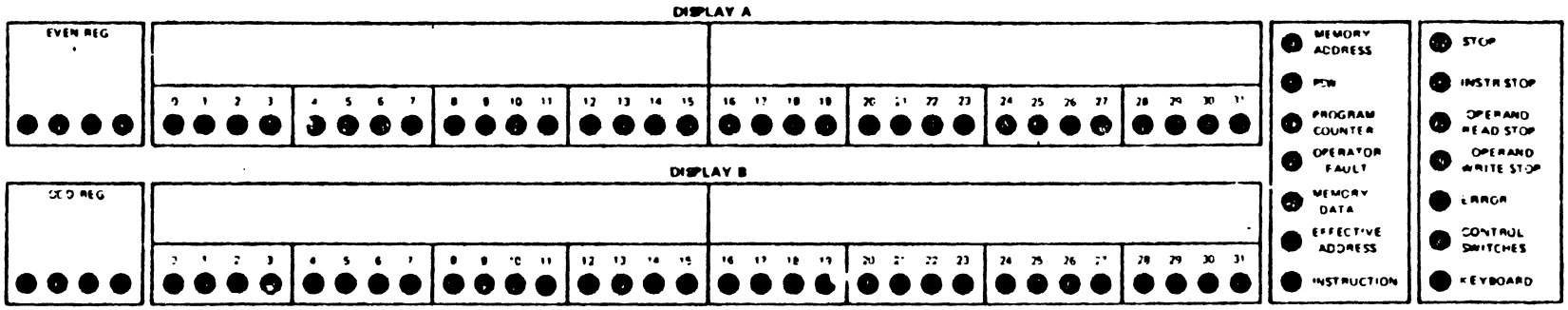
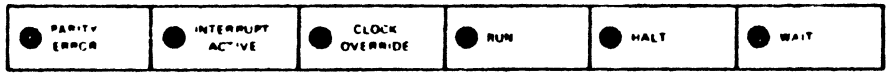


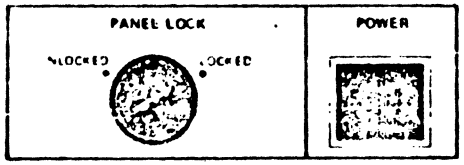

Figure 7-2. System Control Panel (Parallel Operation) Diagram

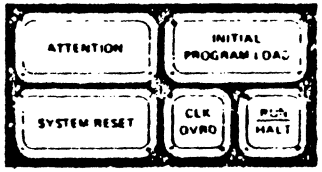
# 32/70 SERIES CONTROL PANEL

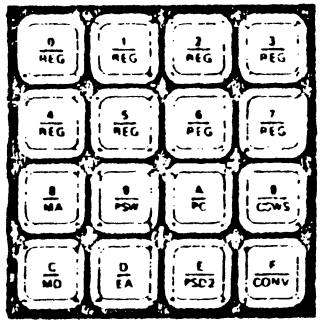


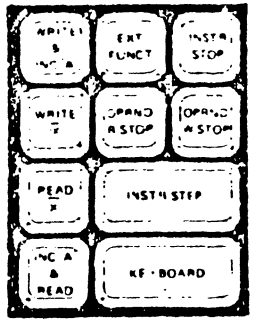

SYSTEMS
ENGINEERING LABORATORIES



ERROR INDICATOR
(Serial Panel)

B Display Error Code Definition

1 CPU UART Error
2 Transmission Error Other Than UART
3 No Response from Memory
4 Non-Present Memory
5 Parity Error in Memory
6 Write/Read Compare Error in Memory
7 Bus Interchange or Memory is Malfunctioning

OPERATOR FAULT INDICATOR
(Serial Panel)

B Display Operator Fault Definitions

1 Does Not Apply to SCP
2 Operation Not Allowed - Run or Lock Restrictions
3 Invalid Operand Source or Destination
4 A Display Not Valid for Operation to be Performed
5 Invalid Extended Function
6 Special Extended Function Not Enabled
7 Does Not Apply to SCP

## OPERATION WITHOUT SCPI

1. In the event that the SCPI has been determined to be causing problems, it is possible to continue operation without it until another is available.

2. The necessary steps to remove the SCPI are listed below:
   a) Remove SCPI Controller
   b) Remove SCPI IOX
   c) Remove 20 pin cable between Control Panel and Turnkey Panel
   d) Jumper location E11 on TLC from '78 to '01.
   e) Modify I.C.L. Deck for the TTY, LP and CR from 78 to 01.
   f) Replace Jumper 22 on Clock Card.

3. This procedure will default the initial input device (C.R.) to 01.

## Extended Functions

Extended functions are available to the user through the System Control Panel. They take advantage of the fact that the System Control Panel Interface is an IOM and has the capability of accessing memory and performing arithmetic operations independent of the CPU.

## Lamp Test

An extended function may be used to check the Control Panel indicators.

Procedure:
1. Depress and release KEYBOARD.
2. Observe KEYBOARD indicator lit.
3. Depress and release EXTENDED FUNCTION.
4. Observe KEYBOARD indicator not lit.
5. Depress and release HEX KEY #4.
6. Observe all indicators lit on SCP.

## Memory Parity Check

Extended function #3 may be used as a quick check for memory integrity. This extended function causes the SCPI to write and read all ones and zeroes to all memory locations. Any parity error will cause the routine to halt with an error code '5' in the 'B' display.

Procedure:
1. KEYBOARD
2. EXTENDED FUNCTION
3. HEX KEY #3

Observe: memory address in the 'A' display incrementing.
Observe: memory data in 'B' display either all ones or all zeroes.

Note: This function destroys all data in memory.
Listed below are the extended functions available.

## Control Panel Extended Functions

Extended Function 0 — Establish Upper Bound
The contents of the B display register is saved as the upper bound for the Fill Memory Operation (Extended Function 2).

* Extended Function 1 — Monitor Memory Location
The memory word specified by the Memory Address in the A display is read every 20 milliseconds and loaded into the B display.

Extended Function 2 — Fill Memory
Using the upper-bound established by Extended Function 0, and the lower limit established by the current Memory Address in the A display, the contents of the B display are stored in all memory locations L, where lower limit $\leq L <$ upper bound.

Extended Function 3 — Memory Write/Read Test
The test pattern all 1's, then all 0's are written into each memory location, then read back and compared with the pattern written. This continues until non-present memory is detected (assumed upper bound) or a memory read error is detected. If non-present memory occurs, the test is restarted with the alternate pattern.

* Extended Function 4 — Lamp Test
The lines for all indicators on the System Control Panel are driven and all indicators should be illuminated.

Extended Function 5 — Memory Read Test
Memory locations starting with location 0 are read and checked for parity errors. The location is displayed in the A display, the data retrieved from that location in the B display, and the parity bits in the register field associated with the B display. This test is restarted at location 0 when non-present memory is detected. If a parity error is detected, the sequencing through memory stops.

Extended Function 8 — B → R, B unchanged
The contents of the B display register are copied and saved in an internal register for hexadecimal arithmetic computation.

Extended Function 9 — A → B, A unchanged
The contents of the A display register are copied tothe B display register.

Extended Function A — R + B → B, R unchanged
The contents of the R register is added to the contents of the B display register and returned to the B display.

Extended Function B — R - B → B, R unchanged
The contents of the B display register is subtracted from the contents of the R register and returned to the B display.

Extended Function C — Negate B
The contents of the B display register is negated (two's complement) and returned to the B display.

Extended Function D — B (Hex) → B (Decimal)
The contents of the B display (B ≤ 000FFFFF) are converted to decimal and returned to the B display.

Extended Functio.. E — B (Decimal) → B (Hex)
The contents of the B display (B ≤ 00099999) are converted to hexadecimal and returned to the B display.

* Only Extended Functions on Serial Panel

Course 340 - 32 Architecture

System Characteristics - Worksession

## Objectives

Upon completion of this worksession, you will be able to:

1. State the function of the main components in the Series 32/70 System.
2. List the main components of the system.

## Reference Material

1. Technical Manual, 32/70 Series Computer 303-320070
2. Reference Manual, System 32/70 Series 301-320070
3. 340 Workbook

# SYSTEM CHARACTERISTICS

1. The CPU is composed of 3 boards listed as A, B, and C, what are their functions? A – MSB ALU
   B – LSB ALU
   C – [illegible]

2. The SEL Bus has a total of how many lines? 184 lines

3. What is the function of the SEL Bus? [illegible] Interface
   IPU – CPU – Memory – RTOM – TLC – MBC

4. What module controls access to the Memory Bus?
   MBC Memory Bus Controller

5. What 3 basic functions does the RTOM provide?
   16 Interrupts
   32 [illegible] Interval Timer
   60/Bank RTC

6. What module provides control for the console devices?
   TLC Terminal Console Card Handler

7. What is the function of any IOM?
   – Communication between peripherals
   – Generates service Interrupts

8. What is the memory cycle time of an 8KW core module?
   600 nSec

9. How many memory modules can one MBC control?
   16 memory modules

10. What are the major differences between a Serial Control Panel and the optional Parallel Panel?
    Keyboards are different
    Serial Panel 2 Ext FNX
    Parallel Panel 16 Ext FNX
    ↳ SCPI Card

# 32 ARCHITECTURE

## WORKSESSION #2

## SYSTEM CONFIGURATION

### A. Objective

This package is intended to aid you in learing to effectively use SEL documentation when troubleshooting maintenance problems in the SEL 32/7X System. Upon completion of this package you will be able to:

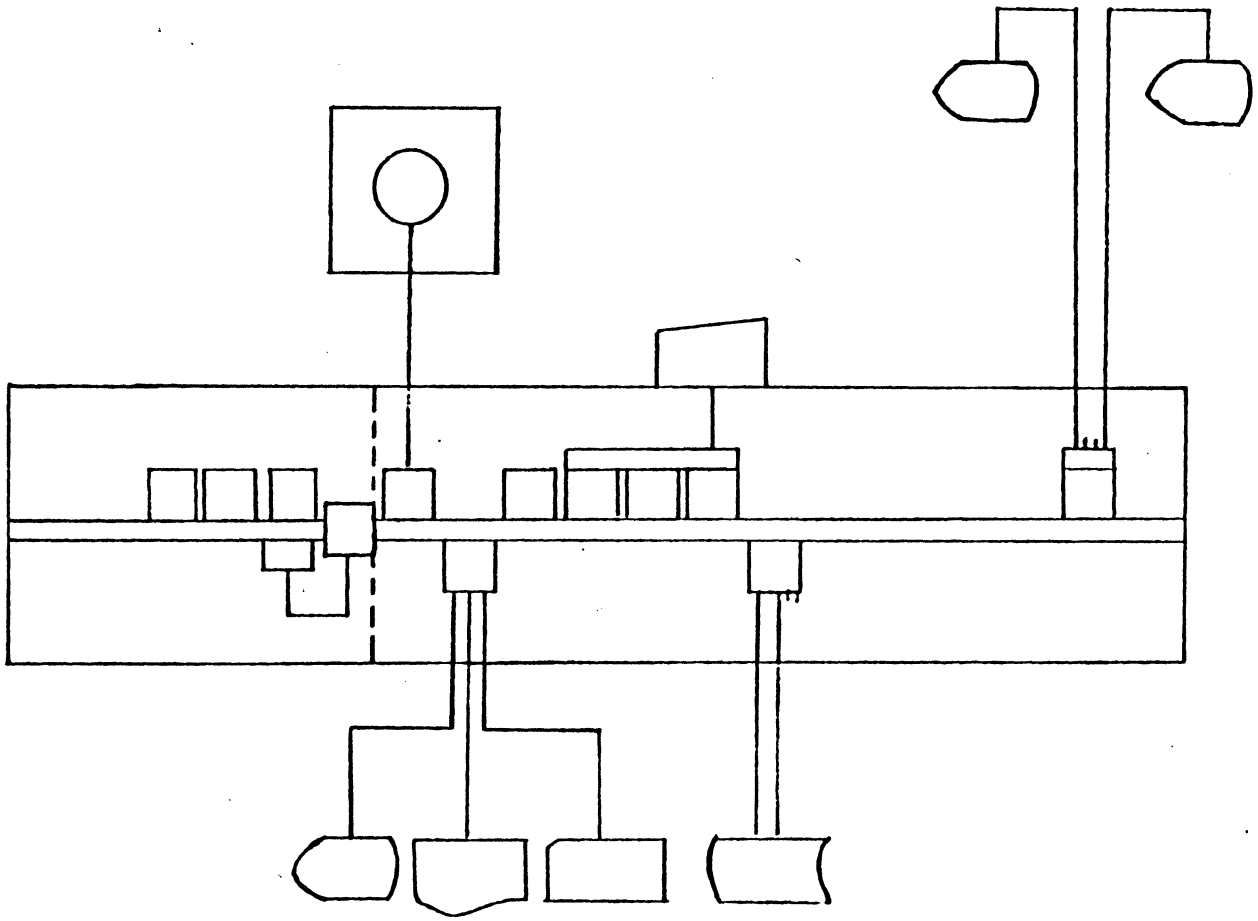
1. Read and understand a drawing of the SEL 32 System layout, and more specifically, recognize the module's name and abbreviation which make up the system.
2. Generate a system configuration listing given the model number, part number and part nomenclature.
3. Cable a system using your configuration listing.

### B. Reference Material

1. Technical Manual, 32/70 Series Computer 303-320070
2. Reference Manual, SYSTEMS 32/70 SERIES 301-320070
3. Drawings Manual handout

### C. Overview

1. Read pages 1-1 through 1-12 in the 32/70 Series Computer Technical Manual.
2. Included in this documentation package is a System layout drawing number 103-250405. Study this drawing, it will help you to further familiarize yourself with the components that comprise the system.



A. SEL BUS

B. MBC

C. MEMORY MODULE

D. MEMORY BUS

E. REFRESH MODULE

F. MAG TAPE

G. MAG TAPE CONT

H. TLC

I. CARD READER

J. LINE PRINTER

K. CRT (CONSOLE)

L. CRT/S

M. ADS

N. SCP

O. RTOM

P. CPU

Q. MHD

R. MHD CONT

# DAY 2

# SECTION 2

# INSTRUCTION SET

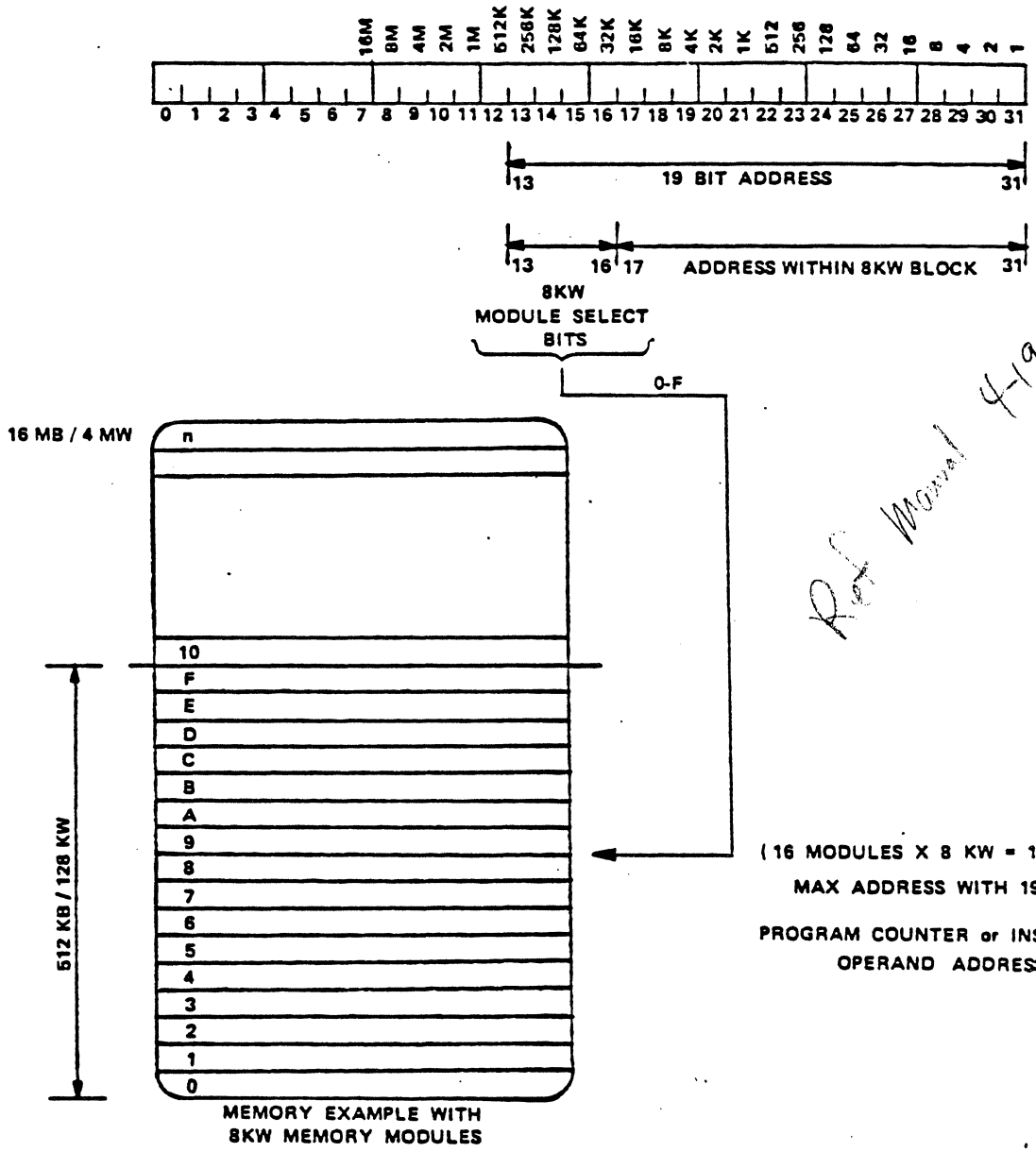
ENVIRONMENT

# OPERATING MODES

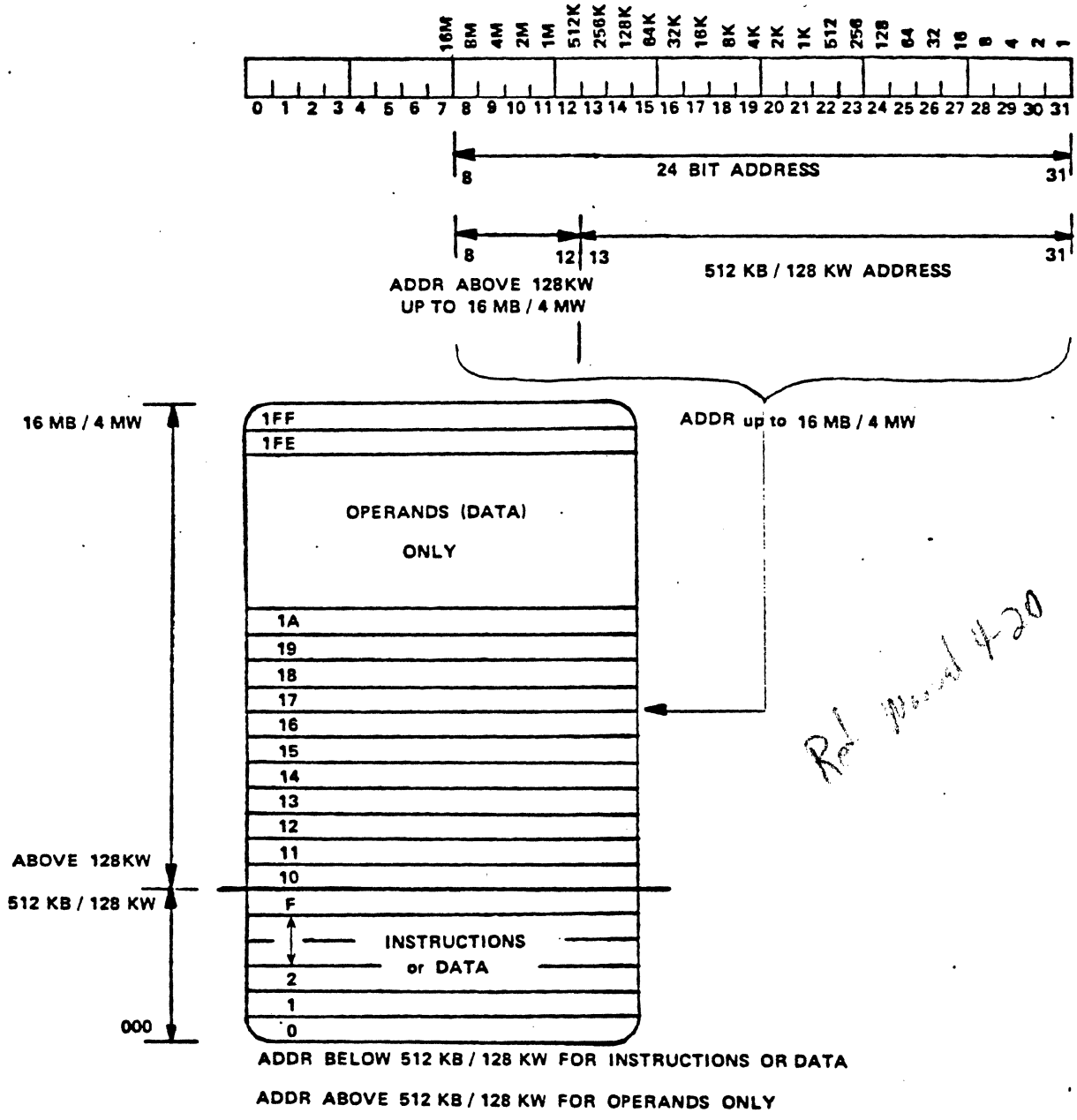
| | PSW | PSD | |
|---|---|---|---|
| | RTM | RTM Z.1 | MPX |
| PROGRAM STATUS: | WORD | DOUBLEWORD | DOUBLEWORD |
| # OF INST: | 161 | 161 + | 189 |
| INTEGRITY: | INT'S ON 1ST RTOM | TRAPS | TRAPS |
| MEM ADDR: | | | |
| NONMAPPED: | | | |
| NONEXTENDED: | 512 KB | 512 KB | 512 KB * |
| EXTENDED: | 16 MB | 16 MB | 16 MB * |
| MAPPED: | | | |
| NONEXTENDED: | NONE | NONE | 512 KB PER USER |
| EXTENDED: | NONE | NONE | 1 MB PER USER (TOTAL |
| CD I/O: | YES | YES | YES |
| ADDRESSING: | 512 KB | 512 KB | 512 KB |
| EXT I/O: | | | |
| (CLASS 'F'): | NO | YES | YES |
| ADDRESSING: | NONE | 16 MB | 16 MB |

* NO SOFTWARE SUPPORT

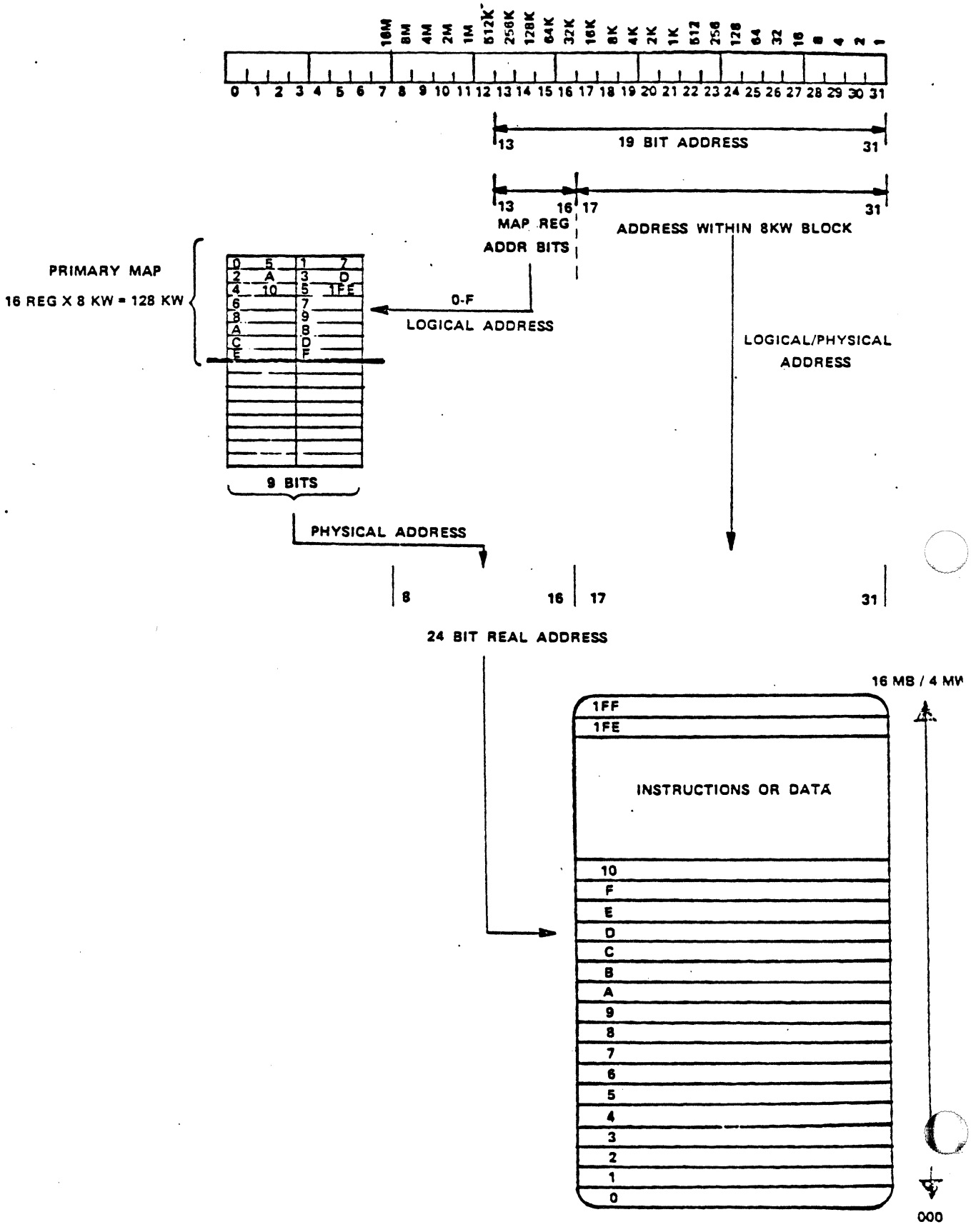
# 512 KB ADDRESSING MODE (NONEXTENDED)



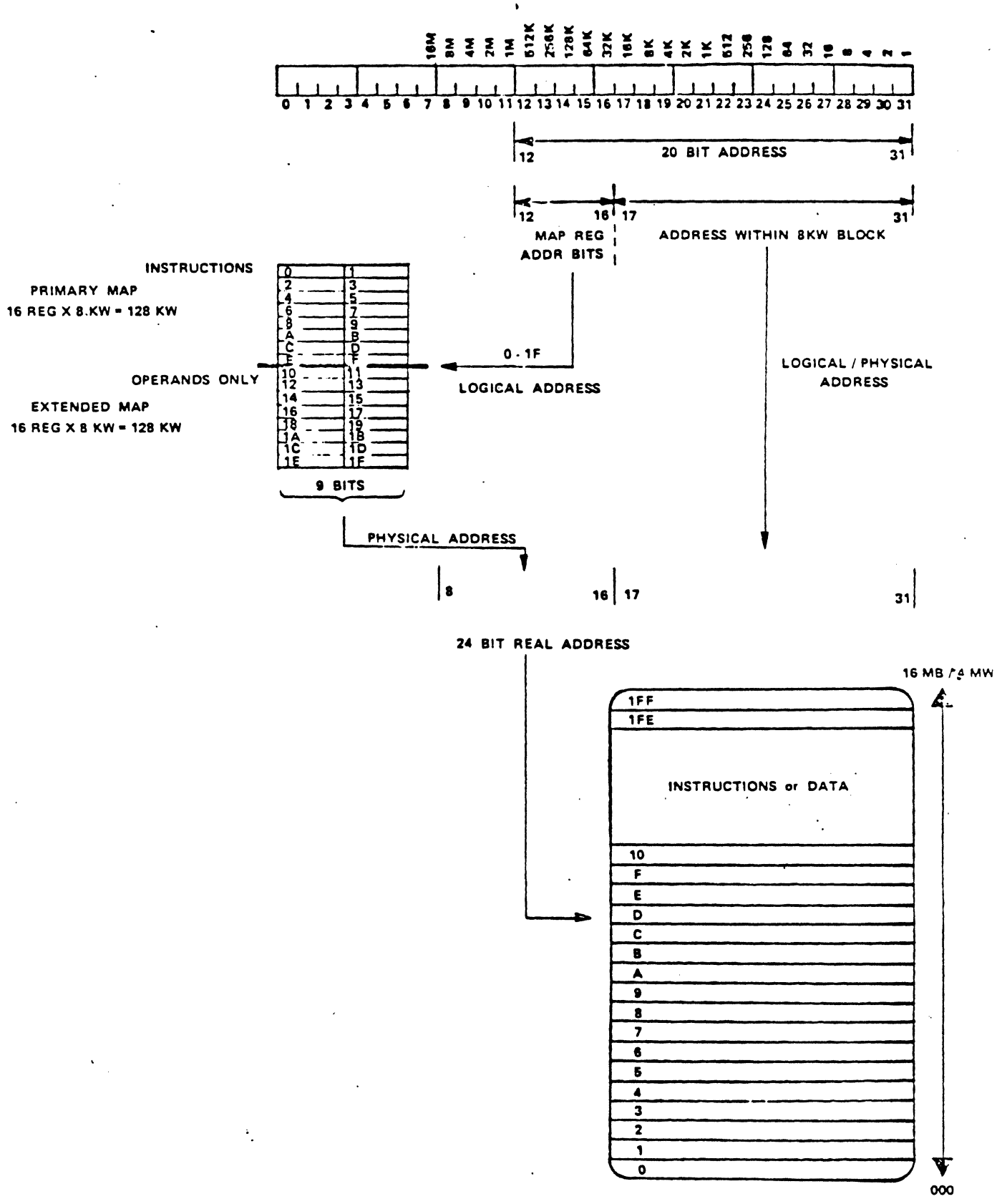
# 512 KB EXTENDED MODE



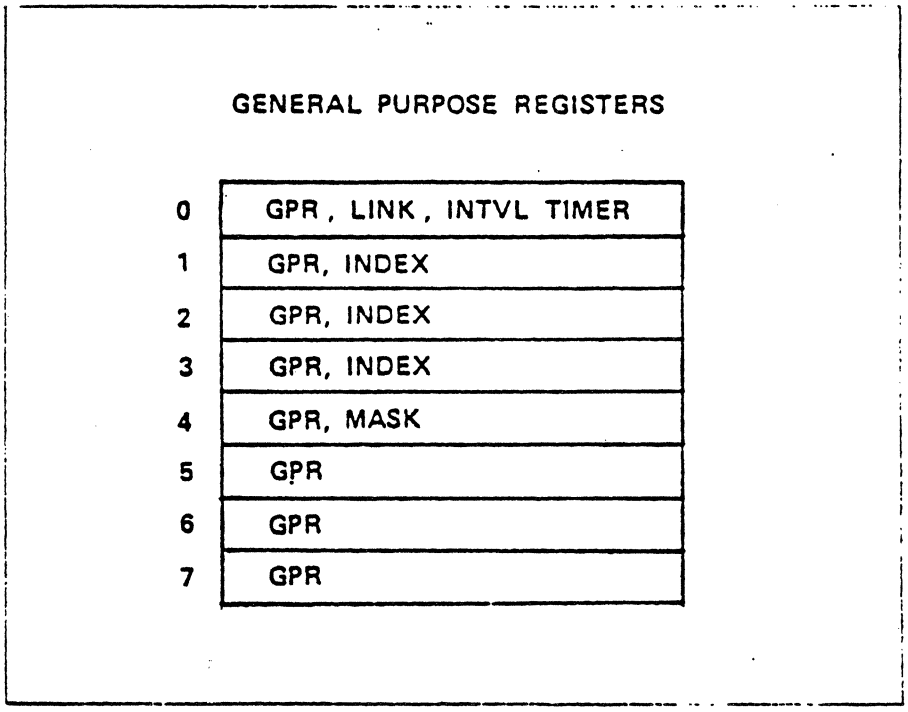
# 512 KB MAPPED MODE



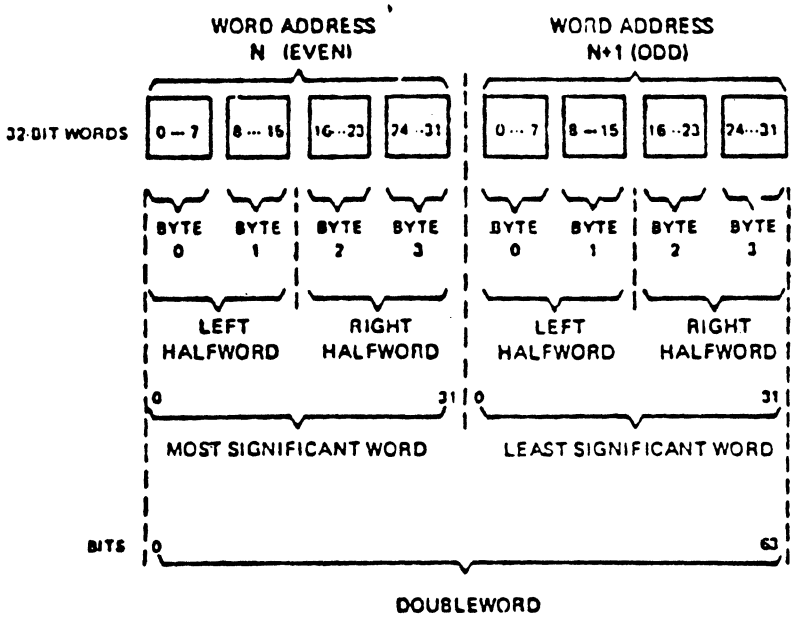
# MAPPED EXTENDED MODE

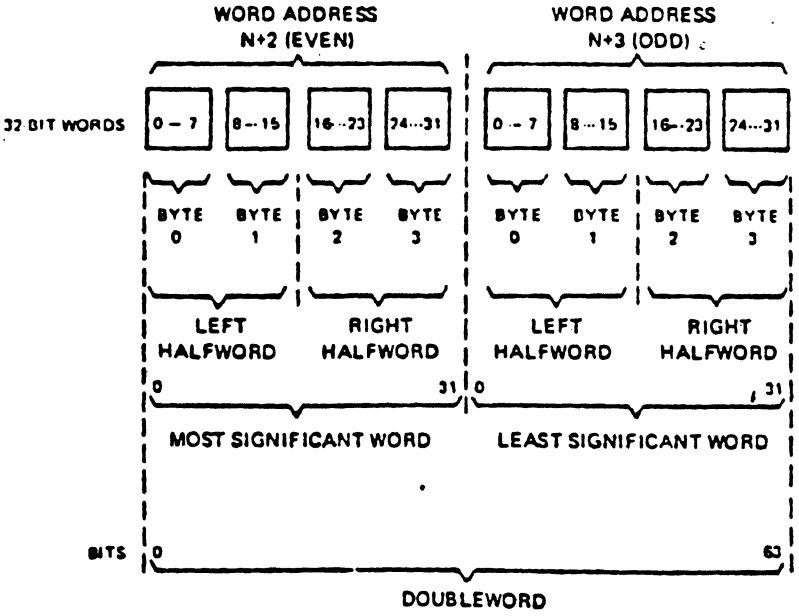


# CPU

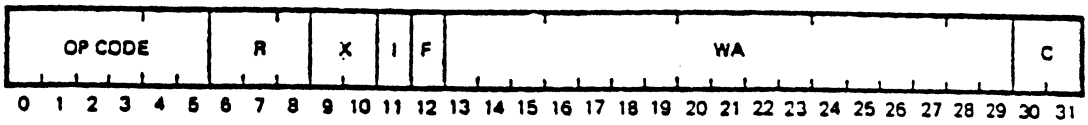

GENERAL PURPOSE REGISTERS
0
GPR , LINK , INTVL TIMER
1
GPR, INDEX
2
GPR, INDEX
3
GPR, INDEX
4
GPR, MASK
5
GPR
6
GPR
7
GPR

## INFORMATION BOUNDARIES IN MEMORY



## MEMORY REFERENCE INSTRUCTION FORMAT




**BITS 0-5** DEFINE THE OPERATION CODE.

**BITS 6-8** DESIGNATE A GENERAL PURPOSE REGISTER ADDRESS (0-7).

**BITS 9-10** DESIGNATE ONE OF THREE GENERAL PURPOSE REGISTERS TO BE USED AS AN INDEX REGISTER.

- X = 00 DESIGNATES THAT NO INDEXING OPERATION IS TO BE PERFORMED.
- X = 01 DESIGNATES THE USE OF R1 FOR INDEXING.
- X = 10 DESIGNATES THE USE OF R2 FOR INDEXING.
- X = 11 DESIGNATES THE USE OF R3 FOR INDEXING.

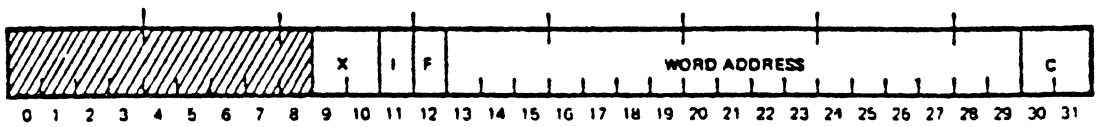
**BIT 11** DESIGNATES IF AN INDIRECT ADDRESSING OPERATION IS TO BE PERFORMED.

- I = 0 DESIGNATES THAT NO INDIRECT ADDRESSING OPERATION IS TO BE PERFORMED.
- I = 1 DESIGNATES THAT AN INDIRECT ADDRESSING OPERATION IS TO BE PERFORMED.

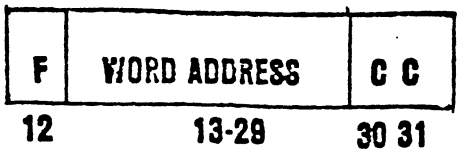
**BITS 12-31** SPECIFY THE ADDRESS OF THE OPERAND WHEN X AND I FIELDS ARE EQUAL TO ZERO.

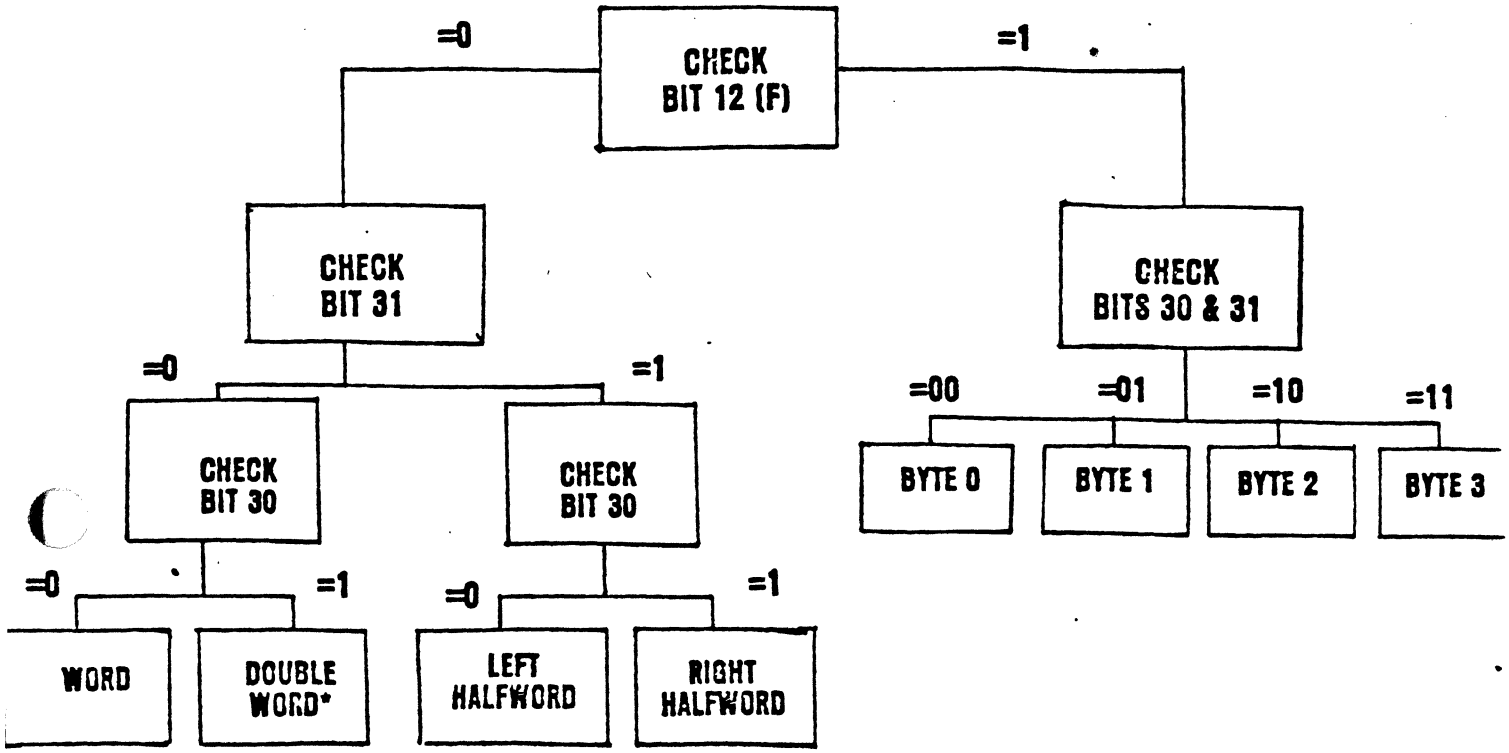
| F | C | Data Type |
|---|---|---|
| 0 | 00 | 32-bit word |
| 0 | 01 | 16-bit left halfword (bits 0-15) |
| 0 | 10 | 64-bit doubleword |
| 0 | 11 | 16-bit right halfword (bits 16-31) |
| 1 | 00 | byte 0 (bits 0-7) |
| 1 | 01 | byte 1 (bits 8-15) |
| 1 | 10 | byte 2 (bits 16-23) |
| 1 | 11 | byte 3 (bits 24-31) |

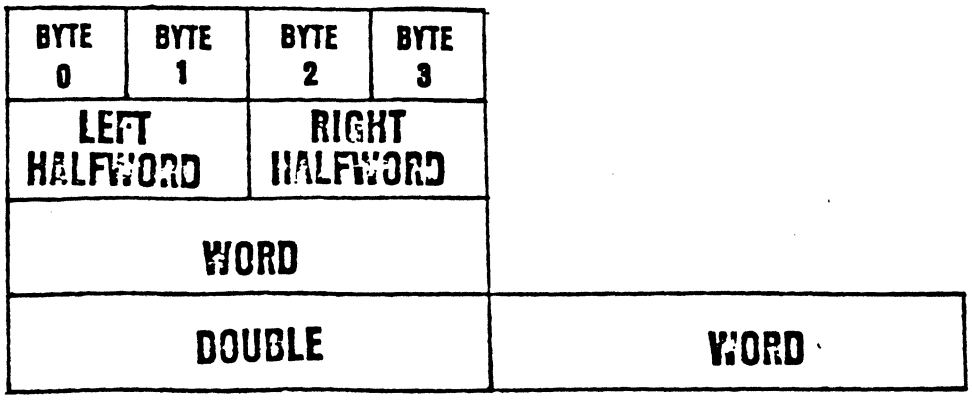
## INDIRECT ADDRESS FORMAT



# ADDRESS and TYPE DECODING TREE

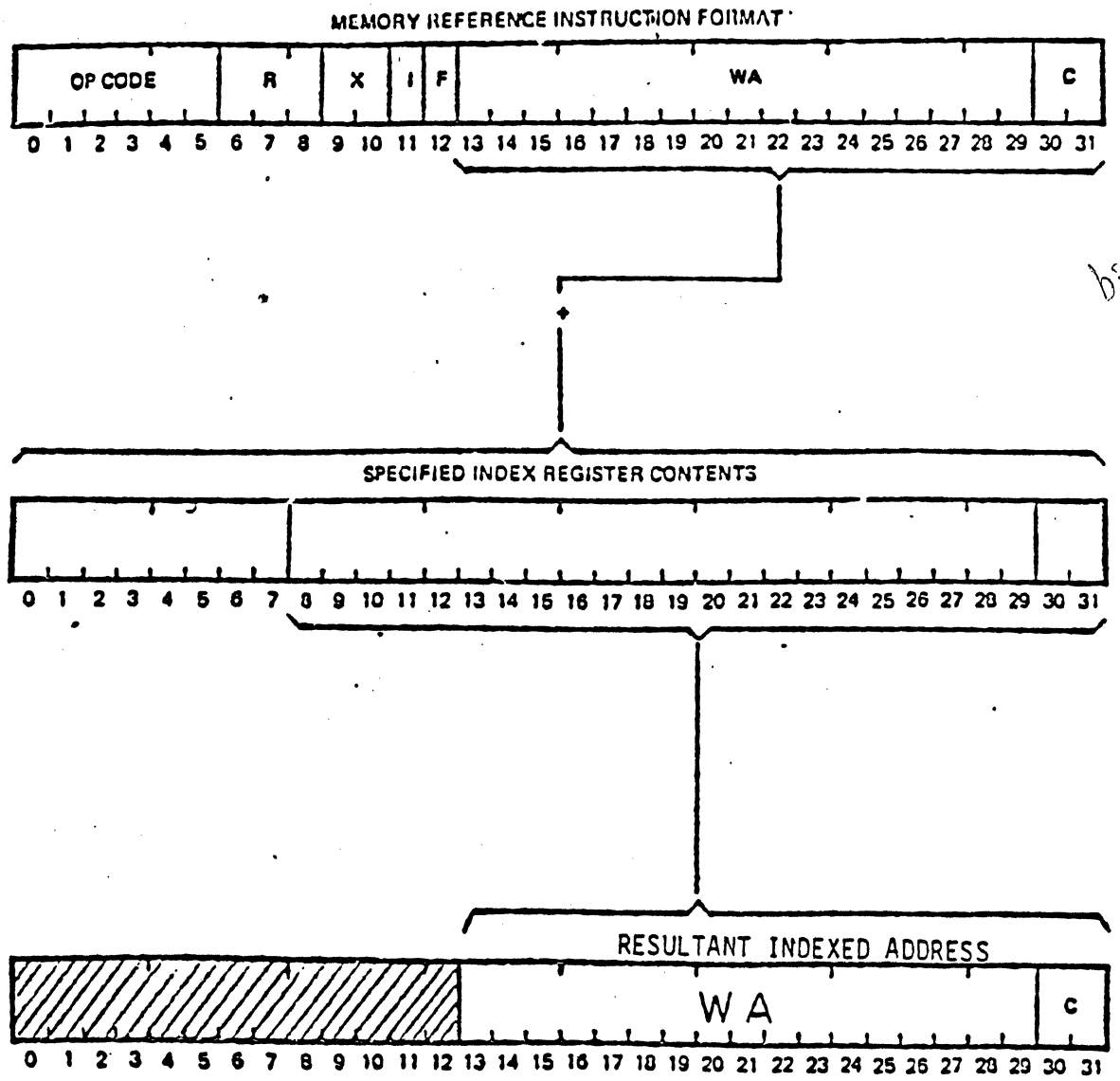



* BIT 29 MUST BE EQUAL TO ZERO

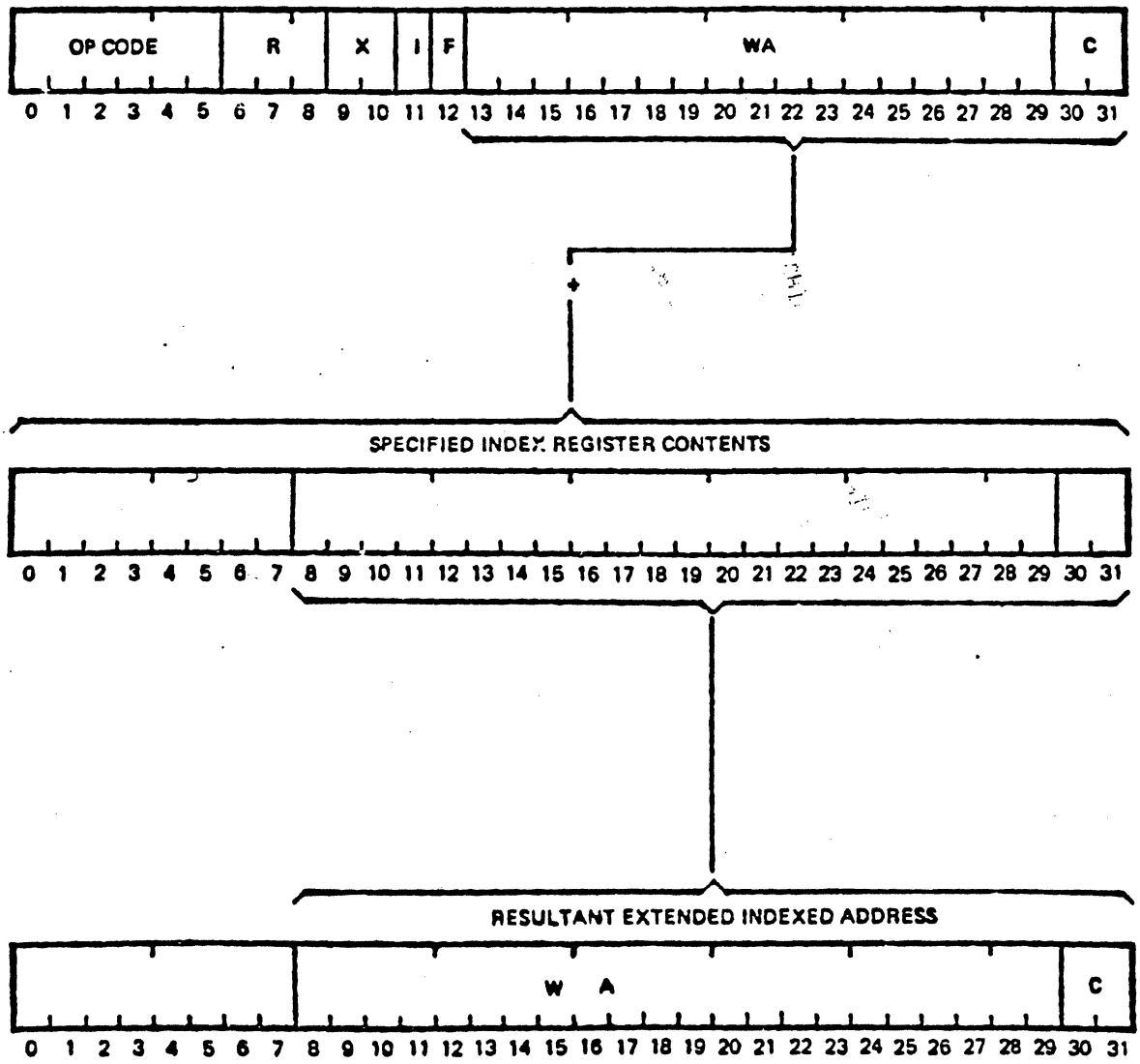
# INDEXED ADDRESSING



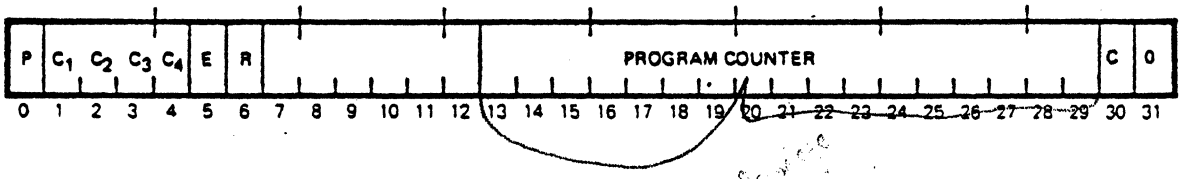
[redacted] INDEXED ADDRESSING

Granularity Bits

MEMORY REFERENCE INSTRUCTION FORMAT

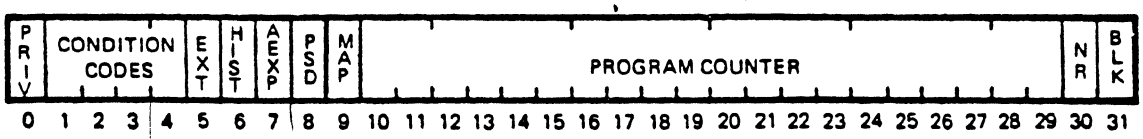


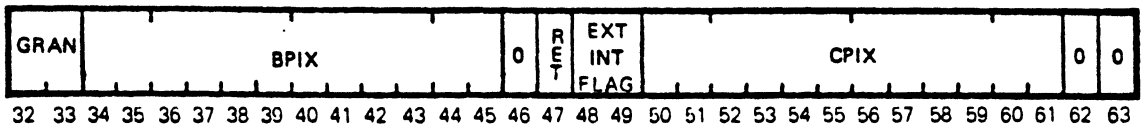
# PROGRAM STATUS WORD (PSW) FORMAT




| | |
|---|---|
| BIT 0 | DESIGNATES THE PRIVILEGED STATE BIT. |
| BITS 1-4 | DESIGNATE THE CURRENT CONDITION CODE. |
| BIT 5 | DEFINES THE CURRENT OPERATING MODE BIT 5=0 NORMAL ADDRESSING MODE BIT 5=1 EXTENDED ADDRESSING MODE |
| BIT 6 | DEFINES THE POSITION OF THE LAST INSTRUCTION EXECUTED.<br>BIT 6=0 LEFT HALFWORD OR FULLWORD<br>BIT 6=1 RIGHT HALFWORD |
| BITS 7,8 | DESIGNATE THE INFORMATION NECESSARY TO ENABLE THE EXECUTION OF LONG INSTRUCTIONS TO BE CONTINUED AFTER AN INTERRUPT, AND MUST NOT BE DISTURBED BY SOFTWARE. |
| BIT 7-12 | UNASSIGNED |
| BITS 13-29 | CONTAIN THE WORD ADDRESS (PC) OF THE INSTRUCTION CURRENTLY BEING EXECUTED. |
| BIT 30 | DEFINES THE POSITION OF THE ~~CURRENT~~ NEXT INSTRUCTION (LEFT OR RIGHT INSTRUCTION).<br>BIT 30=0 LEFT HALFWORD<br>BIT 30=1 RIGHT HALFWORD |

# PSD FORMAT




current process index

BIT 0 = 0 UNPRIVILEGED MODE
= 1 PRIVILEGED MODE

BITS 1-4 ARE CONDITION CODES
BIT 1 = CC1
2 = CC2
3 = CC3
4 = CC4

BIT 5 = 0 EXTENDED MODE (OFF)
= 1 EXTENDED MODE (ON)

BIT 6 = 0 LAST INSTRUCTION EXECUTED WAS NOT A RIGHT HALFWORD
= 1 LAST INSTRUCTION EXECUTED WAS A RIGHT HALFWORD

BIT 7 = 0 ARITHMETIC EXCEPTION TRAP MASK (OFF)
= 1 ARITHMETIC EXCEPTION TRAP MASK (ON)

BIT 8 = 0 COMPUTER IS IN PSW MODE (DISPLAYED PSD ONLY) *
= 1 COMPUTER IS IN PSD MODE (DISPLAYED PSD ONLY) *

BIT 9 = 0 UNMAPPED (DISPLAYED PSD ONLY) * (only shows if mapped)
= 1 MAPPED (DISPLAYED PSD ONLY) *

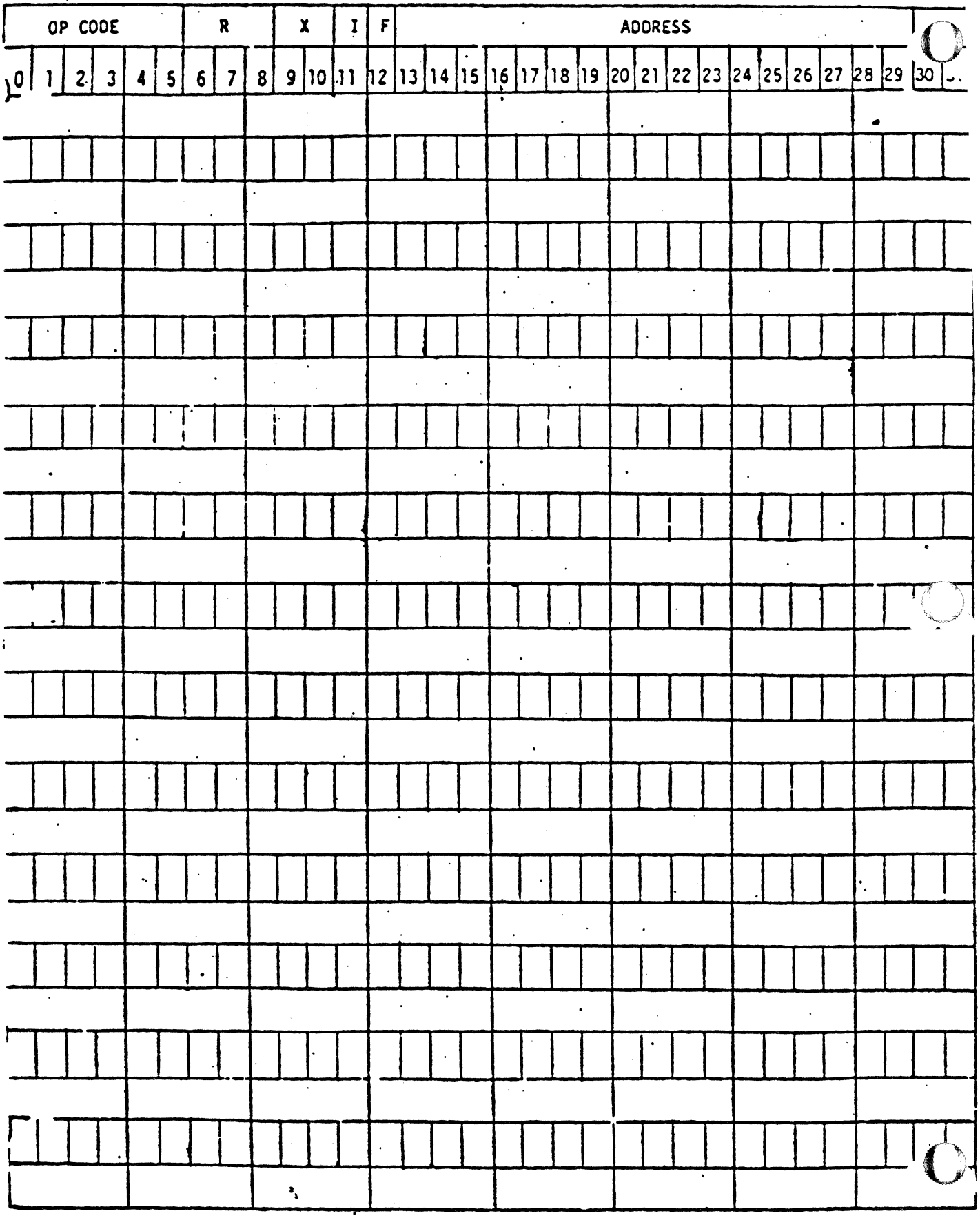
BITS 10-12 ARE NOT USED
BITS 13-29 ARE LOGICAL WORD ADDRESS
BIT 30 NEXT INSTRUCTION IS A RIGHT HALFWORD
BIT 31 BLOCKED (DISPLAYED PSD ONLY) * Interrupts Are Blocked
BITS 32-33 INDICATE MAP GRANULARITY, 00=UNMAPPED AND ALL OTHERS=8K MAP GRANULARITY
BITS 34-45 PROVIDE A WORD INDEX INTO THE MASTER PROCESS LIST (MPL) FOR THE BASE PROCESS INDEX TABLE FOR mpx
BIT 46 NOT USED
BIT 47 RETAIN CURRENT MAP CONTENTS
BITS 48-49 INTERRUPT CONTROL FLAGS

| BITS | | |
|---|---|---|
| 48 | 49 | |
| 0 | 0 | OPERATE WITH UNBLOCKED INTERRUPTS |
| 0 | 1 | OPERATE WITH BLOCKED INTERRUPTS |
| 1 | 0 | RETAIN CURRENT BLOCKING MODE } Same |
| 1 | 1 | RETAIN CURRENT BLOCKING MODE } Same |

BITS 50-61 PROVIDE WORD INDEX INTO MASTER PROCESS LIST (MPL) FOR CURRENT PROCESS
BITS 62-63 NOT USED

* THESE BITS ARE USED FOR DISPLAY ONLY AND ARE NOT PRESENT IN THE PSD STORED IN MEMORY.

## MEMORY REFERENCE INSTRUCTIONS



# INSTRUCTION SET

The tables on the following pages are provided to aid you in coding up machine language instructions.

With the use of these tables, coding machine language instructions should take less time and possibly reduce errors.

R OR RD FIELD

| | |
|---|---|
| R0 | 00000000 |
| R1 | 00800000 |
| R2 | 01000000 |
| R3 | 01800000 |
| R4 | 02000000 |
| R5 | 02800000 |
| R6 | 03000000 |
| R7 | 03800000 |

RS FIELD

| |
|---|
| 0000 |
| 0010 |
| 0020 |
| 0030 |
| 0040 |
| 0050 |
| 0060 |
| 0070 |

INDEX REG.

| | |
|---|---|
| X0 | 00000000 |
| X1 | 00200000 |
| X2 | 00400000 |
| X3 | 00600000 |

INDIRECT

00100000

PRIORITY LEVEL

| | |
|---|---|
| 1 = 0008 | 10 = 0080 |
| 2 = 0010 | 11 = 0088 |
| 3 = 0018 | 12 = 0090 |
| 4 = 0020 | 13 = 0098 |
| 5 = 0028 | 14 = 00A0 |
| 6 = 0030 | 15 = 00A8 |
| 7 = 0038 | 16 = 00B0 |
| 8 = 0040 | 17 = 00B8 |
| 9 = 0048 | 18 = 00C0 |
| A = 0050 | 19 = 00C8 |
| B = 0058 | 1A = 00D0 |
| C = 0060 | 1B = 00D8 |
| D = 0068 | 1C = 00E0 |
| E = 0070 | 1D = 00E8 |
| F = 0078 | 1E = 00F0 |
| | 1F = 00F8 |

EXAMPLE:

To derive the MACHINE CODE for LB 2,X'1000',3 (LOAD BYTE)

The OP CODE is AC08; then add

| | |
|---|---|
| AC081000 | Op Code and Address |
| 01000000 | Register 2 |
| AD081000 | |
| 00600000 | Index By R3 |
| AD681000 | Complete Instruction |

EXAMPLE:

To derive the MACHINE CODE for TRR 2,4 (TRANSFER REG TO REG)

The OP CODE is 2C00; then add

| | |
|---|---|
| 2C00 | Op Code |
| 0200 | $R_D$ 4 |
| 2E00 | |
| 0020 | $R_S$ 2 |
| 2E20 | Complete Instruction |

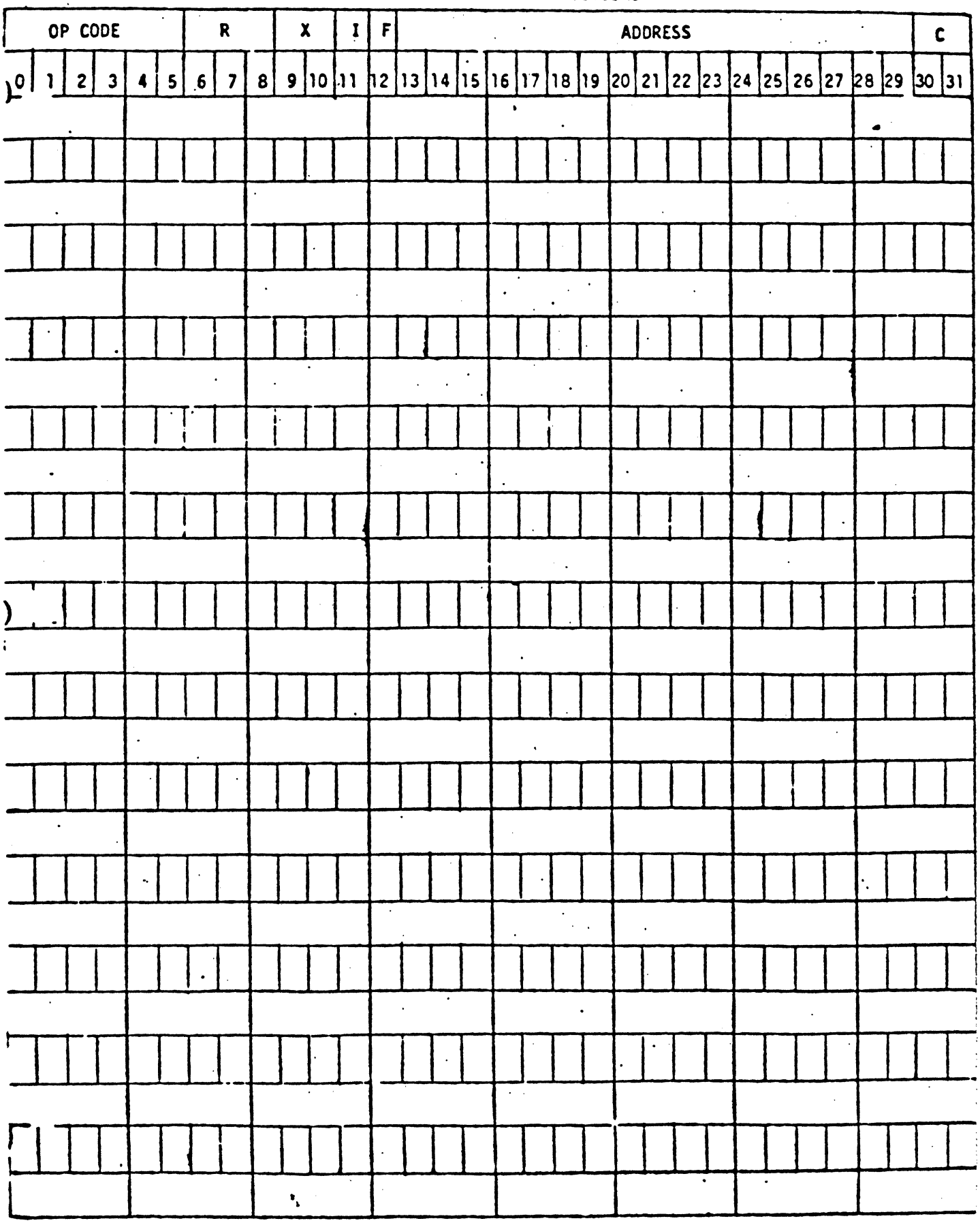
EXAMPLE:

To derive the MACHINE CODE for AI 7 (ACTIVATE INTERRUPT LVL'7')

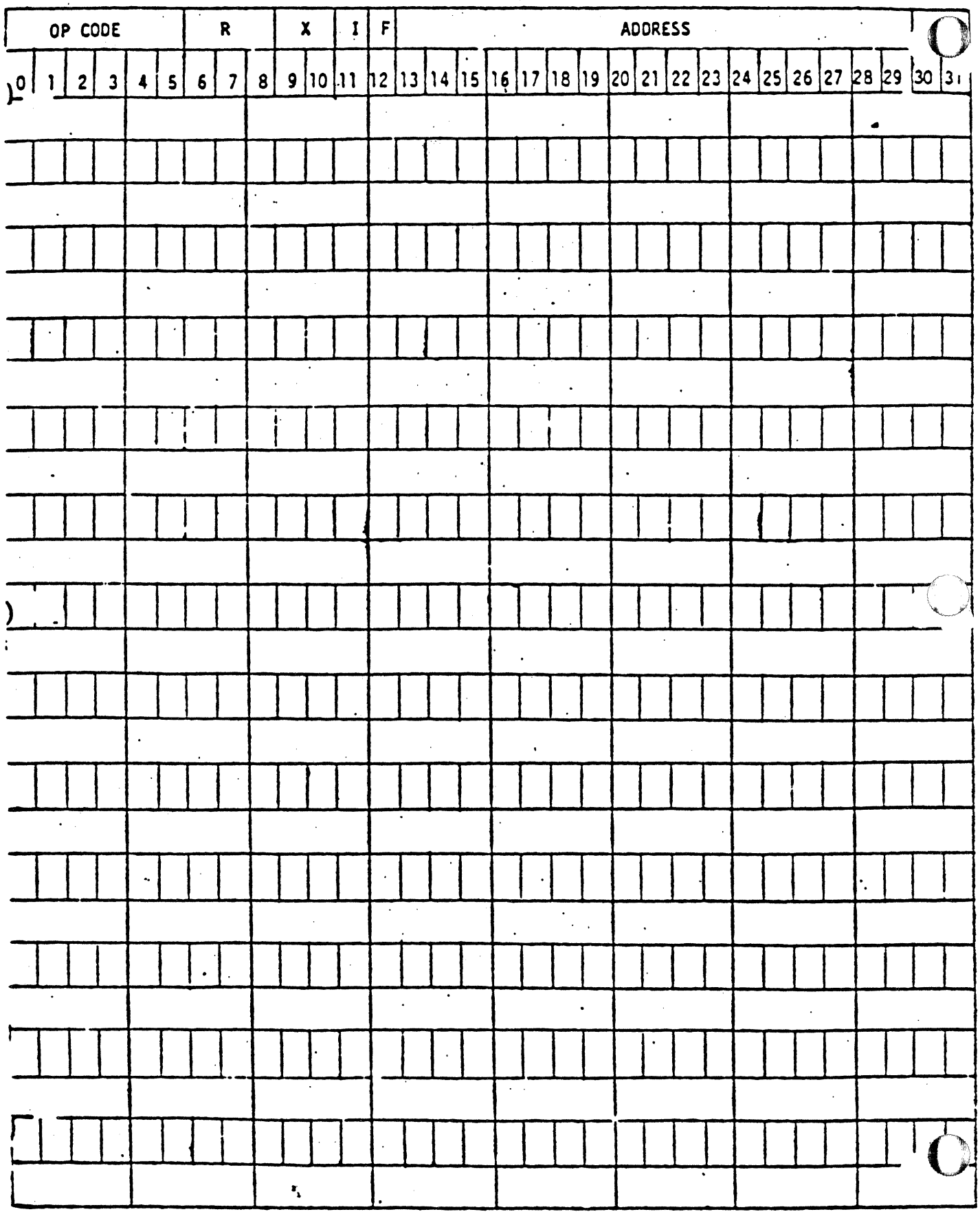
The Op Code is FC03; then add

| | |
|---|---|
| FC030000 | |
| 00380000 | Pri Level |
| FC3B0000 | Complete Instruction |

## MEMORY REFERENCE INSTRUCTIONS



## MEMORY REFERENCE INSTRUCTIONS

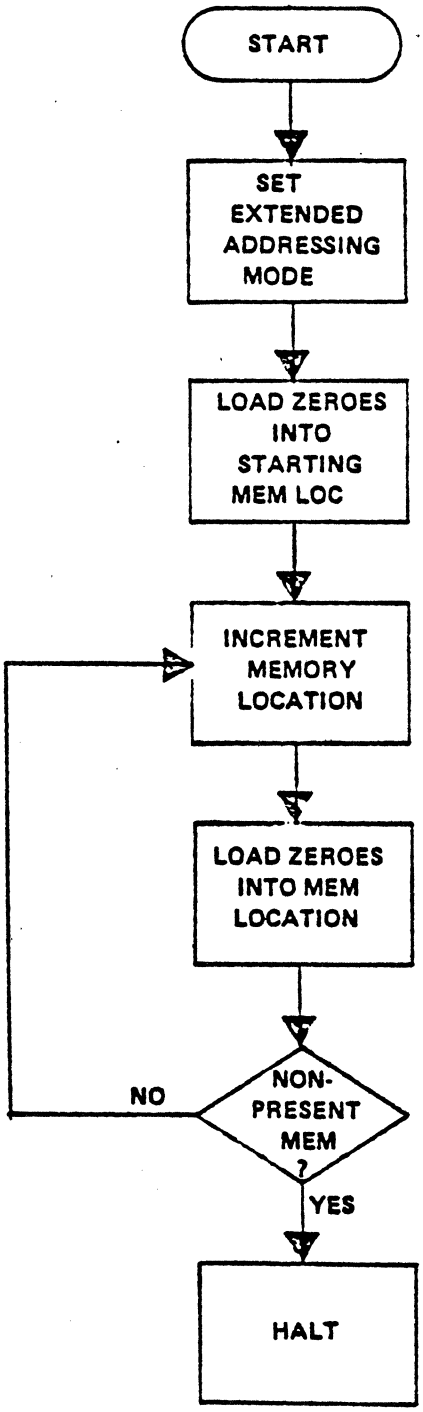


## 32/75 PRIVILEGED INSTRUCTIONS

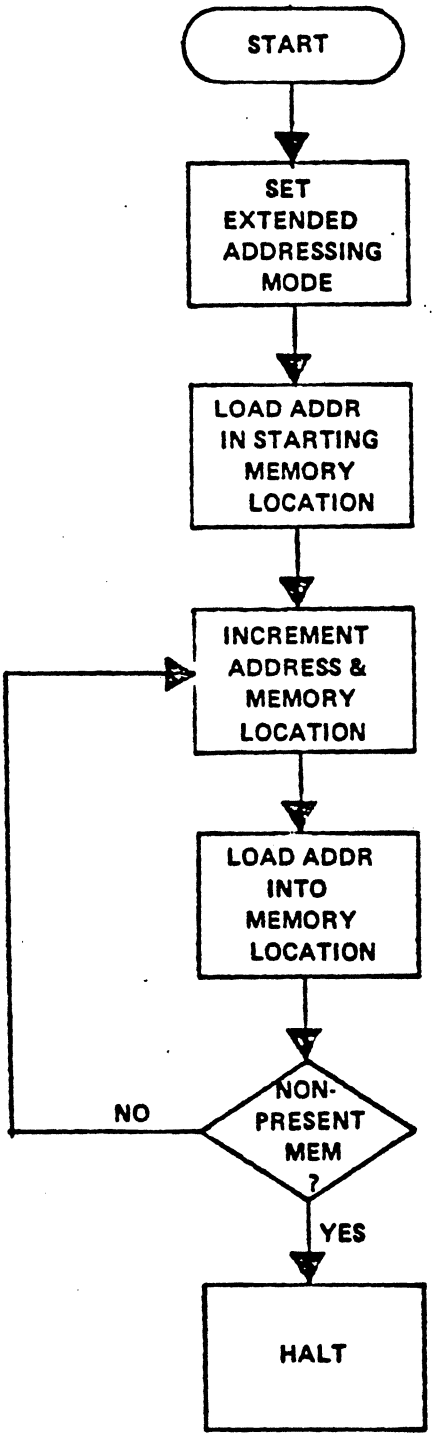
| | | |
|---|---|---|
| 1. | RDSTS | READ CPU STATUS |
| 2. | TSCR | TRANSFER S.P. TO REG. |
| 3. | TRSC | TRANSFER REG. TO S.P. |
| 4. | TRP | TRANSFER REG. TO PROTECT |
| 5. | LMAP | LOAD MAP REGS. |
| 6. | TMAPR | TRANSFER MAP TO REG. |
| 7. | WWCS | WRITE TO WRITABLE CONTROL STO |
| 8. | RWCS | READ WCS |
| 9. | BRI | BRANCH AND RESET INTERRUPT |
| 10. | LPSD | LOAD PSD |
| 11. | LPSDCM | LOAD PSD AND CHANGE MAP |
| 12. | HALT | |
| 13. | EI | ENABLE INTERRUPT |
| 14. | DI | DISABLE INTERRUPT |
| 15. | RI | REQUEST INTERRUPT |
| 16. | AI | ACTIVATE INTERRUPT |
| 17. | DAI | DEACTIVATE INTERRUPT |
| 18. | SETCPU | SET CPU MODE |

Note: ALL I/O INSTRUCTIONS ARE PRIVILEGED.

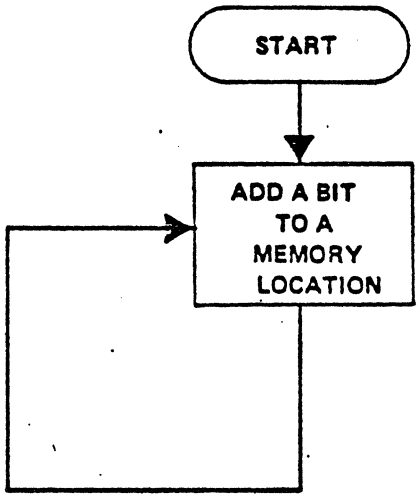
# CLEAR MEMORY PROGRAM



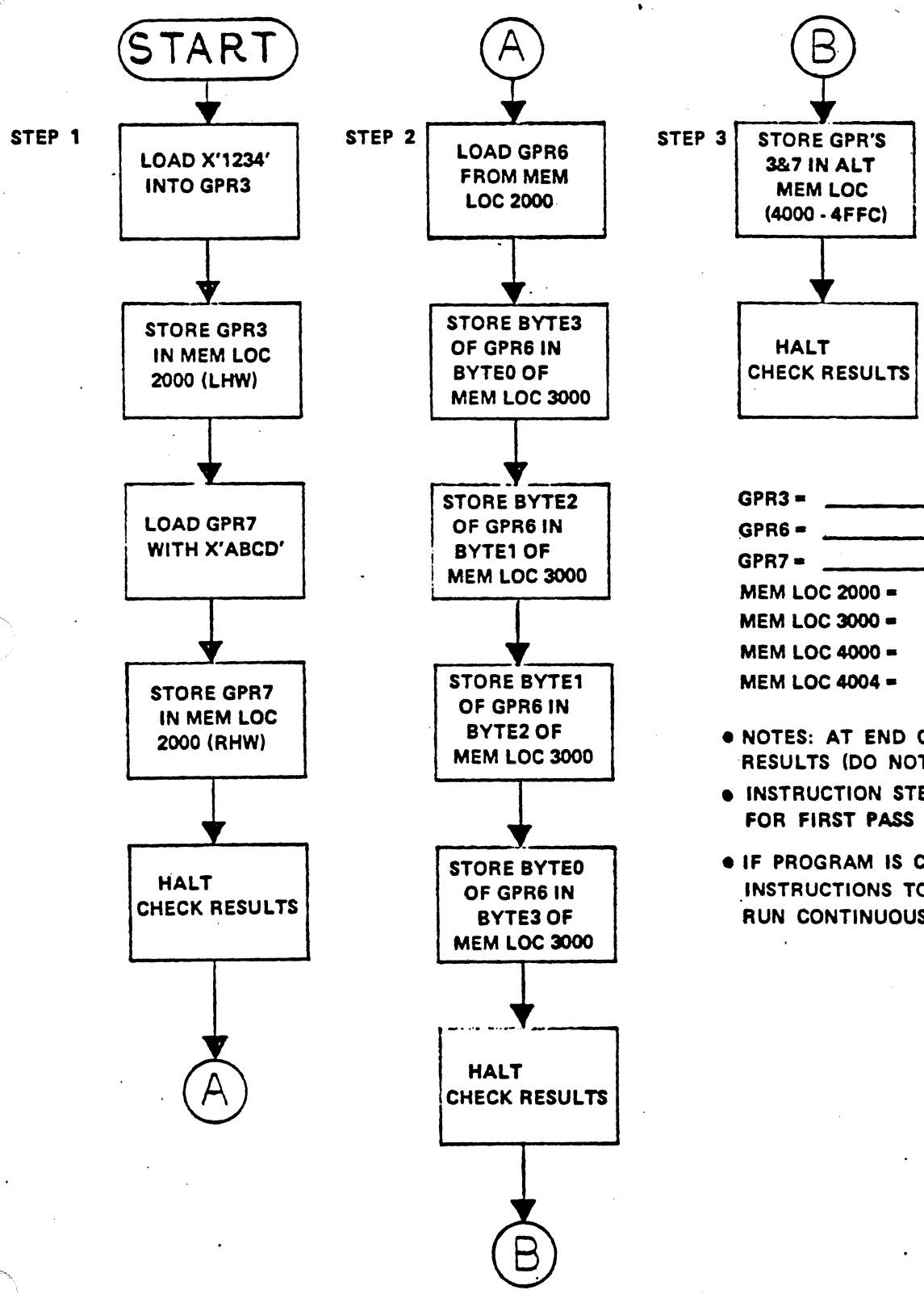
# ADDRESS OF ADDRESS PROGRAM



# ADD A BIT IN MEMORY PROGRAM



# INSTRUCTION SET THUMB IN WORKSESSION




GPR3 = _____

GPR6 = _____

GPR7 = _____

MEM LOC 2000 = _____

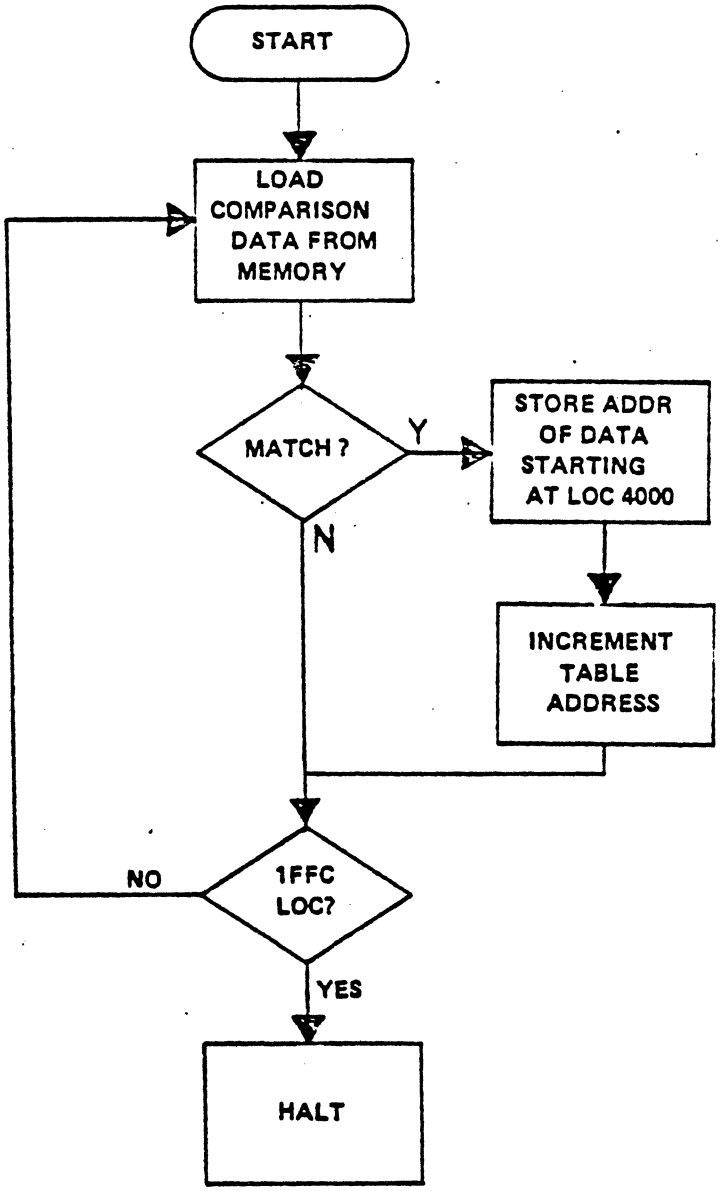
MEM LOC 3000 = _____

MEM LOC 4000 = _____

MEM LOC 4004 = _____

- NOTES: AT END OF EACH STEP CHECK RESULTS (DO NOT HIT SYSTEM RESET)
- INSTRUCTION STEP PROGRAM THRU FOR FIRST PASS
- IF PROGRAM IS CORRECT', CHANGE HALT INSTRUCTIONS TO ALLOW PROGRAM TO RUN CONTINUOUSLY.

# INSTRUCTION SET THUMB IN WORKSESSION




- NOTES: SEARCH THROUGH MEMORY STARTING AT MEMORY LOCATION 000 THRU 1FFC LOOKING FOR ALL LOCATIONS THAT CONTAIN THE HEX DATA 'FC060000'.
- FOR EVERY MATCH, STORE THE ADDRESS OF THE DATA IN A SEPARATE TABLE IN MEMORY STARTING AT HEX LOCATION 4000.

DAY 2

INSTRUCTION SET

WORKSESSION

REFERENCE:

301-320070-000

32/70 SERIES
REFERENCE MANUAL
SECTION SIX

Objective: After completing this worksession, you will be able to use the 32/70 Series reference manual as a tool in generating machine language programs as an aid in troubleshooting hardware problems.

Note: Use Appendix 'A' in the reference manual as an aid in finding instructions.

INSTRUCTION SET WORKSESSION #1

1. Bits 0 through 5 of the memory reference instruction represents its B.

   A. Format
   B. Op-Code
   C. Instruction Identifier
   D. Word Address

2. A two-to-six letter symbolic representation of the instruction name accepted by the assembler program is called a Mnemonic.

3. When a halfword instruction is used, what is contained in the other half of the thirty-two bits?

   No-OP or another Halfword Instruction

4. In a memory reference instruction, what is designated by bits 9 and 10?

   The index register 1, 2, 3

5. In a memory reference instruction, what does bit eleven represent?

   Indirect

6. Adding a register to the word address to get an effective memory address is called

   Indexing

7. Accessing a location in memory, reading that location and using the word read as another address to memory is called Indirect addressing.

8. What is the mnemonic for load byte? A

   A. LB
   B. LH
   C. LW
   D. LD

9. What is the mnemonic for a load word instruction?

   LW

10. What is the op-code of a load word instruction?

    AC00

11. What is the purpose of the load/store word instructions?

    Load Reg Transfer contents register to memory vice versa
    Store mem

12. What is the mnemonic for a store word instruction?

    STW

13. What is the op-code for a store word instruction?

    D400

14. Given: Contents of GPR0=00000000
    Instructions = AC001000
    Location 1000 = FFFFFFFF

    What instruction is this? Load Word

    What is the mnemonic? LW

    What is the effective address? 1000

    What will be contained in GPR0 after execution? FFFFFFFF

15. In a memory reference instruction format, what is bit 12 used for?

    F-bit
    Byte/Word

16. What are bits 30 and 31 used for in a memory reference instruction?

    C bits RA=1 LHW

17. Upon execution of a fullword instruction, the program counter increments normally by four. Why does this happen? Stays on word boundaries

18. What is the purpose of the fixed point arithmetic instructions? Do Arithmetic Functions

19. How are condition codes used in arithmetic operations? Equal, Greater, Less, Over flow (Branching)

20. Describe the uses of the Floating Point Arithmetic group of instructions? For extremely small and extremely large numbers

21. How is the logical instruction group used? For masking (AND, XOR, OR)

22. How are bit manipulation instructions used? Add Bit Memory, etc.

23. What is the purpose of the branch instructions? Looping, run subroutine

24. What is the purpose of the compare instructions? Test for a condition C.

25. What capabilities are provided by the transfer register instructions? Move register to register
make value negative
Xfer scratchpad

26. In the transfer register format, what is $R_D$? Dest. Reg.

27. In an instruction format there is an op-code. The augment code modified the op-code to allow more than one instruction for that op-code. Is this true? If not, why? Yes

28. What capabilities are provided by the shift instuctions?
Move data in CPU Registers

29. What is the purpose of the control instruction group?
No-Op, Halt, Wait Control CPU
Privileged

30. What is the purpose of the "HALT" instruction?
Stop Execution of program by CPU

31. What is the purpose of the "NO-OP" instruction? Complete a halfword instruction. Time Delays

32. What is the difference between a "Load Immediate" (LI) and a "Load Effective Address" (LEA) instruction?
Load Immediate - load register with "value"
Load Effective Address - take address index, increment then load address

89804003

1000 [illegible] 0003 0000 0011

Priv E 4 0 0 3 Int Blocked

4 RH_W

ACRP

DSD PC-9000

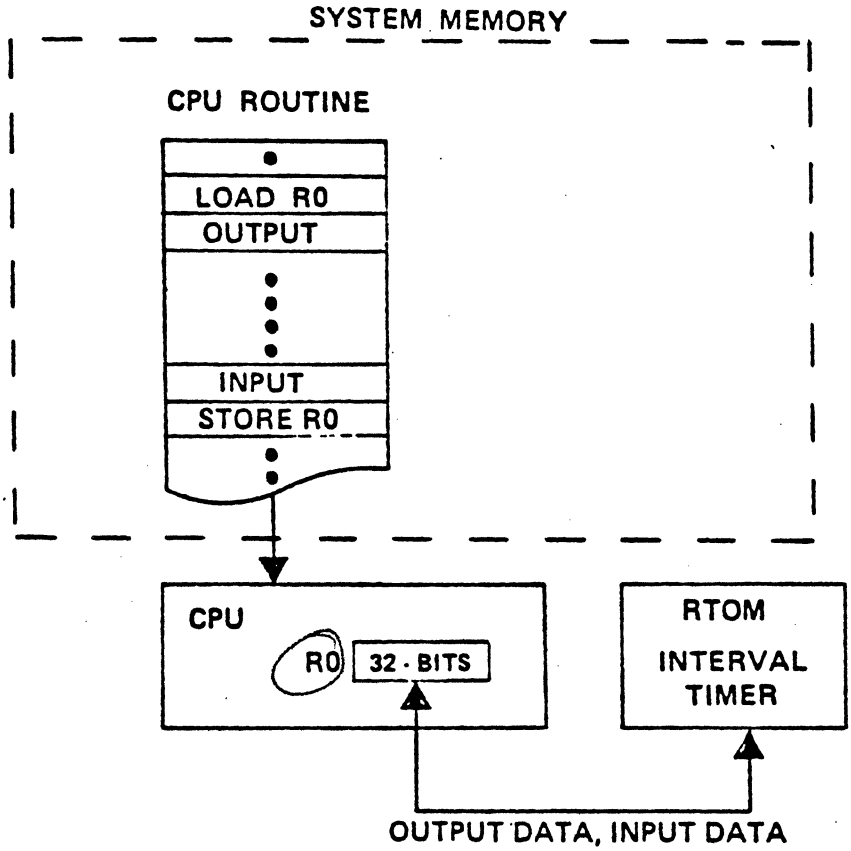
NR Blocked

# DAY 3

# SECTION 3

# I/O WITHOUT INTERRUPTS

# CLASS '3' I/O




CLASS '3' I/O

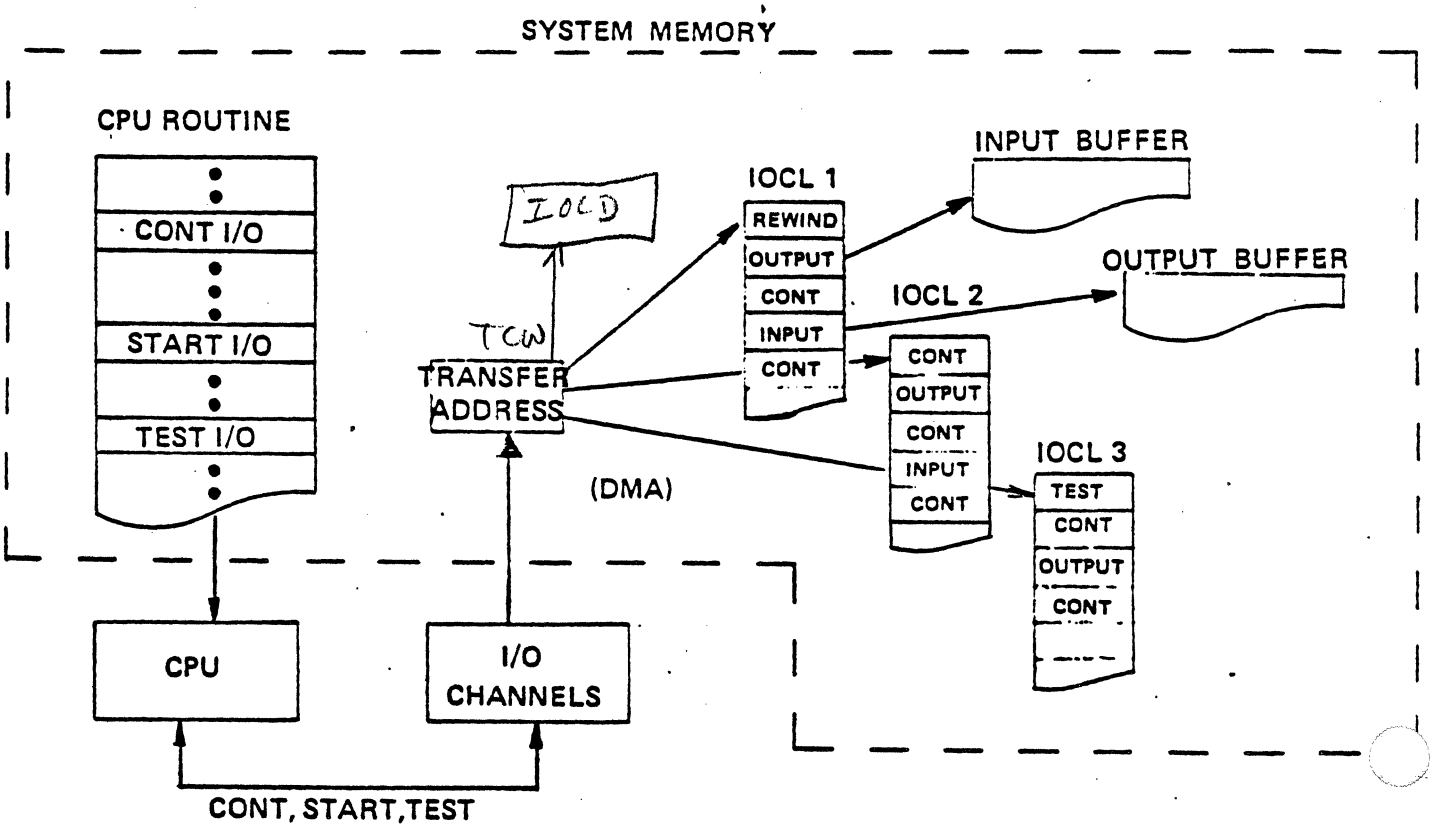
CPU R0 I/O (32 BITS)

32 bit Down Counter

Used by software (R0)

Register I/O

# CLASS 'D' I/O



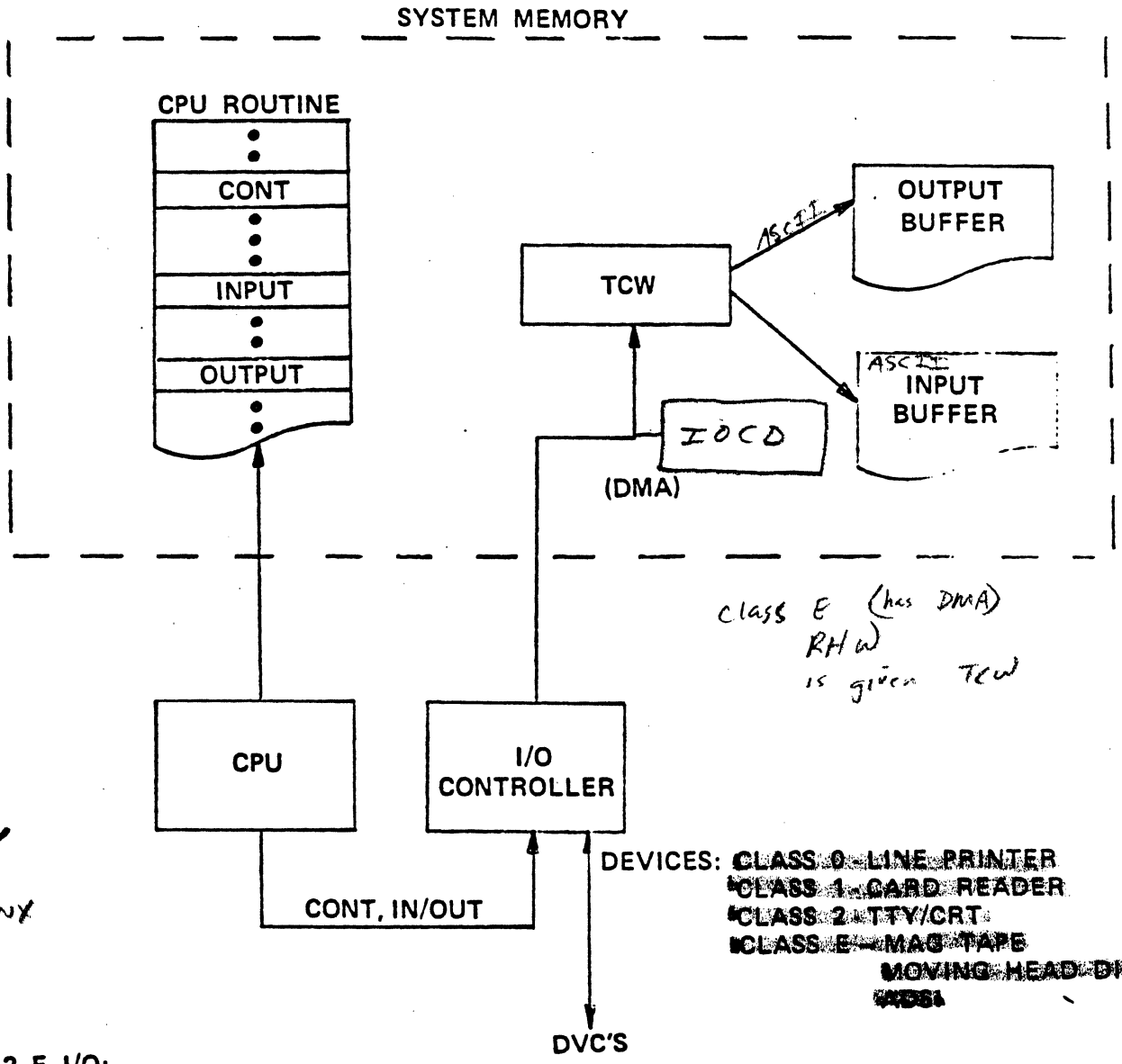

CLASS 'D' I/O:

buffer F 64 12 becomes 8

- DMA, 16 MB ADDRESSING, 65 KB TRANSFERS PER IOCL CMD
- INFINITE RECORD LENGTH (DATA CHAINED IOCL CMDS).
- CLASS 'E' MACRO I/O INSTRUCTION SET USED (CD/TD).
- CONTROL FUNCTIONS ARE PERFORMED IN ONE ROUTINE.

DEVICES: 9103 GPMC, 9131 HSD.

# CLASS 0, 1, 2, E I/O




class E (has DMA) RHW is given TCW

TLC
console FNX

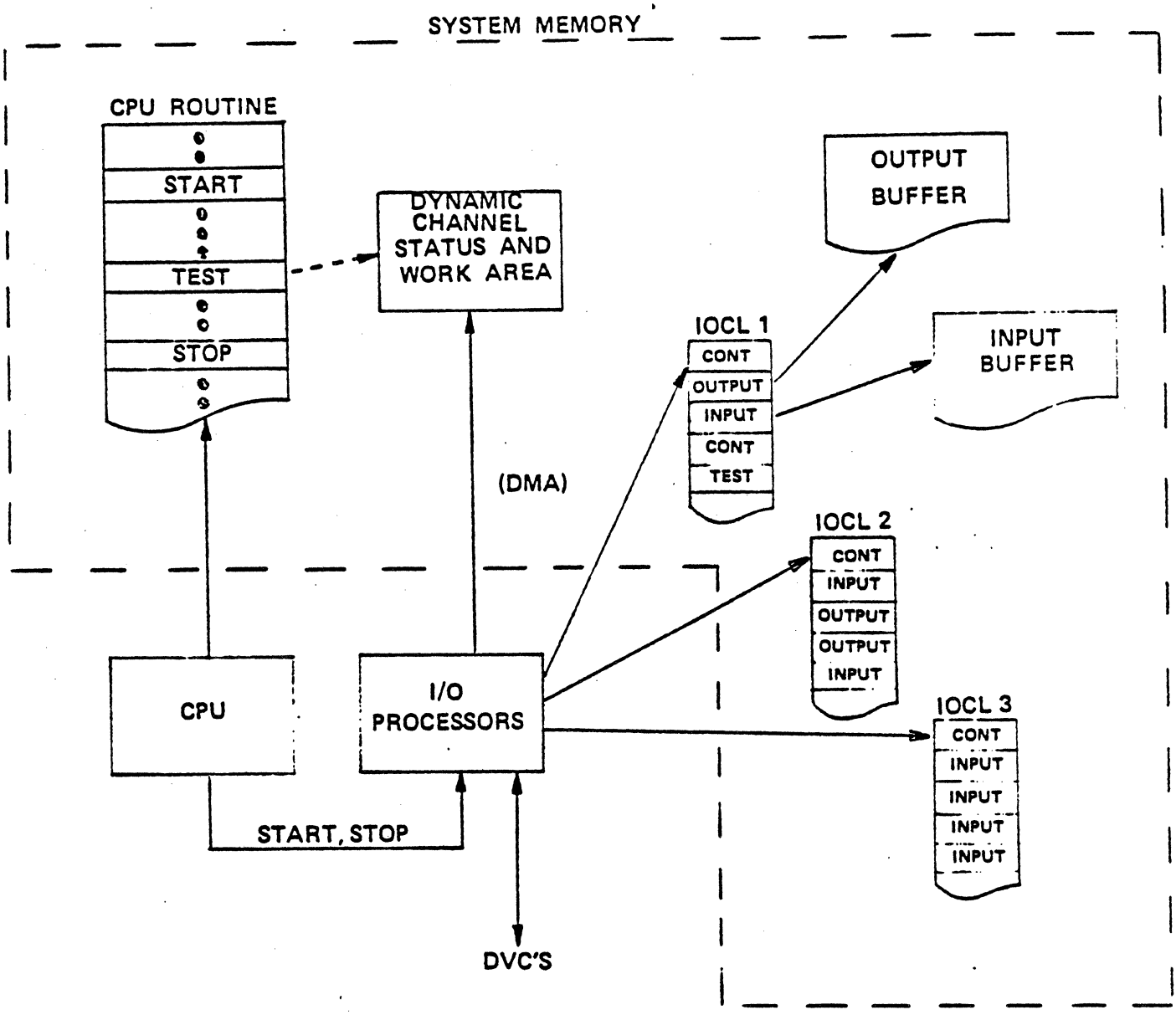
**CLASS 0, 1, 2, E I/O:**

DMA, 512 KB ADDRESSING
16 KB TRANSFERS MAX.

CLASS 0, 1, 2 ARE BYTE TRANSFER CONTROLLERS (4 KB TRANSFERS).
CLASS E ARE HALF WORD TRANSFER CONTROLLERS (8 KB TRANSFERS)
AND FULL WORD TRANSFER CONTROLLERS (16 KB TRANSFERS).
CPU ROUTINE PERFORMS CONTROL OF COMPLETE I/O ACTION.

TLC - Bytes  mag tape - HW  4096 Xfers
disc - W

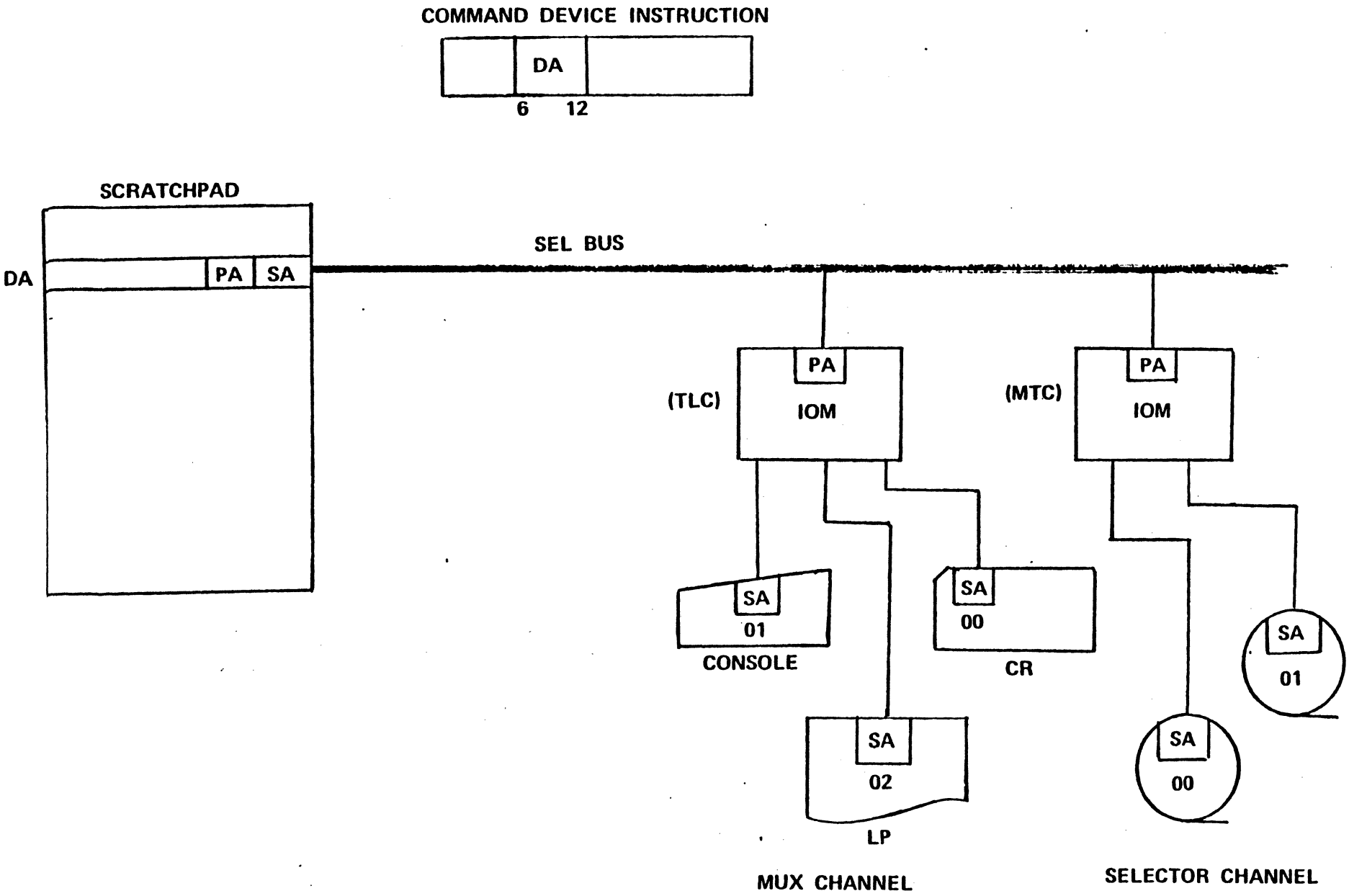
# CLASS 'F' I/O



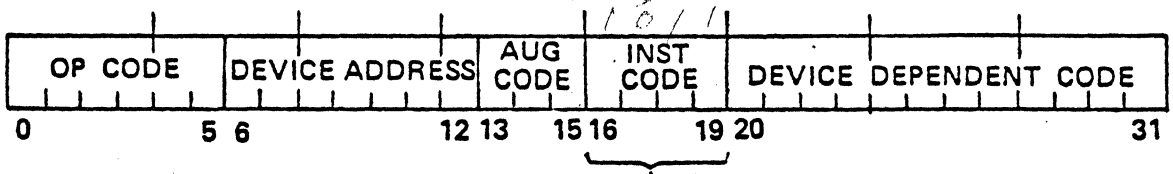

CLASS 'F' I/O:

- DMA, 16 MB ADDRESSING', 65 KB TRANSFERS, INFINITE RECORD LENGTH CAPABILITY (DATA CHAINED IOCL CMDS).
- INTELLIGENCE TO PERFORM CONTROL OF MULTIPLE I/O ACTIONS FOR COMPLEX I/O PROCESS
- CLASS F MACRO INSTRUCTION SET PROVIDES FOR ADDRESSING 128 CHANNELS WITH EACH HAVING 255 SUBADDRESSES (DVC'S).
- DYNAMIC STATUS POSTING IN ASSIGNED AREAS OF MEMORY FOR DIRECT VIEWING.
- CLASS F MACRO INSTRUCTION SET HAS A STANDARD I/O PROTOCOL FOR ALL CLASS F PROCESSORS.
- CLASS F: TAPE PROCESSOR
  DISC PROCESSOR

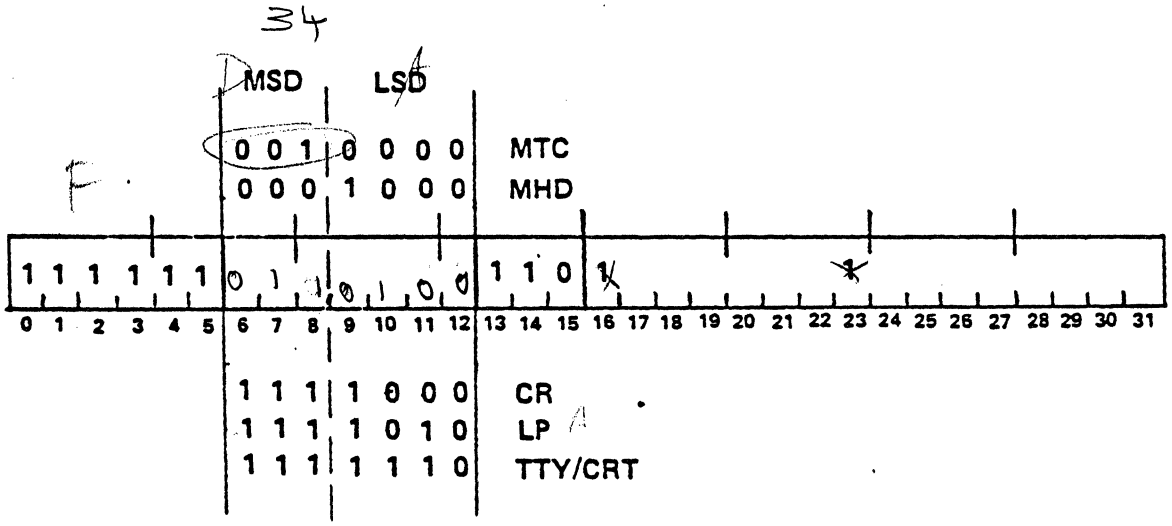
# I/O ADDRESSING SCHEME



# COMMAND DEVICE (CD) INSTRUCTION FORMAT



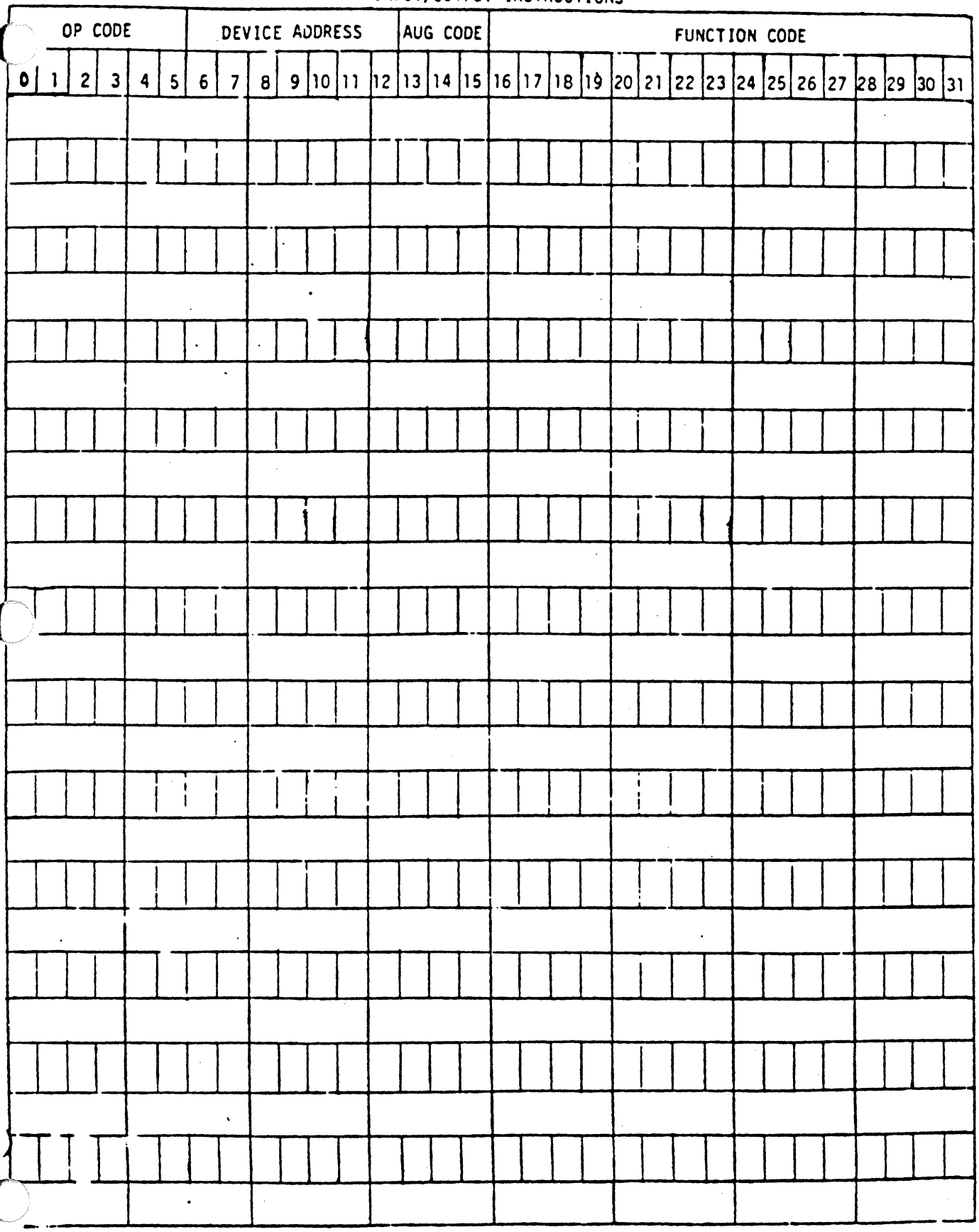

| 16 | 17 | 18 | 19 | INSTRUCTION FUNCTION |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | BITS 20-31 PROVIDE THE DEVICE DEPENDENT FUNCTION CODE' |
| 0 | 0 | 0 | 1 | TERMINATE TRANSFER (RESET I/O CONTROLLER) |
| 0 | 0 | 1 | 0 | TRANSFER CURRENT WORD ADDRESS |
| 1 | 0 | X | 0 | INITIATE OUTPUT TRANSFER (WRITE TO DEVICE) |
| 1 | 0 | X | 1 | INITIATE INPUT TRANSFER (READ FROM DEVICE) |



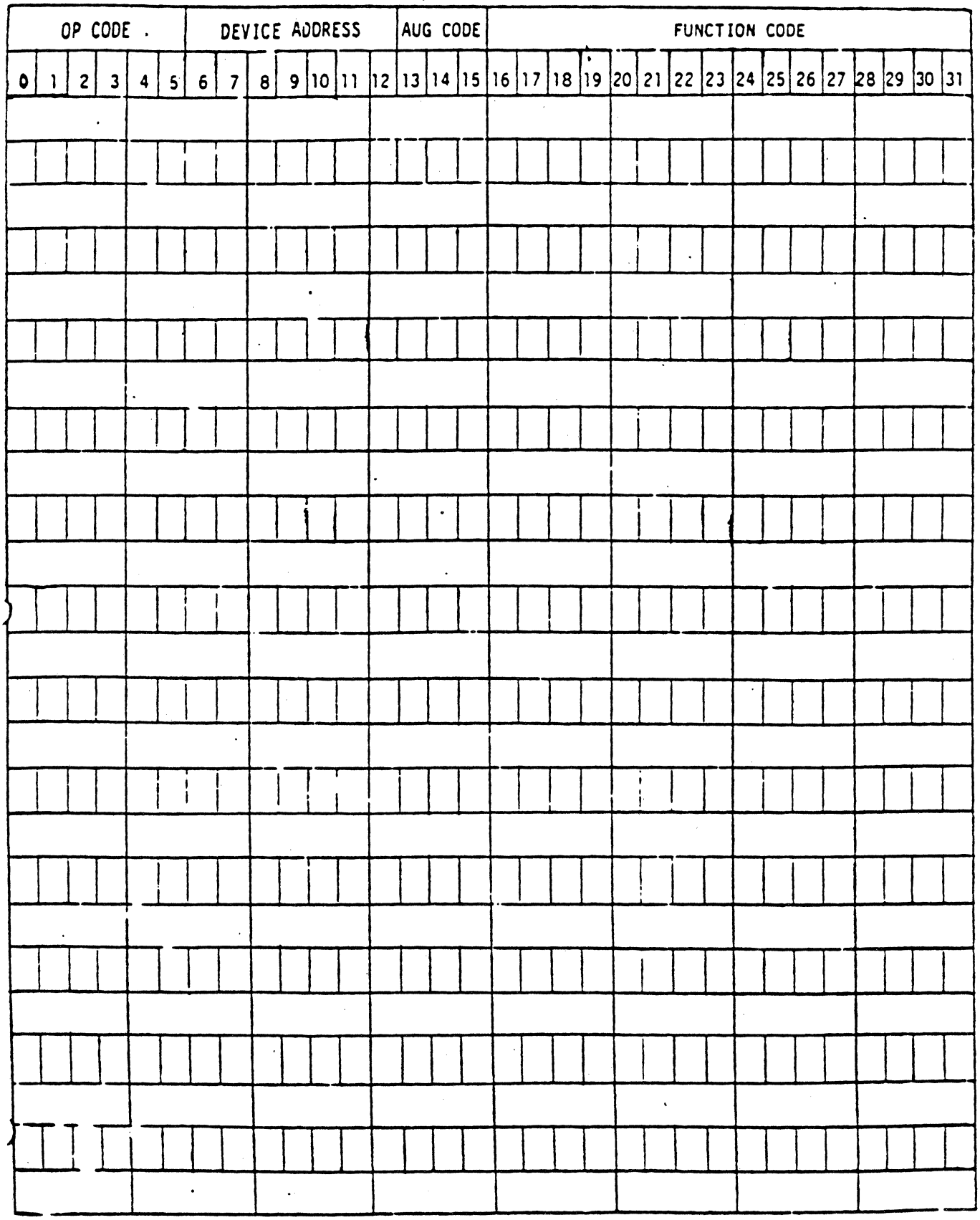

| F | F | D | 6 | 8 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|

EXAMPLE HEX CODE FOR LINE PRINTER CD TO ADVANCE ONE LINE AND PRINT.

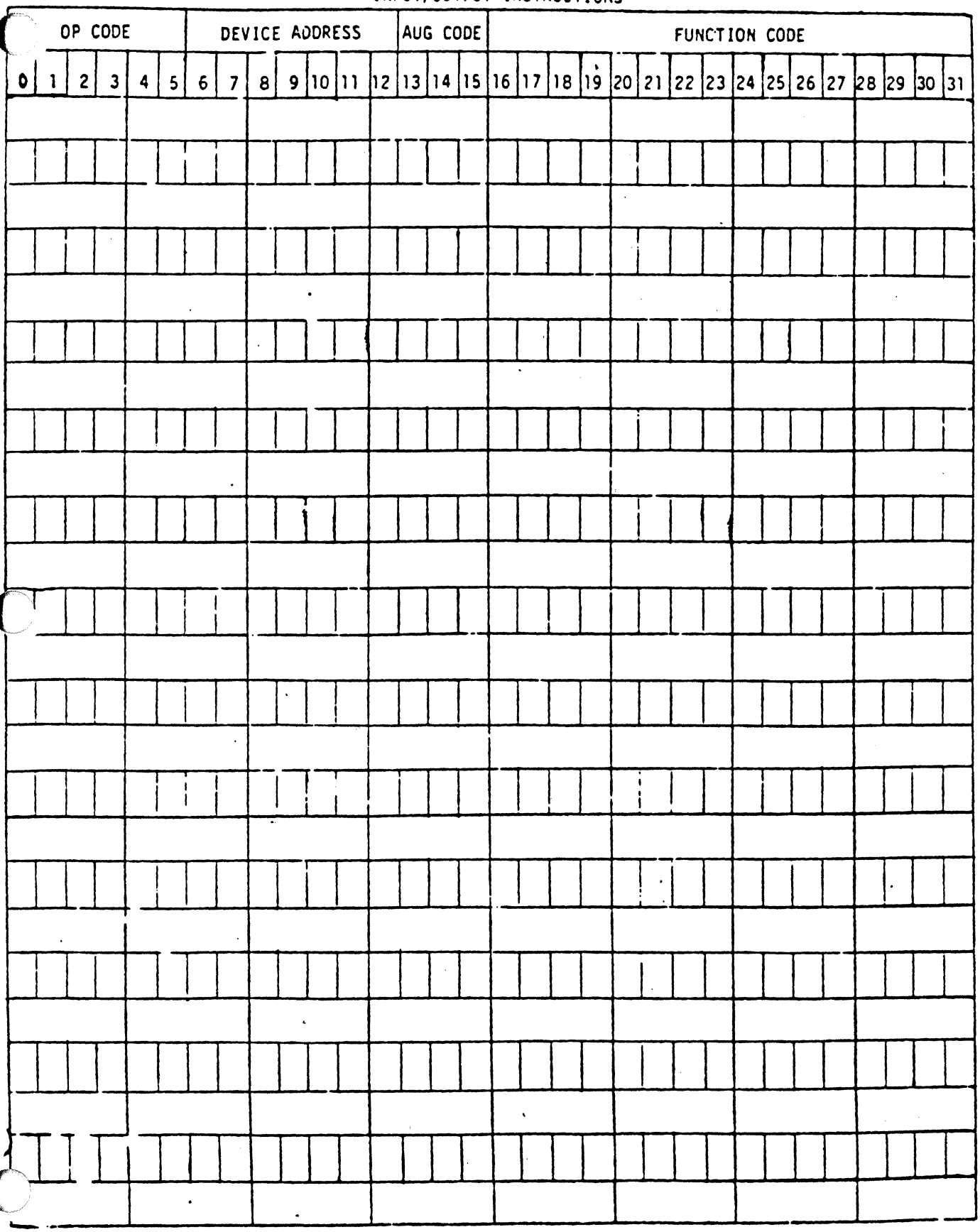
INPUT/OUTPUT INSTRUCTIONS



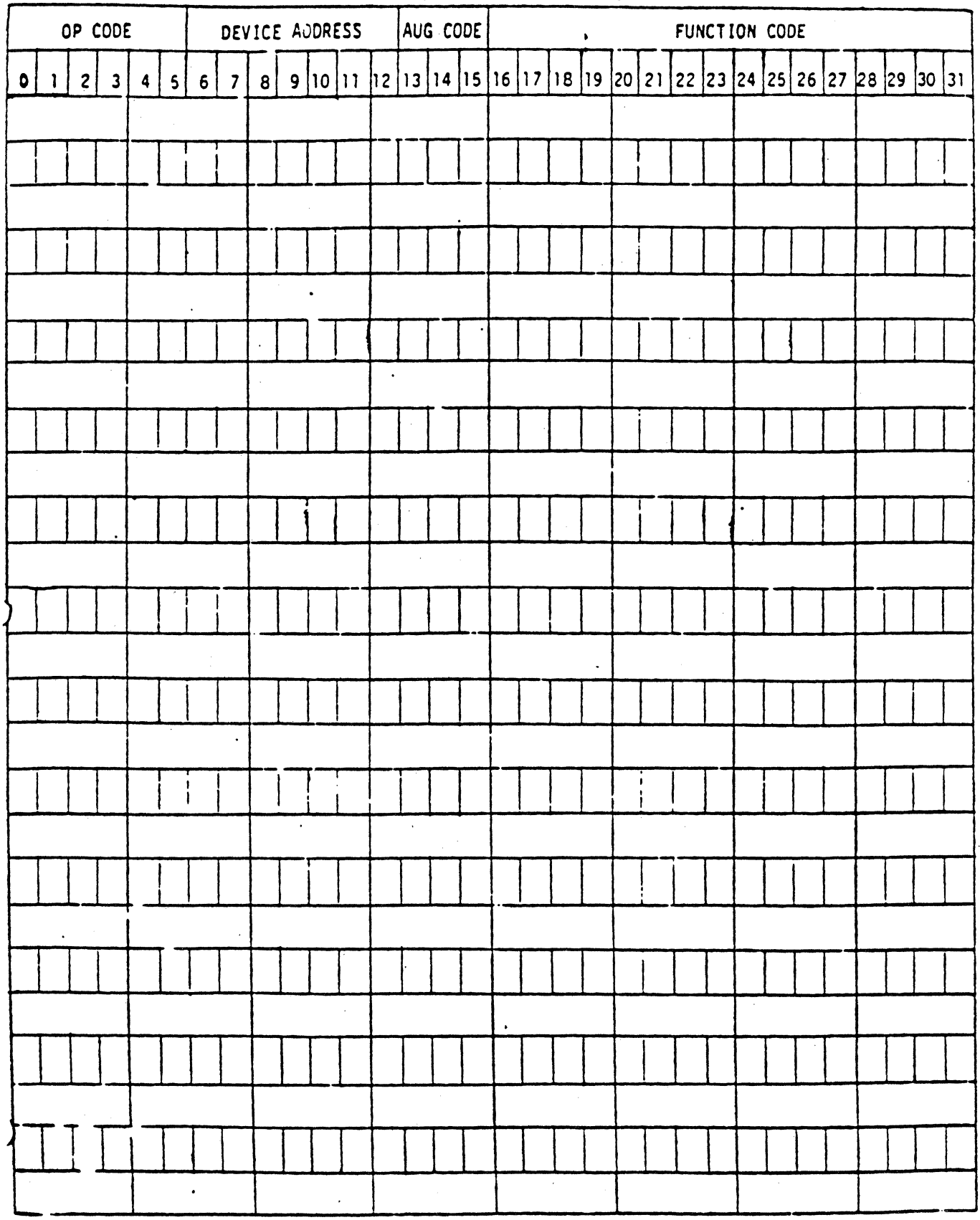
INPUT/OUTPUT INSTRUCTIONS



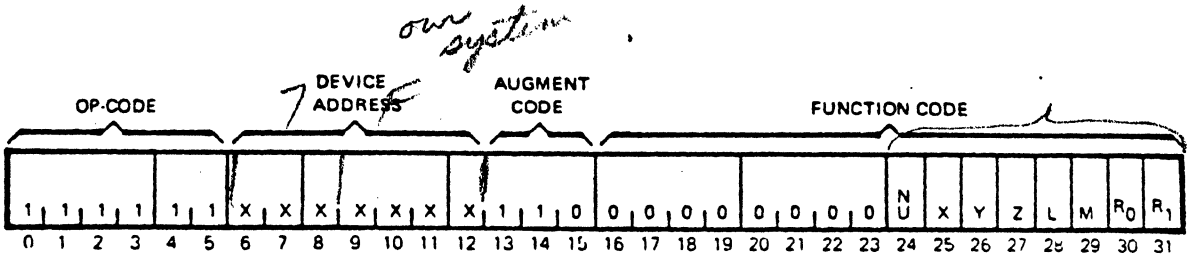
## INPUT/OUTPUT INSTRUCTIONS



INPUT/OUTPUT INSTRUCTIONS



# INTERVAL TIMER PROGRAMMING




X (BIT 25) = 1 — SPECIFIES READ TIMER, CAUSING THE 32-BIT CONTENTS OF THE TIMER TO BE LOADED INTO GPR 0.

Y (BIT 26) = 1 — SPECIFIES PROGRAM INTERVAL TIMER AND THAT BITS 27 THROUGH 31 ARE VALID.

Z (BIT 27) = 1 — ENABLE (START) INTERVAL TIMER.

Z (BIT 27) = 0 — DISABLE (STOP) INTERVAL TIMER.

L (BIT 28) = 1 — LOAD BITS 00-31 FROM THE GPR 0 INTO THE INTERVAL TIMER BITS 00-31.

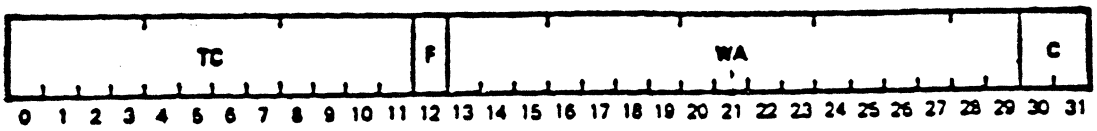
L (BIT 28) = 0 — DO NOT ALTER THE STORED COUNT.

M (BIT 29) = 1 — GENERATE MULTIPLE INTERRUPT. WHEN COUNT ZERO IS REACHED, GENERATE INTERRUPT, RELOAD INITIAL COUNT, AND CONTINUE COUNTING.

M (BIT 29) = 0 — GENERATE SINGLE INTERRUPT ON COUNT ZERO AND THEN COUNT NEGATIVE

| $R_0$ (BIT 30) | $R_1$ (BIT 31) | SELECT COUNT RATE |
|---|---|---|
| 0 | 0 | SELECT HIGH FREQUENCY |
| 0 | 1 | SELECT LOW FREQUENCY |
| 1 | 0 | SELECT 120 HZ |
| 1 | 1 | SELECT EXTERNAL CLOCK |

# TRANSFER CONTROL WORD (TCW) FORMAT




BITS 0-11 — DESIGNATE THE NUMBER OF TRANSFERS TO BE MADE BETWEEN MEMORY AND THE DEVICE CONTROLLER CHANNEL.

BITS 12,30,31 — SPECIFY THE FORMAT CODE FOR EACH TRANSFER (SEE TABLE 5-1).

BITS 13-29 — DESIGNATE THE MEMORY LOCATION FOR EACH TRANSFER.

NOTE

THE WA FIELD IS INTERPRETED AS A 24-BIT REAL ADDRESS BY THE I/O PROCESS. THEREFORE, THE ADDRESS RANGE IS LIMITED TO THE FIRST 512 KB OF MEMORY.

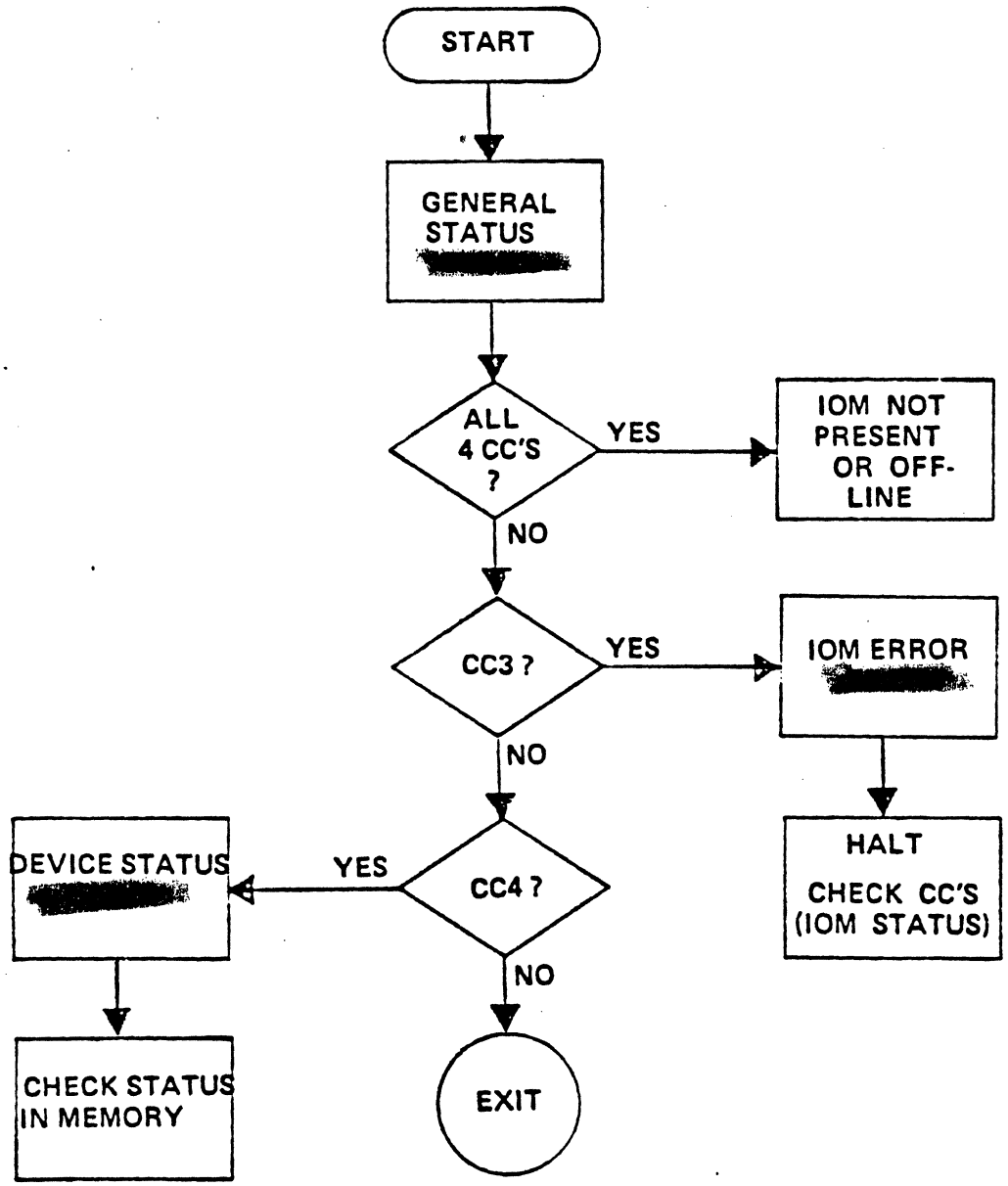
| Information Format | TC |
|---|---|
| Byte | 1XX |
| Halfword | 0Y1 |
| Word | 000 |

XX = Byte number
Y = 0 designates left halfword
Y = 1 designates right halfword

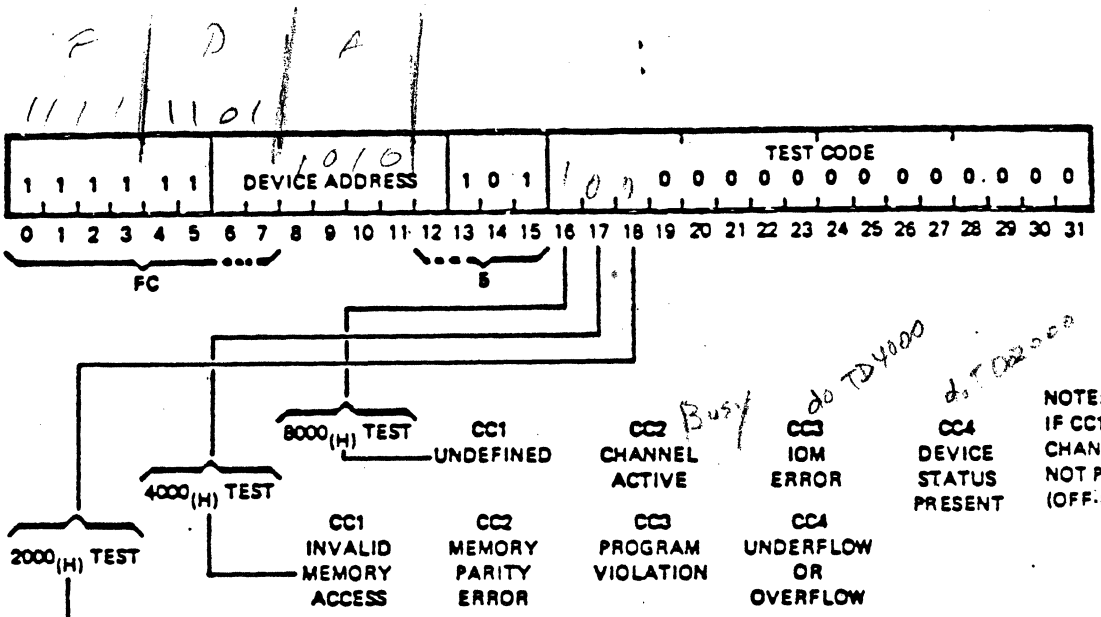
TCW DEDICATED LOCATIONS:

| | | | | | |
|---|---|---|---|---|---|
| 100 | | PRIORITY LEVEL 14 | 120 | | PRIORITY LEVEL 1C |
| 104 | | PRIORITY LEVEL 15 | 124 | | PRIORITY LEVEL 1D |
| 108 | MHD | PRIORITY LEVEL 16 | 128 | | PRIORITY LEVEL 1E |
| 10C | | PRIORITY LEVEL 17 | 12C | | PRIORITY LEVEL 1F |
| 110 | MTC | PRIORITY LEVEL 18 | 130 | CR | PRIORITY LEVEL 20 |
| 114 | | PRIORITY LEVEL 19 | 134 | LP | PRIORITY LEVEL 21 |
| 118 | | PRIORITY LEVEL 1A | 138 | | PRIORITY LEVEL 22 |
| 11C | | PRIORITY LEVEL 1B | 13C | TTY/CRT | PRIORITY LEVEL 23 |

# LEVELS OF TEST DEVICE (TD)



# TEST DEVICE INSTRUCTION FORMAT



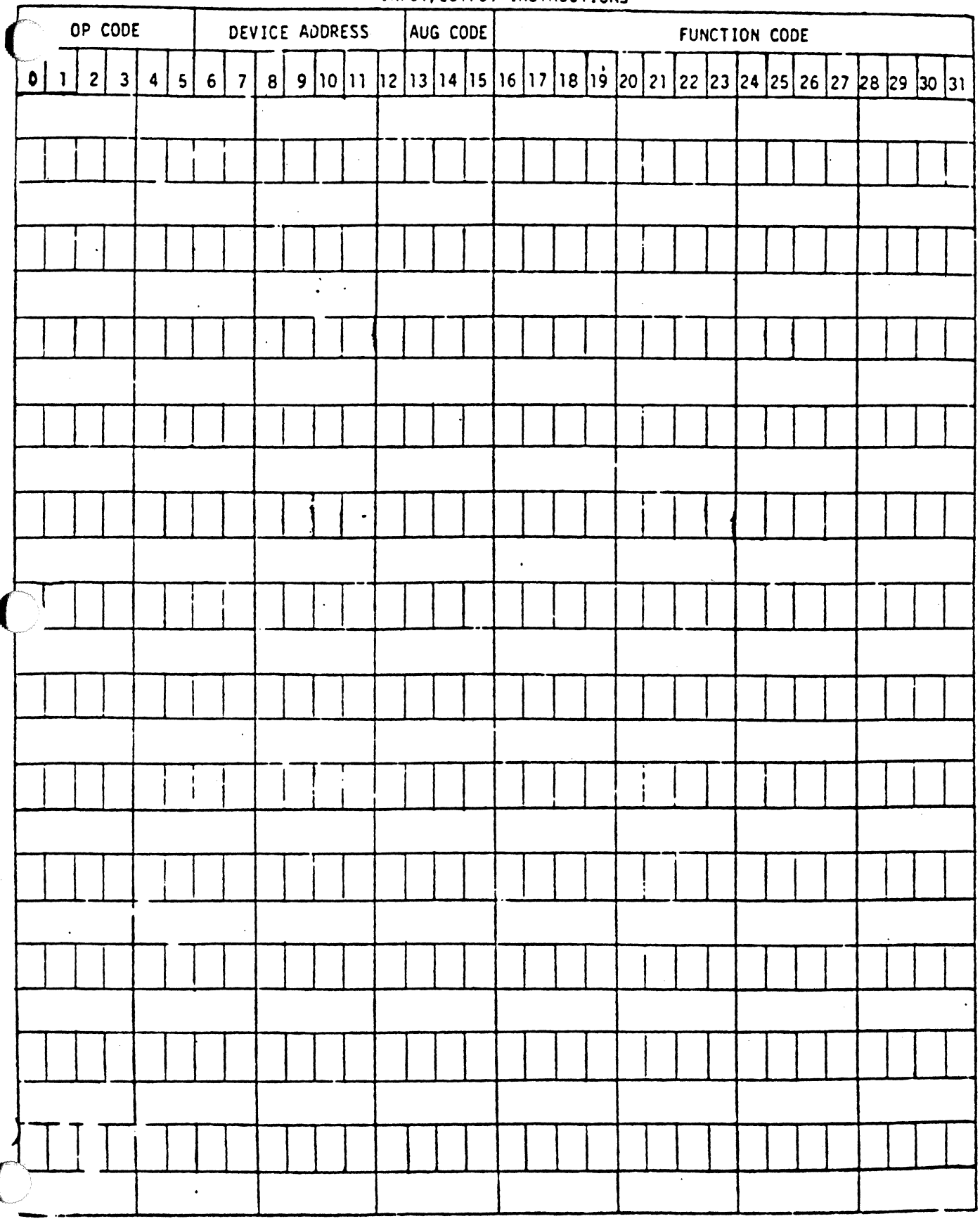

CAUSES A TRANSFER OF 16-BITS OF CONTROLLER STATUS INFORMATION TO THE MEMORY LOCATION SPECIFIED IN THE TCW DEDICATED LOCATION. THE MEANING OF EACH BIT IN THE 16-BIT STATUS HALFWORD DIFFERS ACCORDING TO DEVICE TYPE. SEE FIGURE 5-8.

CC2 = 0 STATUS TRANSFER WAS PERFORMED
CC2 = 1 STATUS TRANSFER WAS NOT PERFORMED
CC4 = 1 CONTROLLER IS ABSENT OR POWERED OFF

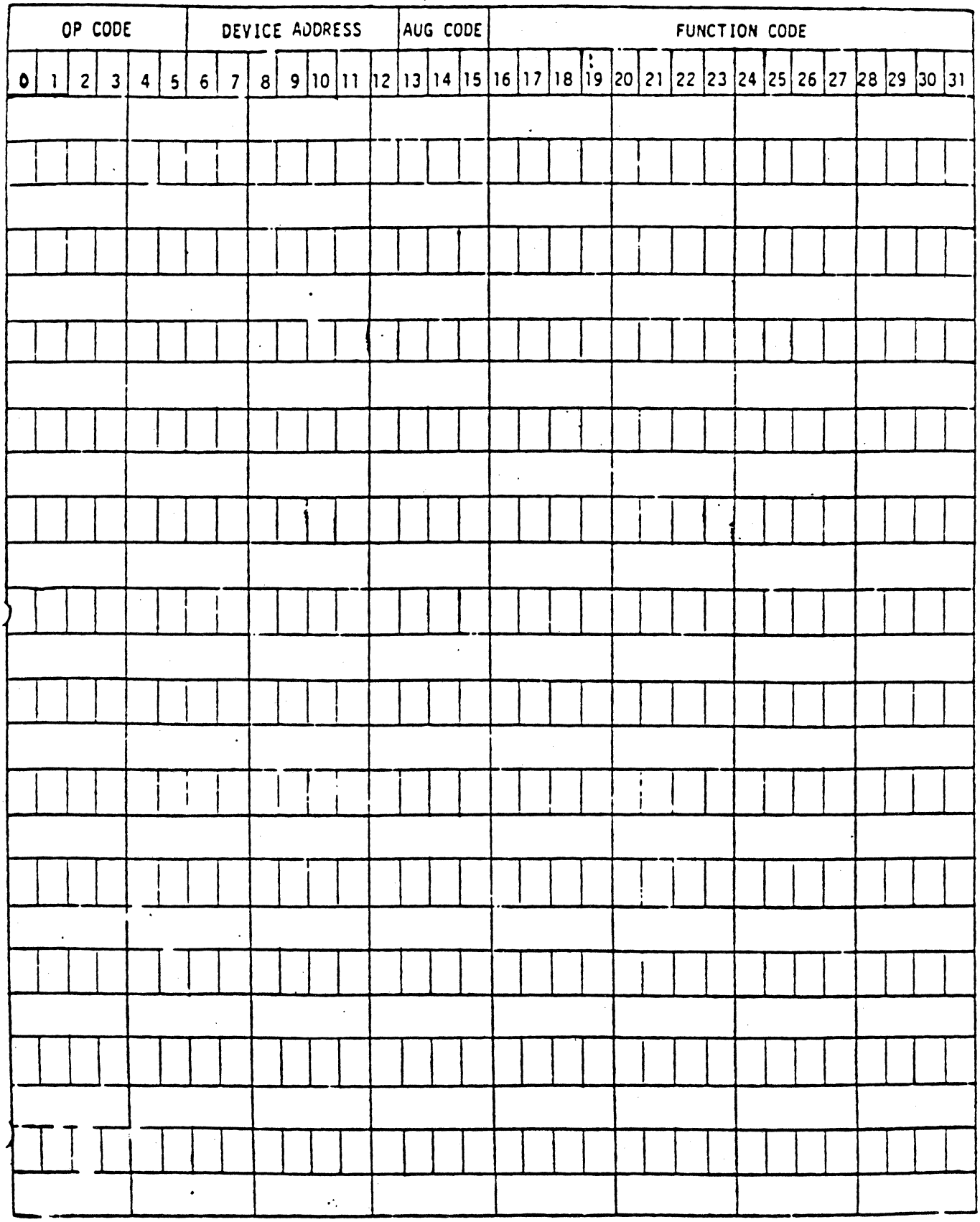
| UPPER HW | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOWER HW | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| LINE PRINTER | 0 | PROG VIOL | DEV INOP | 0 | 0 | 0 | 0 | 0 | 0 | BOF | 0 | 0 | 0 | DEV BUSY | 0 | 0 |
| MAG TAPE | 0 | PROG VIOL | DEV INOP | VRC ERROR | 0 | REW IN PROG | CRC LRC | 0 | 0 | EOT | BOT | EOF | 0 | DEV BUSY | FILE PROT VIO | ODD REC LGT |
| MOVING-HEAD DISC | 0 | PROG VIOL | DEV INOP | UNCORR DATA ERROR | 0 | FILE UN-SAFE | SEEK IN PROG | CORR DATA ERROR | 0 | 0 | ADDR ERROR | 0 | 0 | 0 | 0 | SEEK TRACK ERROR |
| FIXED-HEAD DISC | 0 | PROG VIOL | DEV INOP | CHK SUM | 0 | 0 | 0 | 0 | 0 | 0 | SECTOR ERROR | 0 | MUX BSY (DUAL CPU) | 0 | FILE PROT VIO | SEEK TRACK ERROR |
| CARD READER/ PUNCH | 0 | 0 | FILE MARK RD | READ CHECK | 0 | STACKER FULL | PUNCH CHECK | HOPPER EMPTY | 0 | PICK FAILURE | TRANSMIT ERROR | INCORRECT LENGTH | UNRS CHAN END | ILLEGAL END | INT PEND | CHAN END |

THE STATUS HALFWORD IS STORED IN THE MEMORY HALFWORD SPECIFIED BY THE ASSOCIATED TRANSFER CONTROL WORD (TCW).

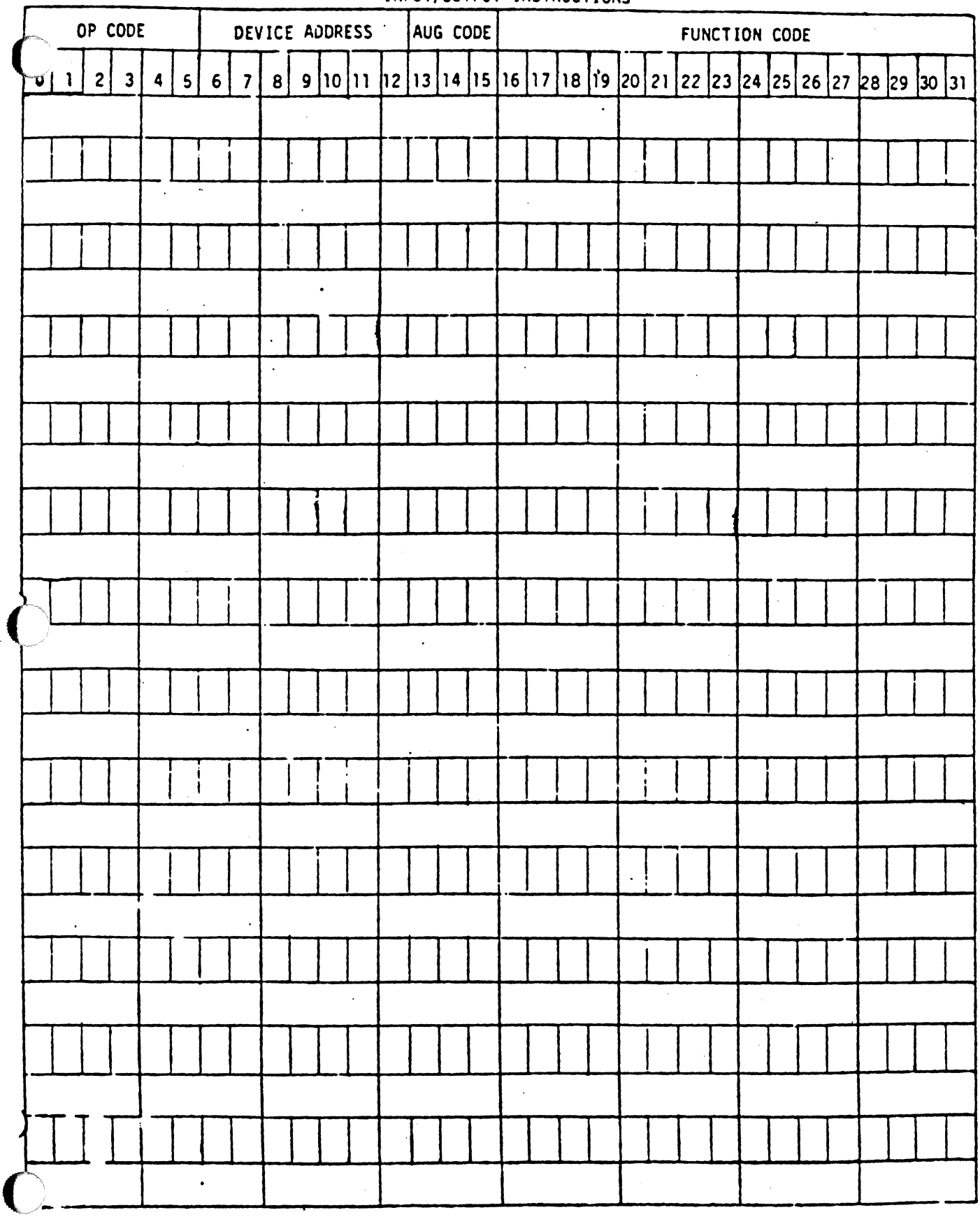
INPUT/OUTPUT INSTRUCTIONS



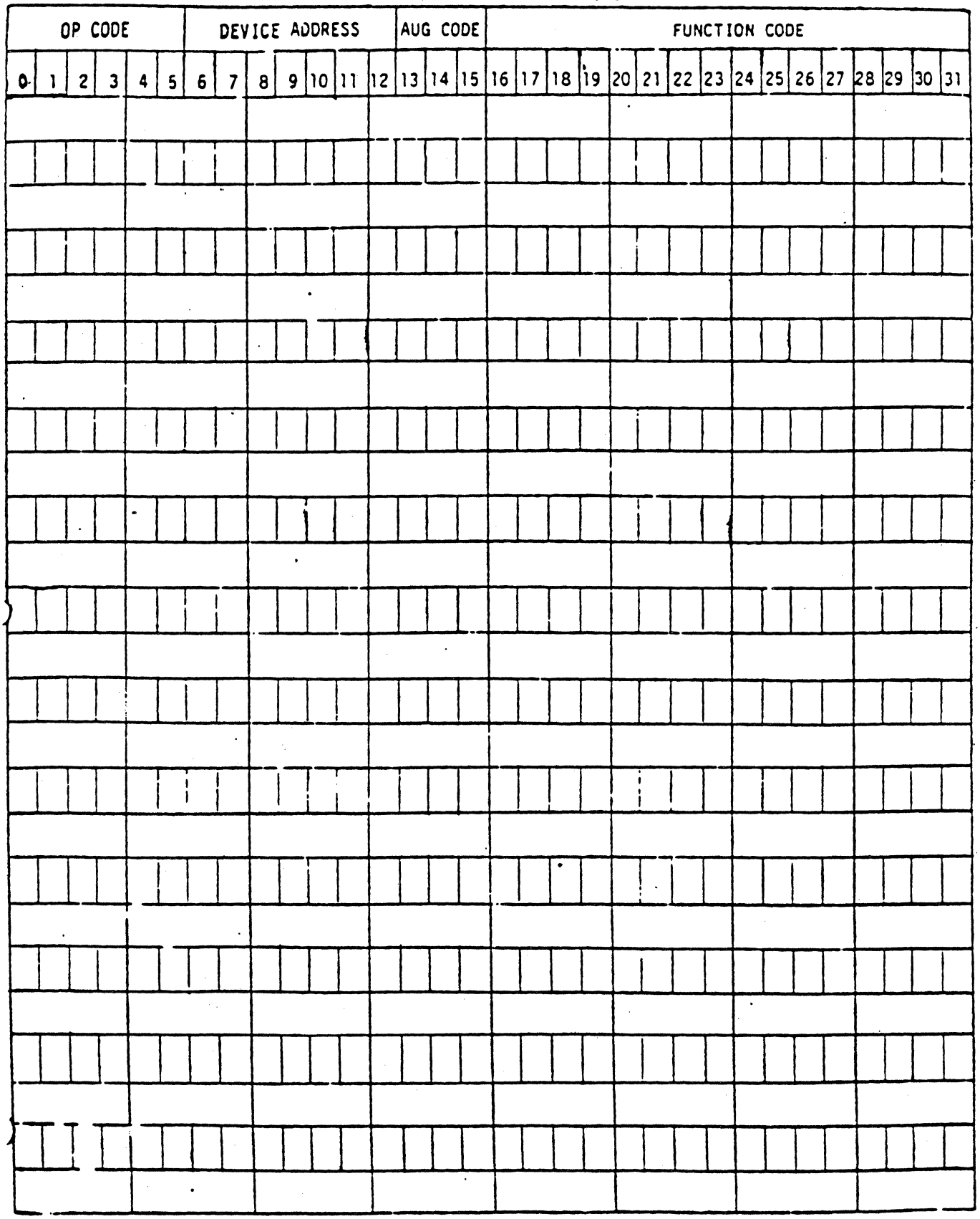
INPUT/OUTPUT INSTRUCTIONS



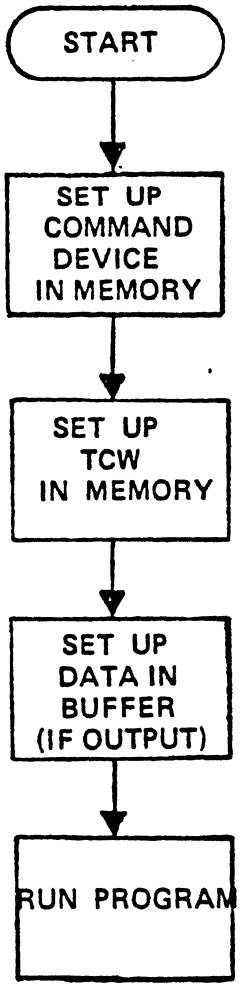
## INPUT/OUTPUT INSTRUCTIONS



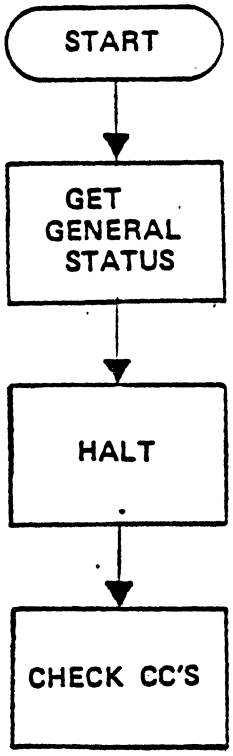
INPUT/OUTPUT INSTRUCTIONS



# OUTPUT TO TTY/CRT



# CHECK GENERAL STATUS OF MAG TAPE

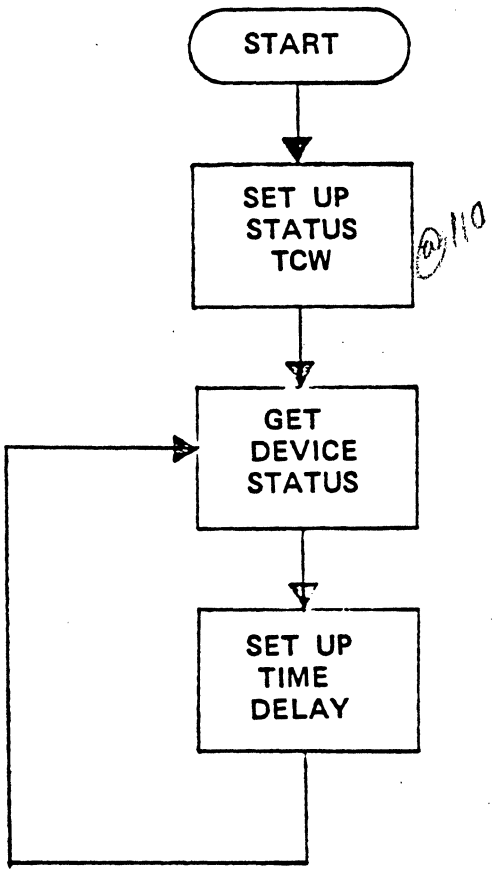



NOTES: CHECK STATUS WITH IOM ONLINE & DRIVE ONLINE

CHECK STATUS WITH IOM ONLINE & DRIVE OFFLINE

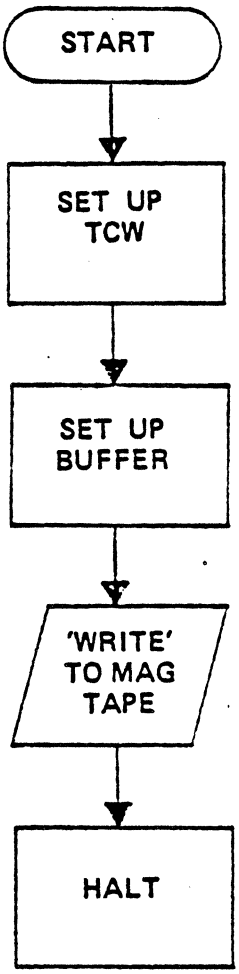
CHECK STATUS WITH IOM OFFLINE & DRIVE ONLINE

# CHECK DETAILED DEVICE STATUS

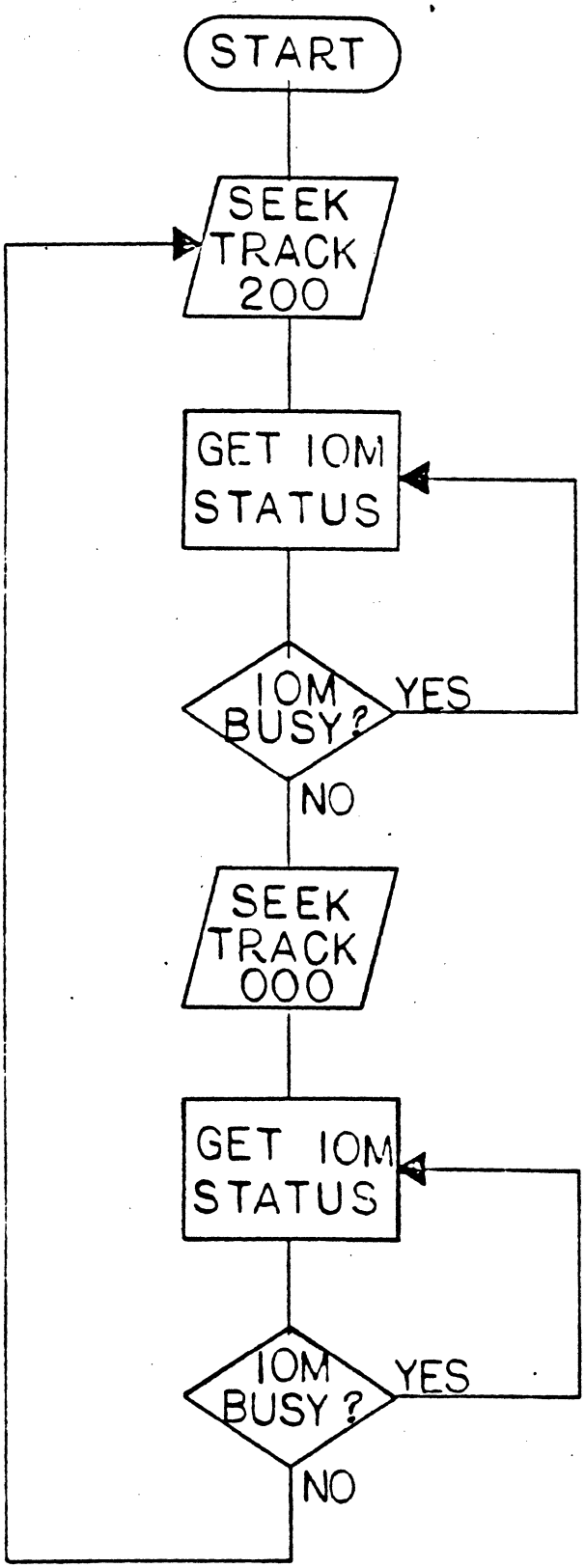



NOTE: EXECUTE EXTENDED FUNCTION 1 ON STATUS LOCATION TO MONITOR STATUS BITS.

# WRITE TO MAG TAPE

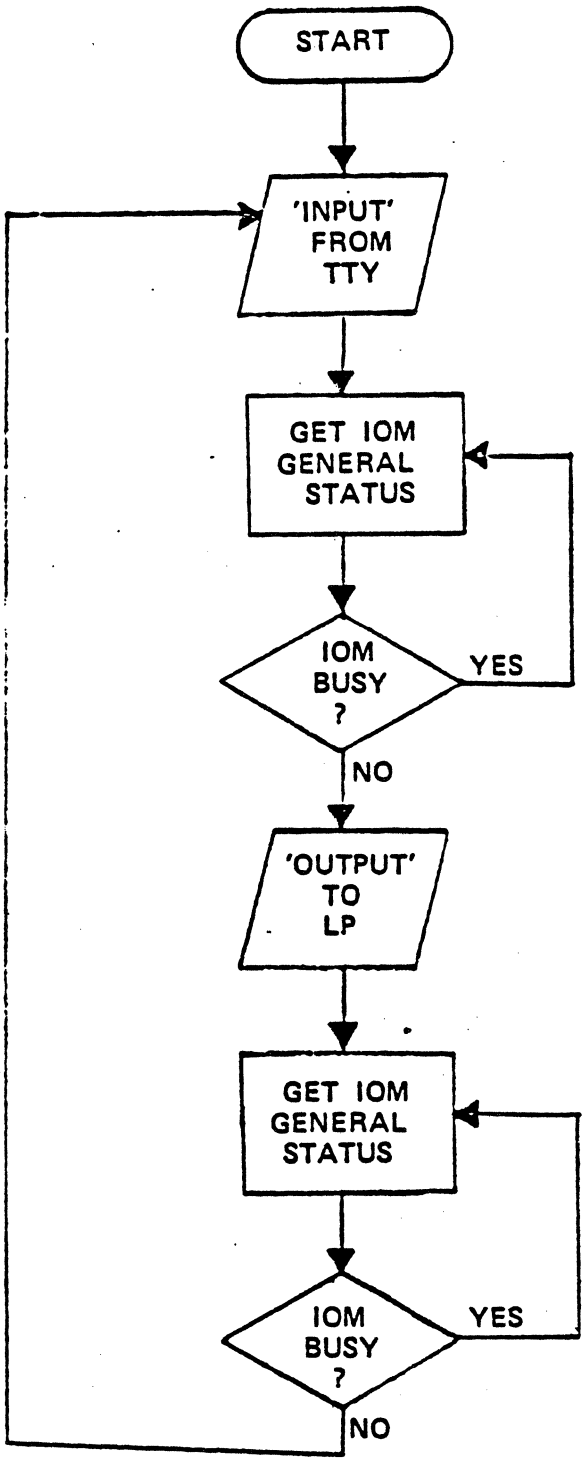


# DISC SEEKS WITH NO INTERRUPTS

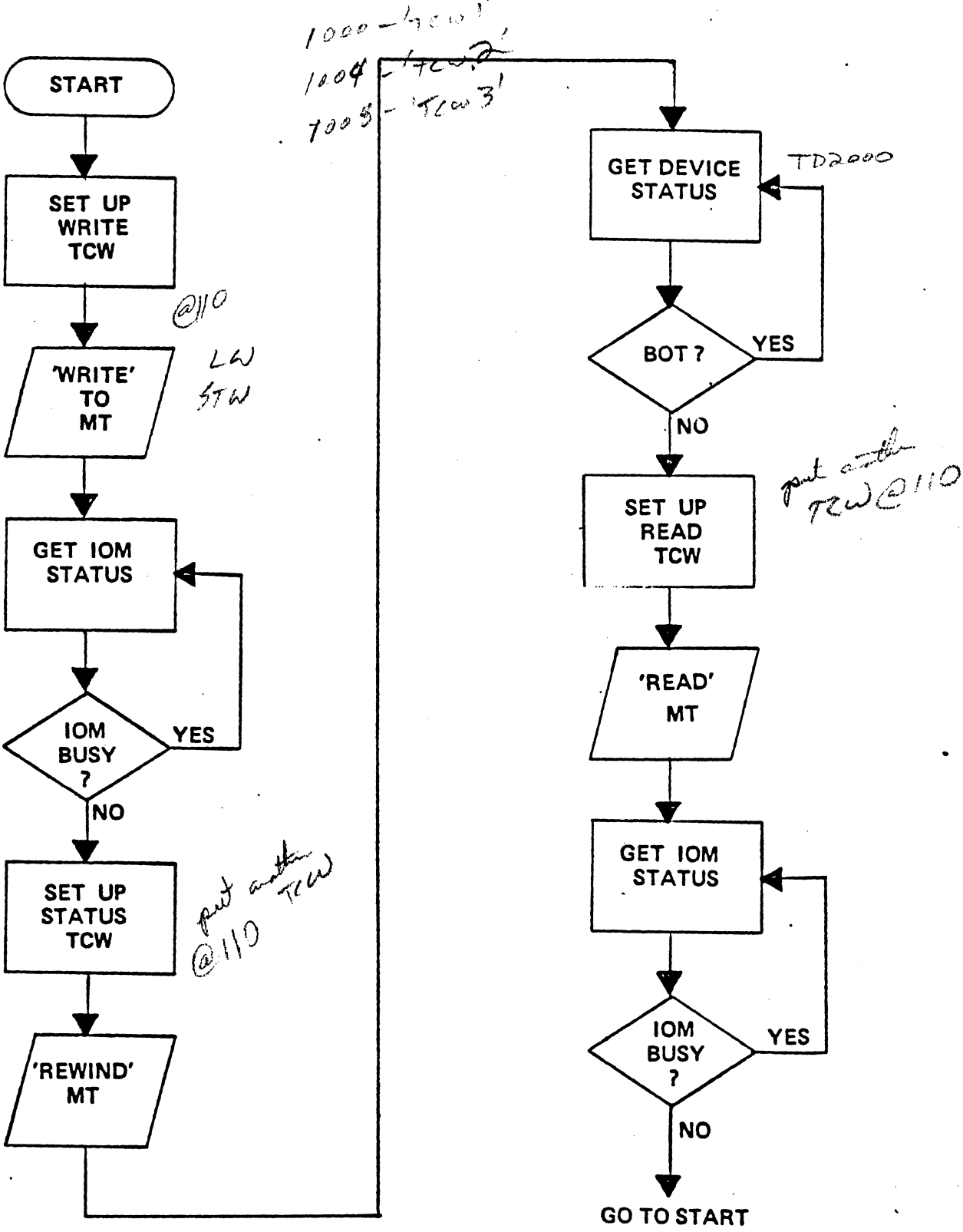


CR1

# INPUT FROM TTY - OUTPUT TO LP WITH NO INTERRUPTS



# WRITE, REWIND, READ MAG TAPE WITH NO INTERRUPTS



DAY 3

INSTRUCTION SET

I/O PROGRAMMING WORKSESSION

REFERENCES:

32/70 SERIES
REFERENCE MANUAL
SECTION FIVE

WORKBOOK

Objective: After completing this worksession, you will be able to use the 32/70 Series reference manual as a tool in generating machine language programs as an aid in troubleshooting INPUT/OUTPUT problems.

1. List the fields of a C/D instruction and define each.

b-12 Device Address - The 7 bit field identify Logical address ie. LP=7A CRT=7E 16-19 Instruction Function
16-31 Command Code - Input/Output 13-15 = Augment Code

2. How is the "device address" of a C/D instruction used?

Identify device by a channel Address in scratchpad Logical to physical Address

3. Code up a C/D instruction for a Mag Tape read.

| 1111 | 1100 | 1000 | 0110 | 1011 | 0000 0000 0000 |
|---|---|---|---|---|---|
| F | C | 8 | 6 | B | 0 0 0 |

4. List the fields of a TCW and define each.

0-11 - define number of transfers (4096 max)
30,31,12 F bit - type of format for each transfer
13-29 designate the memory location for each transfer (buffer)

5. Code up a TCW to transfer 16 bytes starting at address X'04000'.

| 0000 | 0001 | 0000 | 1000 | 0100 | 0000 | 0000 | 0000 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 8 | 4 | 0 | 0 | 0 |

6. What are the TCW dedicated locations for MHD, MT, TLC?

108, 110, 130, 134, 13C

7. When must the TCW be in the TCW dedicated location?

Before execution of CD instruction TD 2000

8. When does an IOM normally generate an SI? When I/O terminates

9. List the fields of a T/D instruction and define each.

Device Address ie CRT = 7E LP = 7A (6-12)

Test Code = 8000, 4000, 2000 (16-18)

Augment Code separates TD & CD

10. Code up a T/D 2000 to store status for Mag Tape.

1111 1100 1000 1010 0010 0000 0000 0000

F C 8 5 2 0 0 0

11. When checking general or IOM status, what defines the status? Condition Codes in PSD, PSW

12. On execution of a T/D 2000, where is the device status stored? 16 bit status Halfword stored in memory location specified by TCW

13. Code up a TCW to store Device Status of a Mag Tape in the Left Half Word of Memory location X'2000'.

0000 0000 0001 0010 0000 0000 0001

0 0 1 0 2 0 0 1

14. List the different classes of I/O.

0, 1, 2, 3, D, E, F

15. What is the difference between a Multiplexer Channel and a Selector Channel?

A selector channel only one channel is active at a time. A multiplexer allows more than one channel at a time to be active

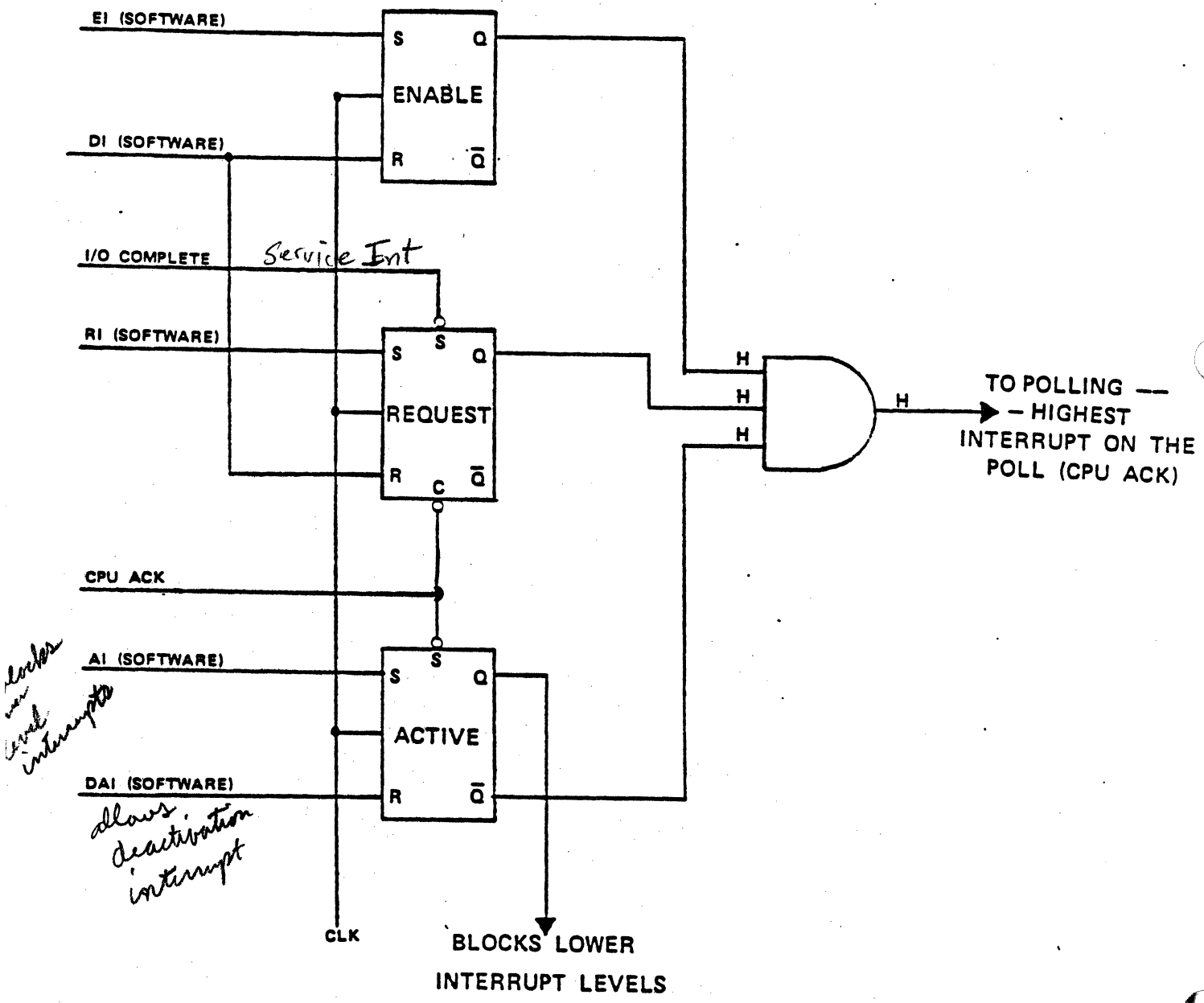
# I/O INTERRUPT CHART

Scratchpad address

| IOM TYPE | DEV ADDR | S.I. INT LVL | DED IVL ADDR | TCW = IVL-40 |
|---|---|---|---|---|
| FHD | 00 | 14 | 140 | 100 |
| | 04 | 15 | 144 | 104 |
| MHD | 08 | 16 | 148 | 108 |
| | 0C | 17 | 14C | |
| MTC | 10 | 18 | 150 | |
| | 18 | 19 | 154 | |
| GPMC | 20 | 1A | 158 | |
| | 30 | 1B | 15C | |
| HSD | 40 | 1C | 160 | |
| | 50 | 1D | 164 | |
| ADS | 60 | 1E | 168 | |
| | 70 | 1F | 16C | |
| CR | 78 | 20 | 170 | |
| LP | 7A | 21 | 174 | |
| | 7X | 22 | 178 | |
| TTY/CRT | 7E | 23 | 17C | |

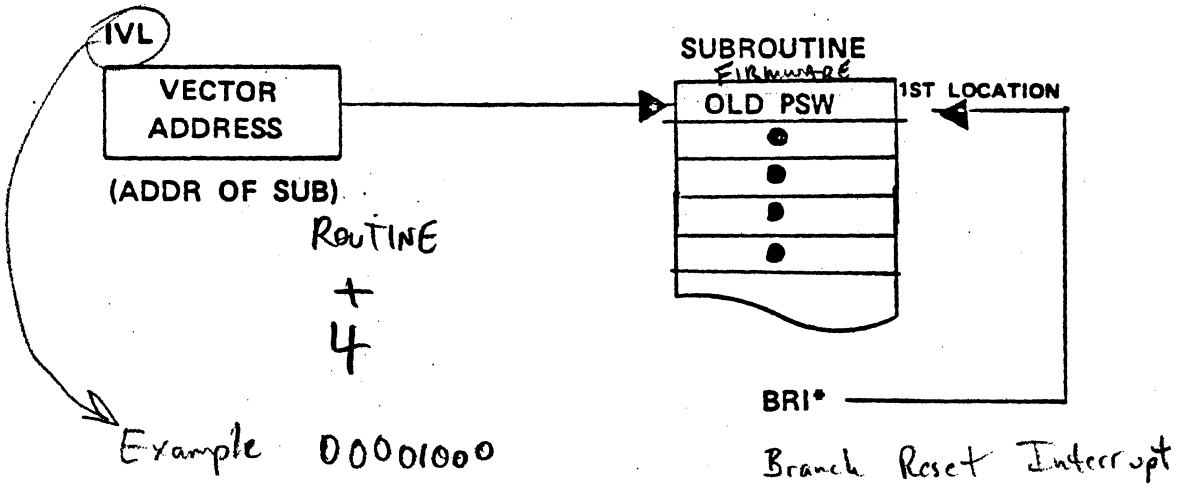
BASED ON
S.I. INT LVL

# INTERRUPT STATES

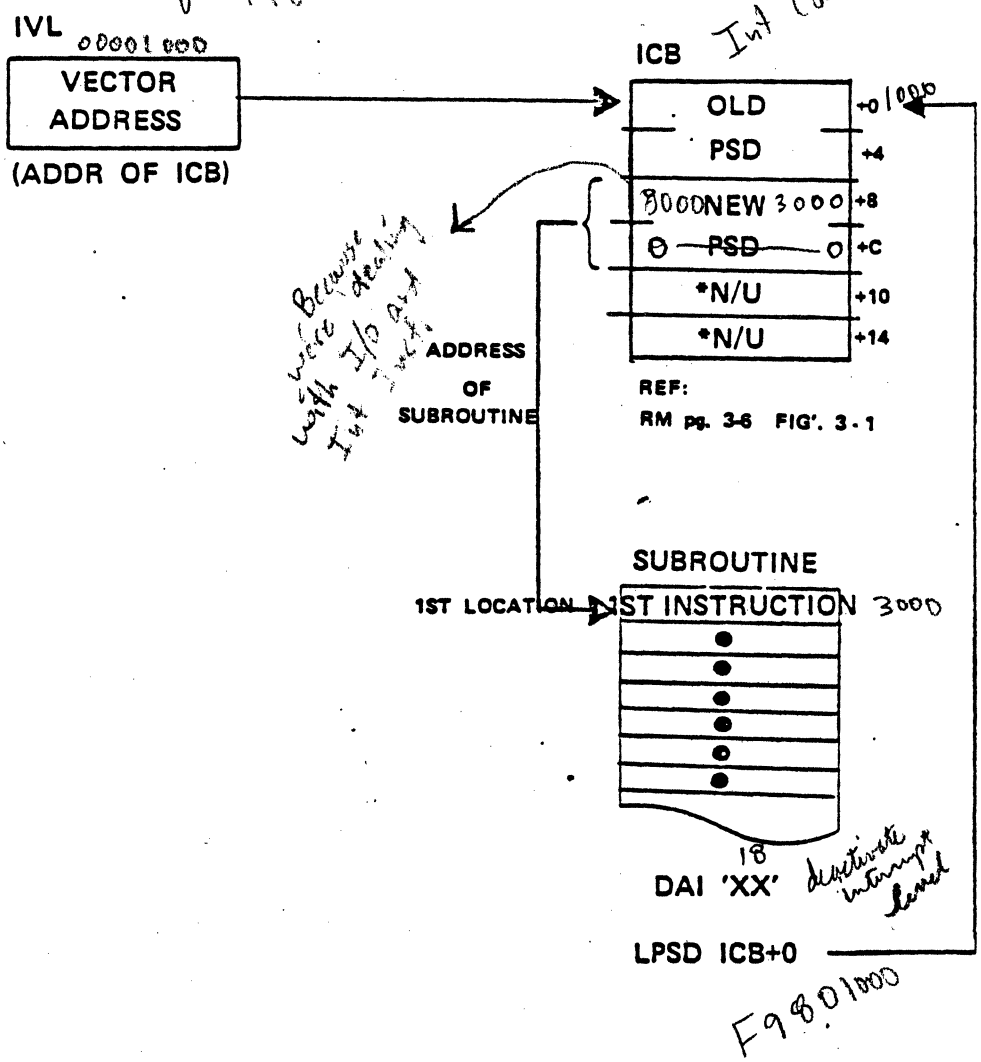


# PSW MODE

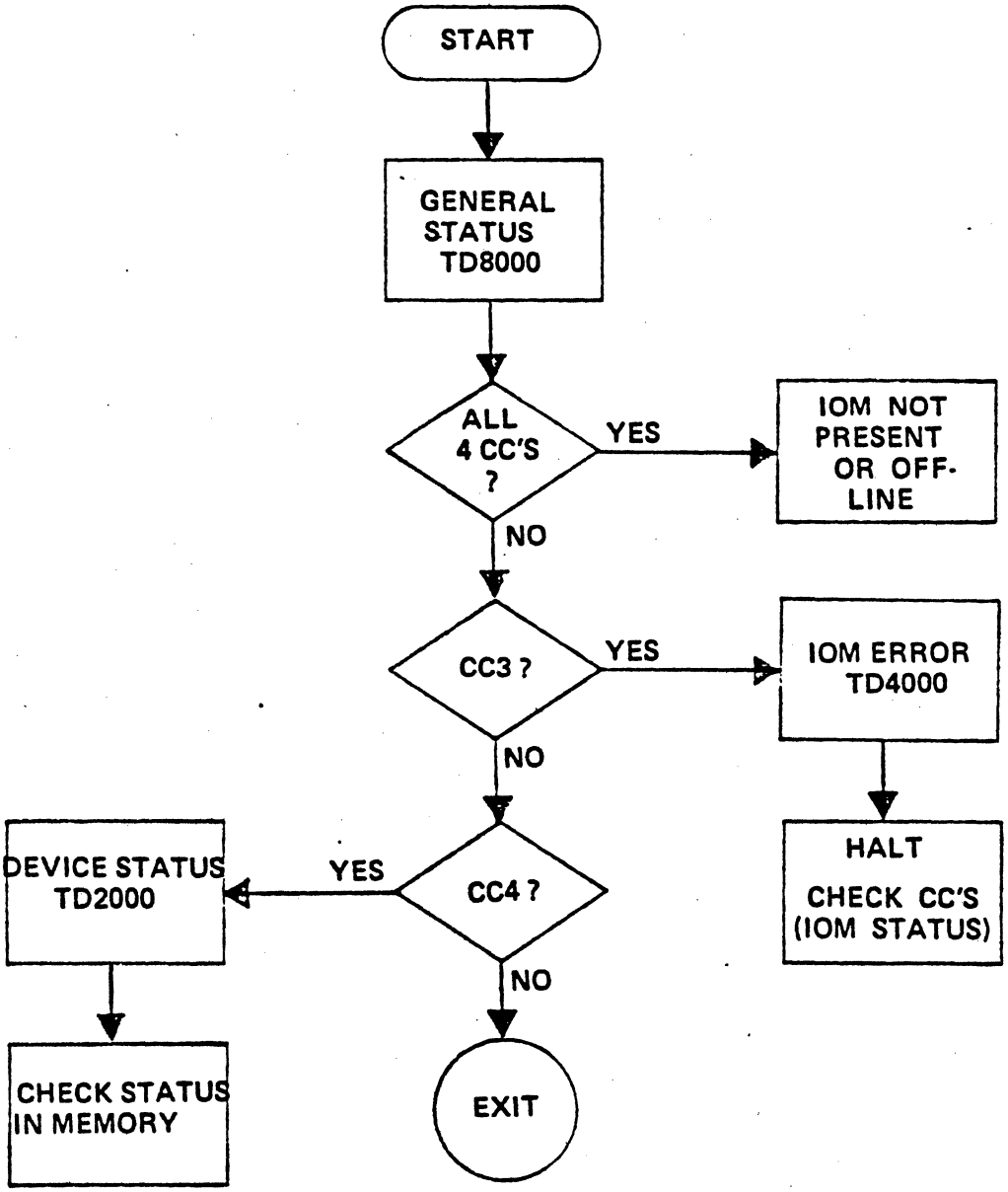
## I/O INTERRUPT PROCESSING



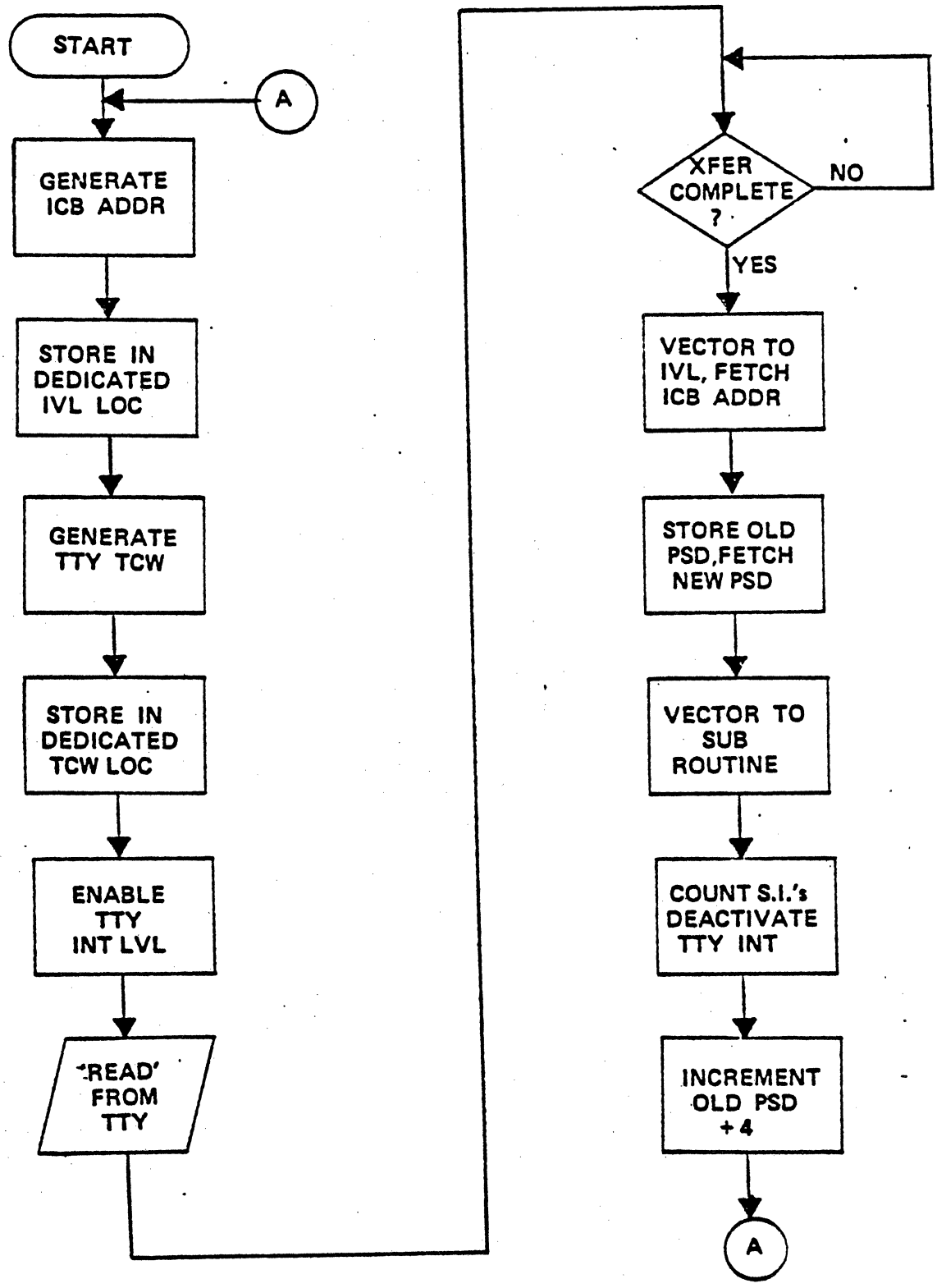
# PSD MODE
# I/O INTERRUPT PROCESSING



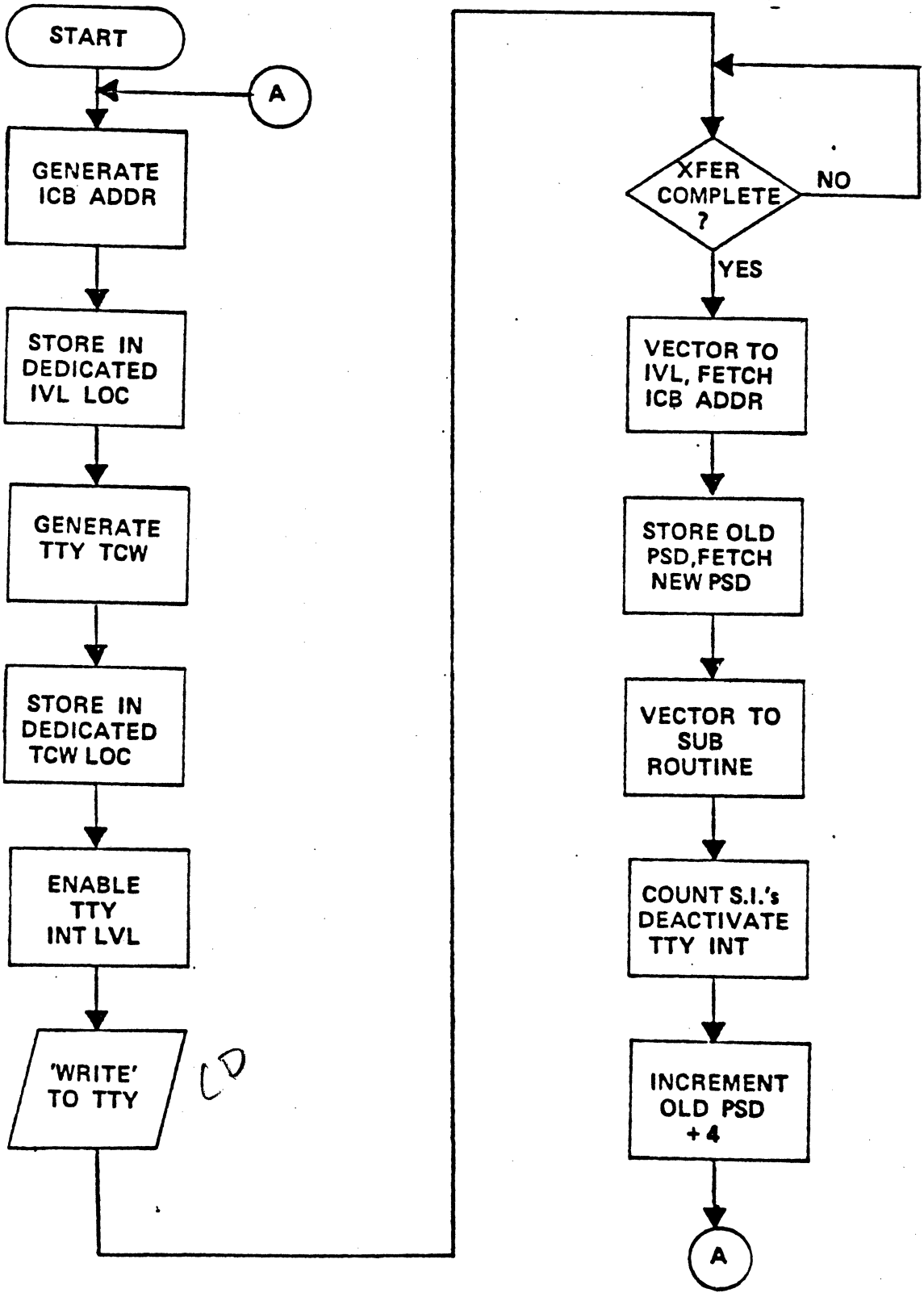
# LEVELS OF TEST DEVICE (TD)



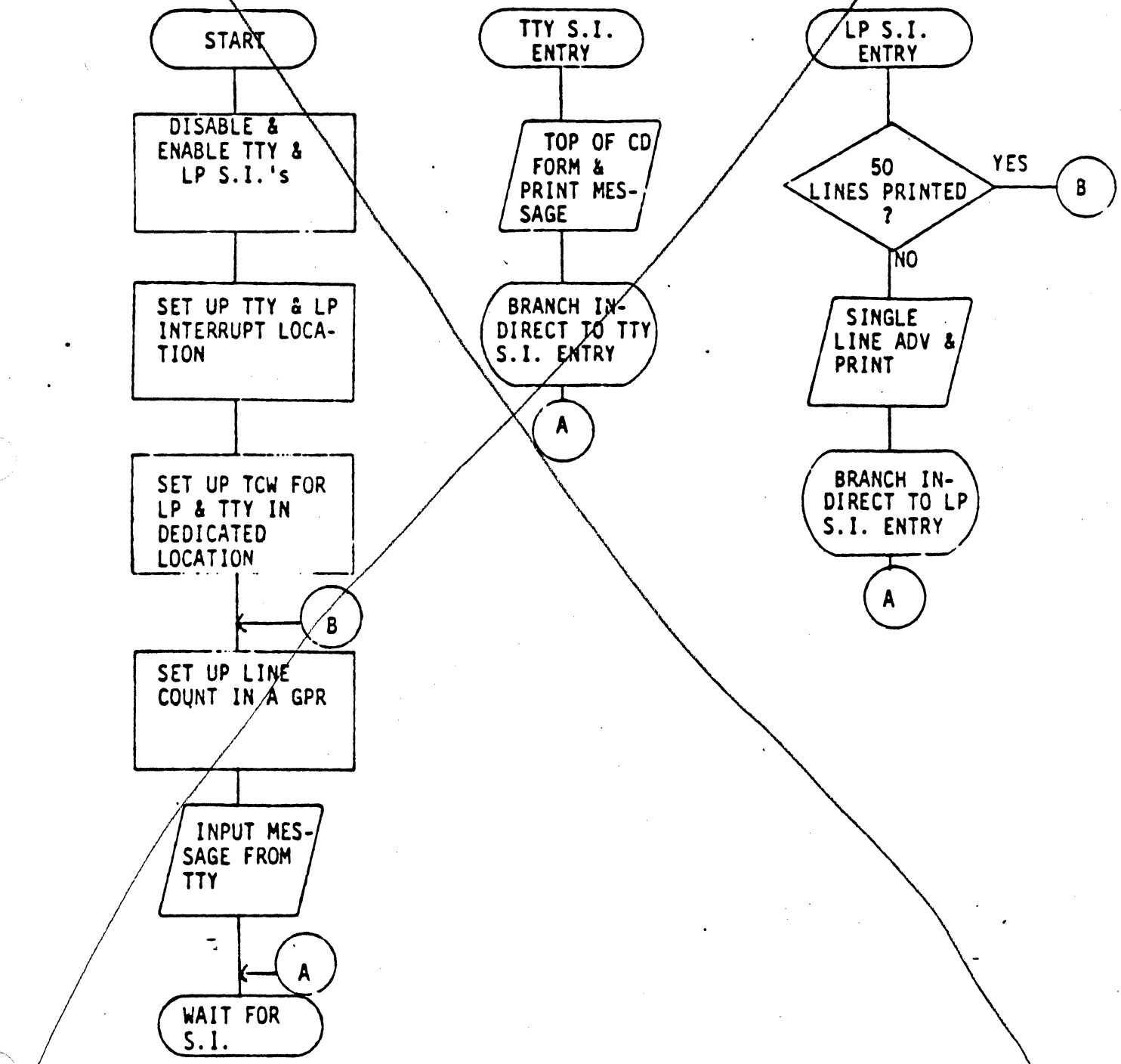
# INPUT FROM CONSOLE



# OUTPUT TO CONSOLE



# INPUT FROM TTY & OUTPUT TO LINE PRINTER



DAY 4

# INSTRUCTION SET

## I/O INTERRUPTS - WORKSESSION

REFERENCES:

32/70 SERIES
TECHNICAL MANUAL
SECTIONS III & V

32/70 SERIES
REFERENCE MANUAL
SECTIONS III & IV

Objectives: State the basic purpose of I/O interrupts.

Become familiar with I/O interrupt philosophy.

Identify and list steps required to process I/O interrupts.

Overview: I/O Interrupts

System is interrupt driven, an I/O interrupt request can be caused by any of the following:

A. <u>I/O Termination</u> - A normal or abnormal termination will cause an interrupt request.

1. Normal Termination - completion of a block transer. (Transfer CNT = 0)
2. Abnormal Termination - Device terminates prematurally because of end of record or Inoperable condition.

B. Software (Interrupt Control Instructions)

C. Each I/O channel can handle its Device Interrupts (S.I.'s).

D. 16 interrupt levels dedicated to I/O.

E. Each IOM has its own self contained interrupt generating logic.

1. The interrupt vector location (IVL) is dedicated to:

   (a) The device address of the IOM.
   (b) The interrupt priority of the IOM.
   (c) The physical address of the IOM.

2. What are the IVL dedicated locations for MT, MHD, TLC?

   MT = 150
   MHD = 148
   TLC = 170, 174, 17C

3. What does the IVL contain? Address of ICB or Subroutine

4. What does an I/O (non Class F) ICB contain? OLD PSD / NEW PSD

5. When should the new PSD be stored in the ICB? Before enabling interrupts

6. What bit(s) in the new PSD should be set? The privelege bit and PC address bits

7. What are the 3 states of an interrupt - how are they set/reset? Enable, Request, Active
   Software / Software  Service Int / Software  Software / Software

8. What interrupt control instruction must be used to allow servicing of an interrupt? Enable Interrupt

9. What interrupt control instruction will clear any pending interrupt request to that level? Disable Interrupt

10. When an interrupt goes active, what interrupt levels are blocked? Interrupts that interrupt level and all levels underneath below

11. In the PSD mode what instruction(s) are used to exit an interrupt servicing routine? Deactivate Int. L PSD

12. In the PSW mode what instruction(s) are used to exit an interrupt servicing routine? BRI

13. What interrupt levels are dedicated for I/O controllers? 14-23

14. What instruction should be executed prior to enabling an interrupt? Why? Disable Interrupts To allow new interrupts in.

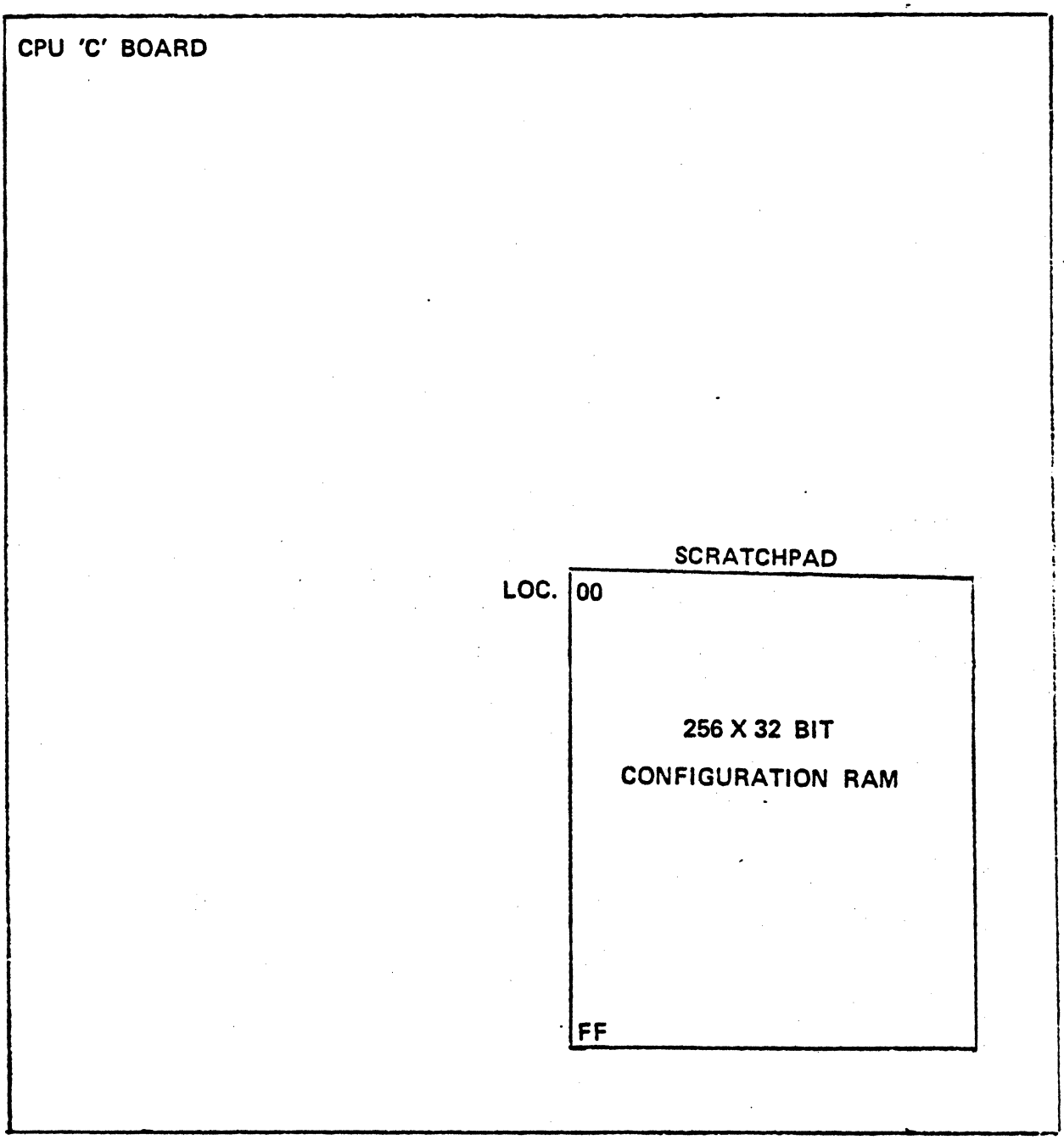
# DAY 5

# SECTION 5

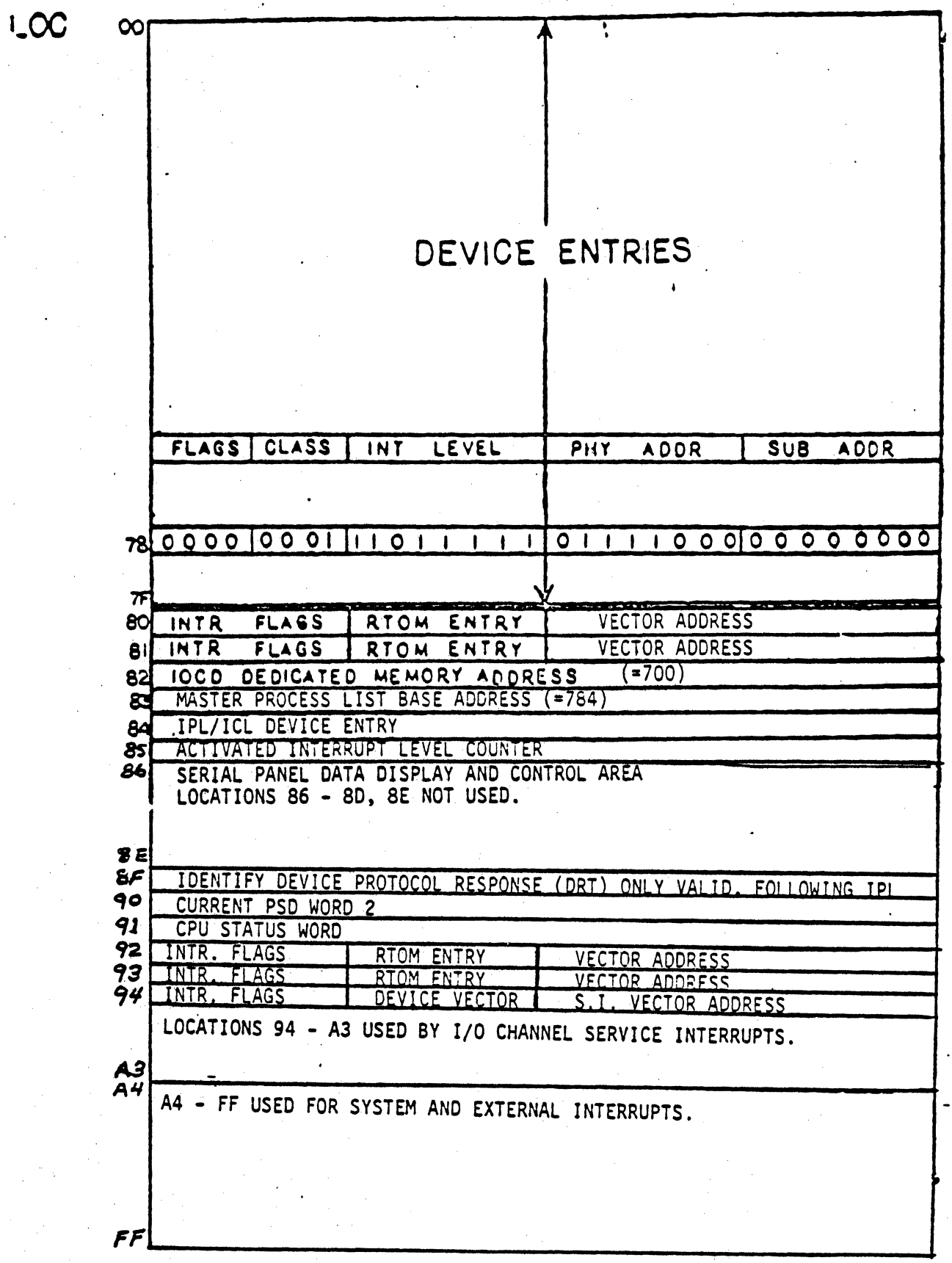
# SCRATCHPAD

# TRAPS & INTERRUPTS

# CPU SCRATCHPAD LOCATION



# SEL 32 SCRATCHPAD LAYOUT



# SCRATCHPAD DUMP

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 000300(R00000) | 0E6B0100 | 0E6B0101 | 0E6B0102 | 0E6B0103 | 0E6A0400 | 0E6A0401 | 0E6A0402 | 0E6A0403 |
| 000320(R00008) | 0E690800 | 0E690801 | 0E690802 | 0E690803 | 0E680C00 | 0E680C01 | 0E680C02 | 0E680C03 |
| 000340(R00010) | 0E671000 | 0E671001 | 0E671002 | 0E671003 | 0E671004 | 0E671005 | 0E671006 | 0E671007 |
| 000360(R00018) | 0E661800 | 0E661801 | 0E661802 | 0E661803 | 0E661804 | 0E661805 | 0E661806 | 0E661807 |
| 000380(R00020) | 0E652000 | 0E652001 | 0E652002 | 0E652003 | 0E652004 | 0E652005 | 0E652006 | 0E652007 |
| 0003A0(R00028) | 0E65200B | 0E652009 | 0E65200A | 0E65200B | 0E65200C | 0E65200D | 0E65200E | 0E65200F |
| 0003C0(R00030) | 0E643000 | 0E643001 | 0E643002 | 0E643003 | 0F643004 | 0E643005 | 0E643006 | 0E643007 |
| 0003E0(R00038) | 0E64300B | 0E643009 | 0E64300A | 0E64300B | 0E64300C | 0E64300D | 0E64300E | 0E64300F |
| 000400(R00040) | 0E634000 | 0E634001 | 0E634002 | 0F634003 | 0E634004 | 0F634005 | 0E634006 | 0E634007 |
| 000420(R00048) | 0F63400B | 0E634009 | 0E63400A | 0E63400B | 0E63400C | 0E63400D | 0F63400E | 0E63400F |
| 000440(R00050) | 0E625000 | 0E625001 | 0E625002 | 0F625003 | 0F625004 | 0E625005 | 0E625006 | 0E625007 |
| 000460(R00058) | 0E62500B | 0E625009 | 0E62500A | 0F62500B | 0F62500C | 0E62500D | 0E62500E | 0E62500F |
| 0004B0(R00060) | 0E616000 | 0F616001 | 0F616002 | 0E616003 | 0F616004 | 0E616005 | 0E616006 | 0E616007 |
| 0004A0(R00068) | 0E61600B | 0E616009 | 0E61600A | 0F61600B | 0E61600C | 0E61600D | 0E61600E | 0E61600F |
| 0004C0(R00070) | 0E607000 | 0E607001 | 0F607002 | 0E607003 | 0F607004 | 0E607005 | 0E607006 | 0E607007 |
| 0004F0(R00078) | 015F7800 | 00000000 | 005E7802 | 00000000 | 005D7C02 | 005D7C03 | 025C7801 | 03547904 |
| 000500(R00080) | 009F00F0 | 009E00F8 | 00000700 | 00000764 | 00001000 | 00000000 | 00000000 | BFFFFFFF |
| 000520(R00088) | BFFFFFFF | BFFFFFFF | BFFFFFFF | BFFFFFFF | 88FFFFFC | 00000000 | 00000000 | 00000000 |
| 000540(R00090) | 00000000 | 00000001 | 009D00E8 | 009C00EC | 00000140 | 00040144 | 00080148 | 000C014C |
| 000560(R00098) | 80100150 | 00180154 | 00200158 | 0030015C | 00400160 | 00500164 | 00600168 | 0070016C |
| 000580(R000A0) | 00780170 | C17A0174 | 007C0178 | C17E017C | 009B0190 | 009A0194 | 00990198 | 0098019C |
| 0005A0(R000A8) | 009601A0 | 009701A4 | 009501A8 | 009401AC | 009301B0 | 009201B4 | 009101B8 | 009001BC |
| 0005C0(R000B0) | 00AF01C0 | 00AE01C4 | 00AD01C8 | 00AC01CC | 00AB01D0 | 00AA01D4 | 00A901D8 | 00A801DC |
| 0005F0(R000B8) | 00A701E0 | 00A601E4 | 00A501E8 | 00A401EC | 00A301F0 | 00A201F4 | 00A101F8 | 00A001FC |
| 000600(R000C0) | 00BF0200 | 00BE0204 | 00BD0208 | 00BC020C | 00BB0210 | 00BA0214 | 00B90218 | 00B8021C |
| 000620(R000C8) | 00B70220 | 00B60224 | 00B50228 | 00B4022C | 00B30230 | 00B20234 | 00B10238 | 00B0023C |
| 000640(R000D0) | 00CF0240 | 00CE0244 | 00CD0248 | 00CC024C | 00CB0250 | 00CA0254 | 00C90258 | 00C8025C |
| 000660(R000D8) | 00C70260 | 00C60264 | 00C50268 | 00C4026C | 00C30270 | 00C20274 | 00C10278 | 00C0027C |
| 000680(R000E0) | 00DF0280 | 00DE0284 | 00DD0288 | 00DC028C | 00DB0290 | 00DA0294 | 00D90298 | 00D8029C |
| 0006A0(R000E8) | 00D702A0 | 00D602A4 | 00D502A8 | 00D402AC | 00D302B0 | 00D202B4 | 00D102B8 | 00D002BC |
| 0006C0(R000F0) | 00CF02C0 | 00CE02C4 | 00CD02C8 | 00CC02CC | 00CB02D0 | 00CA02D4 | 00C902D8 | 00C802DC |
| 0006E0(R000F8) | 00E702E0 | 00E602E4 | 00E502E8 | 00E402EC | 00E302F0 | 00E202F4 | 00E102F8 | 00E002FC |

## INITIAL PROGRAM LOAD (IPL)

Initialization and configuration of a 32/70 Series System is accomplished through the use of the Initial Program Load (IPL) sequence. This sequence initializes the system, sets up the I/O configuration, and boots in the operating system. The usual method of initializing the system is through the use of the card reader to read in a deck of cards containing the I/O device configuration and assigned interrupt organization. The IPL sequence is initiated by placing the Initial Configuration Load (ICL) deck of cards in the card reader, setting up of the address of the card reader on the system front panel, and depressing the IPL button on the system front panel.

It should be noted that if the mode jumper on the CPU is set up for the PSD mode, the CPU will come up in the PSD mode. If, when placing the address of the IPL device in the B-Display of the front panel, additional information is added, then the CPU can be made to come up in the PSW mode of operation. The procedure for establishing the PSW mode of operation is as follows:

1. If using either the parallel or serial front panel for data entry, add 8000 to the device address (sets bit 16 to One). For example, if the address of the card reader is 7800, then by the setting of bit 16 to One (or adding 8000), the resultant address becomes F800.

2. If using the serial front panel, entering a 55 plus the card reader address results in the CPU coming up in the PSW mode. The resultant address in the B-Display is then 00557800.

After the cards are read into the system, the SYSTEM RESET button is depressed, the address of the device (disc) containing the operating system is entered on the front panel, and the IPL button is again depressed, thereby booting in the operating system.

The Initial Configuration Load (ICL) deck of cards contains three basic record formats. The following sections provide descriptions for each format.

## FORMATS OF THE INITIAL CONFIGURATION LOAD (ICL)

Initial Configuration Load (ICL) records are read from a default or selected peripheral device. The ICL records are converted into information that is used to initialize the 256- x 32-bit Configuration RAM (CR) contained in the 32/70 Series Central Processor Unit (CPU). Information contained in the CR is used by the CPU to address and maintain the status of the 128 possible devices and the 112 possible interrupts.

Initial Configuration Load records must be in the following ASCII or Hollerith formats:

[redacted]

Format #1 *DEVXX=FCILCASA (,NN)

where:

*DEV defines that the record contains a controller definition entry.

XX is the hexadecimal address that will be used by macro level input/output instructions to address the controller.

= is a necessary delimiter. Each letter to the right of this delimiter represents one hexadecimal digit (four binary bits).

F flags used by the CPU for input/output emulation. Presently, this field is always zero.

C defines the class of controller being emulated. Presently, this field can contain one of the following values:

| | |
|---|---|
| 0 | = LINE PRINTER |
| 1 | = CARD READER |
| 2 | = TELETYPE |
| 3 | = INTERVAL TIMER |
| 4 | = PANEL |
| 5 to D | = Unassigned |
| E | = ALL OTHERS |
| F | = EXTENDED I/O (32/75 ONLY) |

IL is the hexadecimal interrupt level of the Service Interrupt (i.e., priority levels $14_{16}$ through $23_{16}$) for the defined controller.

CA is the hexadecimal controller address as defined by the hardware switches on the IOM.

SA is the lowest hexadecimal device subaddress used by the controller. This field is normally zero when more than one device is configured.

( ) denotes optional parameter.

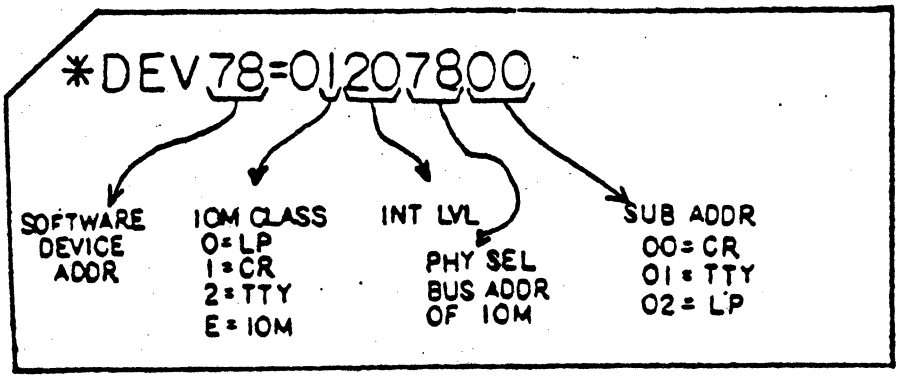
, is a delimiter that must be used when more than one device is configured.

NN is a 2-digit hexadecimal number that specifies the number of devices configured on the controller.

NOTE 1: The subaddress (SA) field must reflect the following for the Teletype, Line Printer, Card Reader (TLC) controller:

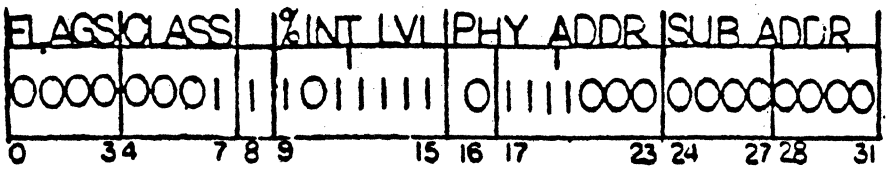
1. Card Reader is subaddress $0_{16}$.
2. Teletype is subaddress $1_{16}$.
3. Line Printer is subaddress $2_{16}$.

# DEVICE ENTRY CARD



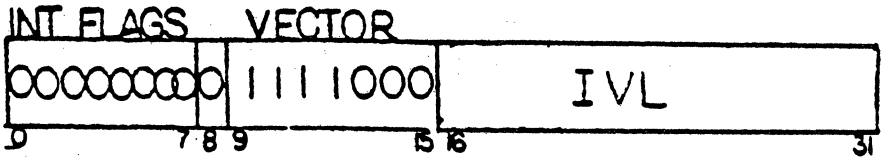

## SCRATCHPAD DEVICE ENTRY

LOC 78 IN SCRATCHPAD

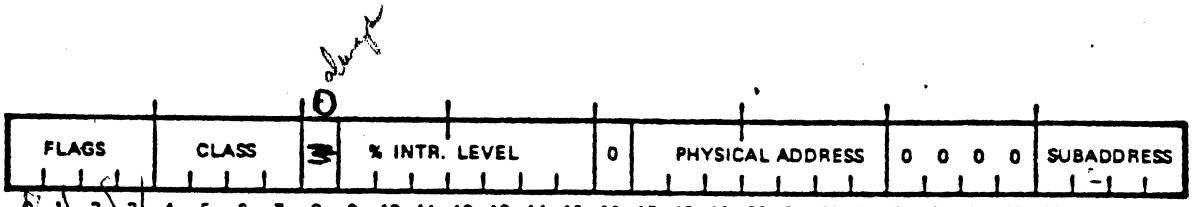



## SCRATCHPAD INTERRUPT ENTRY

LOC AØ IN SCRATCHPAD



# SCRATCHPAD DEVICE ENTRY




| FLAGS | |
|---|---|
| BIT | DEFINITION |
| 00 | PROGRAM VIOLATION |
| 01 | NOT ASSIGNED |
| 02 | NOT ASSIGNED |
| 03 | NOT ASSIGNED |

| CLASS | | | | |
|---|---|---|---|---|
| BITS | | | | DEFINITION |
| 04 | 05 | 06 | 07 | |
| 0 | 0 | 0 | 0 | CLASS 0 - 'TLC' LINE PRINTER |
| 0 | 0 | 0 | 1 | CLASS 1 - 'TLC' CARD READER |
| 0 | 0 | 1 | 0 | CLASS 2 - 'TLC' TELETYPEWRITER |
| 0 | 0 | 1 | 1 | CLASS 3 - RTOM INTERVAL TIMER |
| 1 | 1 | 1 | 0 | CLASS 'E' STANDARD I/O CONTROLLERS |
| 1 | 1 | 1 | 1 | CLASS 'F' I/O CONTROLLER |

| INTERRUPT LEVEL |
|---|
| BIT 08 = 0<br>BITS 09-15 PROVIDE THE ONES COMPLEMENT OF THE I/O CONTROLLER INTERRUPT LEVEL |

| PHYSICAL ADDRESS |
|---|
| BIT 16 = 0<br>BITS 17 - 23 PROVIDE THE I/O CONTROLLER (IOM) PHYSICAL ADDRESS, WHICH MUST MATCH THE IOM'S ADDRESS SWITCH CONFIGURATION |

| SUBADDRESS |
|---|
| BITS 28-31 PROVIDE THE SUBADDRESS OF SPECIFIC I/O DEVICE |

NOTES:
1. DEVICE ENTRIES ARE IN SCRATCHPAD ADDRESSES 00-7F.
2. CORRESPONDS TO THE SCRATCHPAD ADDRESS.
3. DEVICE ENTRY SCRATCHPAD ADDRESS CAN ALSO BE OBTAINED FROM THE INTERRUPT TABLE ENTRY, VECTOR FIELD (REFER TO FIGURE 5-50).

Format - Scratchpad Device Entry

# ICL INTERRUPT ENTRY FORMAT

**Format #2** *INTXX=RS

where:

*INT — defines that the record contains an interrupt definition entry.

XX — is the hexadecimal interrupt priority level that is to be emulated.

= — is a necessary delimiter. Each letter to the right of this delimiter represents one hexadecimal digit (four binary bits).

R — is the hexadecimal RTOM board number to which the interrupt XX is assigned.

S — is the hexadecimal subaddress on the RTOM board to which the interrupt XX is assigned (in one's complement).

NOTE 1: RTOM physical controller address $79_{16}$ is RTOM board number 1, address $7A_{16}$ is RTOM board number 2, etc.

NOTE 2: Real-Time Clock hardware is connected to subaddress $6_{16}$ on the RTOM board.

NOTE 3: Interval Timer hardware is connected to subaddress $4_{16}$ on the RTOM board.

NOTE 4: RTOM physical controller addresses must be $79_{16}$ or above. [redacted]

**Format #3** *END

where:

*END — is the last record of an Initial Configuration Load (ICL) deck. This record signifies the end of the load process.

EXAMPLES OF [redacted] INITIAL CONFIGURATION LOAD (ICL) RECORDS

*DEV04=0E140100,04

The device entry above specifies the following information:

1. The 32/55 input/output commands will address the controller as $4_{16}$.
2. The ",04" is an optional parameter that specifies that there are $4_{16}$ devices on the controller. There will be four entries defined in the Configuration RAM (CR). The 32/55 input/output commands will address the devices as $4_{16}$, $5_{16}$, $6_{16}$, and $7_{16}$.

3. The controller is an "E" class controller.
4. The priority of the Service Interrupt (SI) is $14_{16}$.

Assigning a priority to a controller has the following implications:

a. The Transfer Interrupt location for priority $14_{16}$ is $100_{16}$.

b. The Service Interrupt vector location for priority $14_{16}$ is $140_{16}$.

c. The emulation IOCD will be stored at location $700_{16}$.

d. The interrupt control instructions (i.e., DI, EI, RI, AI, DAI) will control the interrupt on the controller by addressing priority $14_{16}$.

5. The physical address of the controller is $01_{16}$.

[redacted]

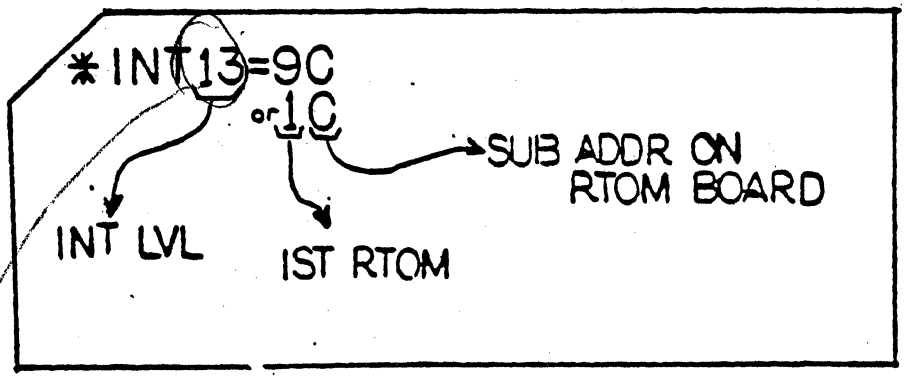
*INT28=16 NOTE: 28 + 80 = A8 (location in scratchpad)

The interrupt entry above specifies the following information:

1. The 32 interrupt control instructions (i.e., DI, EI, RI, AI, DAI) will control the interrupt on the RTOM by addressing priority $28_{16}$.
2. The number of the RTOM board is 1.
3. The subaddress on the RTOM board is $6_{16}$. (one's complement)

A sample Initial Configuration Load (ICL) Deck is given in Figure 1.

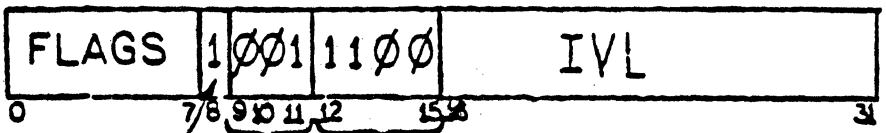
# INTERRUPT ENTRY CARD




No Device Entry

## SCRATHPAD INTERRUPT ENTRY (RTOM)

LOC 93 IN SCRATCH PAD

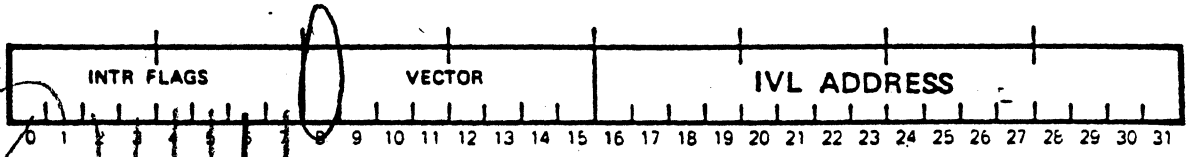



RTOM ENTRY

3 LSBs of RTOM PHY ADDR (79)

4 MSBs of SUB ADDR (C)

# SCRATCHPAD INTERRUPT TABLE ENTRY




| INTERRUPT FLAGS | |
|---|---|
| BIT | DEFINITION |
| 0 | RAM LOADED / ON CONTROLLER |
| 1 | I/O IN PROGRESS |
| 2 | 2 IOCD'S REQUIRED (HSD maybe) |
| 3 | ARITHMETIC EXCEPTION PENDING |
| 4 | REFETCH WITH IGNORE ADDRESS STOP |
| 5 | ACTIVATE/DEACTIVATE INTERRUPT ISSUED<br>0 = DEACTIVATE ISSUED<br>1 = ACTIVATE ISSUED |
| 6 | REQUEST INTERRUPT ISSUED |
| 7 | ENABLE/DISABLE INTERRUPT ISSUED<br>0 = DISABLE ISSUED<br>1 = ENABLE ISSUED |

**VECTOR**

BIT 08 = 0; BITS 09-15 PROVIDE DEVICE VECTOR (IOM ENTRY)
BIT 08 = 1; BITS 09-15 PROVIDE THE RTOM ENTRY. 9, A, B, C, D . . . F.

**IOM ENTRY:**

IF BIT 08 = 0; BITS 09-15 PROVIDE THE SCRATCHPAD ADDRESS OF THE DEVICE ENTRY (REFER TO FIGURE 5-51 FOR FORMAT) AND ALSO CORRESPOND TO THE COMMAND DEVICE OR TEST DEVICE INSTRUCTION DEVICE ADDRESS FIELD.

**RTOM ENTRY:**

IF BIT 08 = 1; BITS 09 11 PROVIDE THE THREE LEAST SIGNIFICANT BITS OF THE RTOM PHYSICAL ADDRESS. (FOUR MOST SIGNIFICANT BITS OF PHYSICAL ADDRESS ARE (1111

BITS 12-15 PROVIDE THE FOUR BITS OF THE RTOM INTERRUPT LEVEL SUBADDRESS.

NOTES:
1. INTERRUPT TABLE ENTRIES ARE IN SCRATCHPAD ADDRESSES 80-81, AND 92-FF. OF THESE ADDRESSES, ADDRESSES 94-A3 ARE IOM ENTRIES, AND THE REMAINING ADDRESSES ARE RTOM ENTRIES.
2. SCRATCHPAD INTERRUPT TABLE ADDRESSES ARE OBTAINED BY ADDING $80_H$ TO THE INTERRUPT LEVEL.

Format - Scratchpad Interrupt Table Entry

# EXAMPLE INITIAL CONFIGURATION DECK TYPICAL OF 55

| EXAMPLE | COMMENTS |
|---|---|
| (SEE NOTE) | READ ASCII CARD READER IOCD |
| *DEV04=0E150400,02 | CARTRIDGE DISC WITH TWO PLATTERS |
| *DEV08=0E160800,04 | MOVING-HEAD DISC |
| *DEV10=0E181000,04 | 9-TRACK MAG TAPE |
| *DEV20=0E1A2000,10 | GPMC |
| *DEV60=0E1E6000,08 | ADS |
| *DEV78=01207800 | PRIMARY CARD READER |
| *DEV7A=00217802 | PRIMARY LINE PRINTER |
| *DEV7E=02237801 | PRIMARY TELETYPE |
| *INT00=1F | POWER FAIL/AUTO RESTART |
| *INT01=1E | SYSTEM OVERRIDE |
| *INT12=1D | MEMORY PARITY TRAP |
| *INT13=1C | CONSOLE INTERRUPT |
| *INT24=1B | NONPRESENT MEMORY |
| *INT25=1A | UNDEFINED INSTRUCTION TRAP |
| *INT26=19 | PRIVILEGE VIOLATION |
| *INT27=18 | CALL MONITOR |
| *INT28=16 | REAL-TIME CLOCK |
| *INT29=17 | ARITHMETIC EXCEPTION |
| *INT2A=15 | EXTERNAL INTERRUPT |
| *INT2B=14 | EXTERNAL INTERRUPT |
| *INT2C=13 | EXTERNAL INTERRUPT |
| *INT2D=12 | EXTERNAL INTERRUPT |
| *END | LAST CARD |

NOTE: THE FIRST RECORD IS DEVICE DEPENDENT AND REPRESENTS TWO 32-BIT WORDS. THE FIRST BEING ALL ZEROS AND THE SECOND A VALID IOCD TO READ THE FOLLOWING RECORDS.

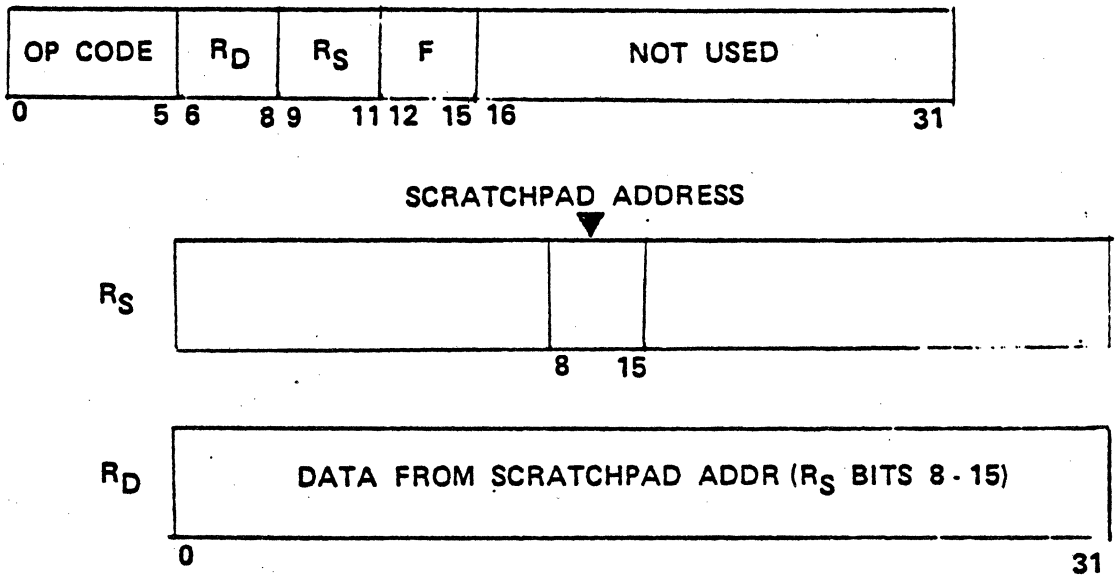
Figure 1 : System Initial Configuration Load (ICL) Deck

ICL

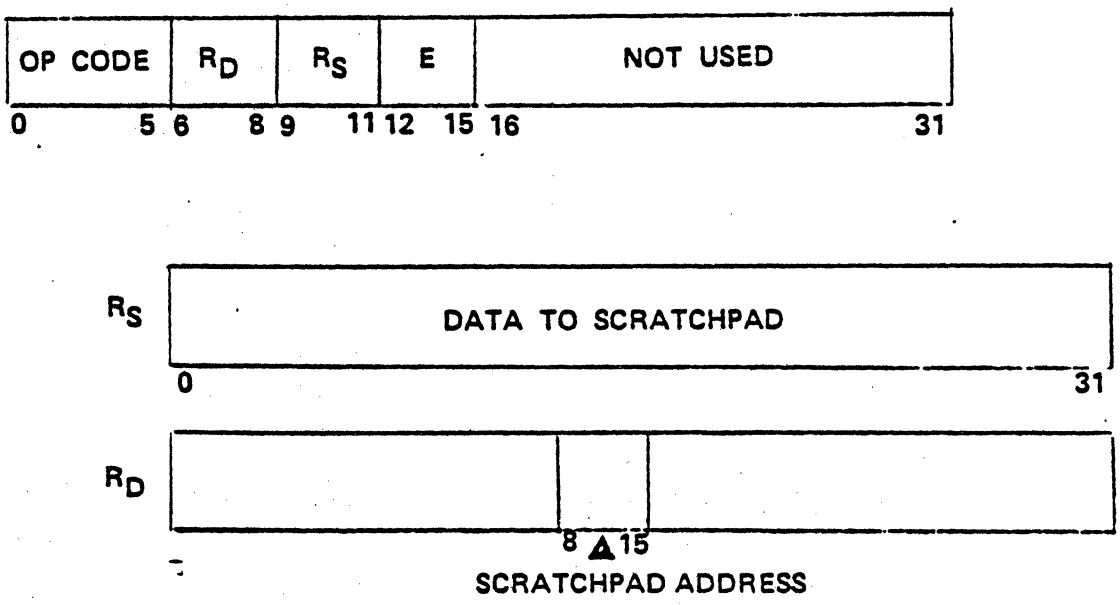
| | Interrupt Level | Sub Address ICL |
|---|---|---|
| RTOM #1 (2117) WITH INTERVAL TIMER | 00 | 1(9) F |
| | 01 | E |
| | 12 | D |
| | 13 | C |
| | 24 | B |
| | 25 | A |
| | 26 | 9 |
| | 27 | 8 |
| | 28 | 6 |
| | 29 | 7 |
| | 2A | 5 |
| | 2B | 4 |
| | 2C | 3 |
| | 2D | 2 |
| | 2E | 1 |
| | 2F | 0 |
| RTOM #2 (2118) WITHOUT INTERVAL TIMER | 30 | 2(A) F |
| | 31 | E |
| | 32 | D |
| | 33 | C |
| | 34 | B |
| | 35 | A |
| | 36 | 9 |
| | 37 | 8 |
| | 38 | 7 |
| | 39 | 6 |
| | 3A | 5 |
| | 3B | 4 |
| | 3C | 3 |
| | 3D | 2 |
| | 3E | 1 |
| | 3F | 0 |
| RTOM 3, 4, 5, etc. | | |

Interrupt, Sub Address (ICL Card), Logic Relationship

## TRANSFER SCRATCHPAD TO REGISTER INSTRUCTION FORMAT




## TRANSFER REGISTER TO SCRATCHPAD INSTRUCTION FORMAT



PSD [redacted] 77

| TRAP LEVEL | IVL (TVL) | DESCRIPTION |
|---|---|---|
| 00 | 0F4 | POWER FAIL SAFE TRAP |
| 01 | 0FC | SYSTEM OVERRIDE TRAP (NOT USED) |
| 02 | 0E8* | MEMORY PARITY TRAP |
| 03 | 190* | NONPRESENT MEMORY TRAP |
| 04 | 194* | UNDEFINED INSTRUCTION TRAP |
| 05 | 198* | PRIVILEGE VIOLATION TRAP |
| 06 | 180 | SUPERVISOR CALL TRAP |
| 07 | 184 | MACHINE CHECK TRAP |
| 08 | 188 | SYSTEM CHECK TRAP |
| 09 | 18C | MAP FAULT TRAP |
| 0A | | NOT USED |
| 0B | | NOT USED |
| 0C | | NOT USED |
| 0D | | NOT USED |
| 0E | 0E4 | BLOCK MODE TIMEOUT TRAP |
| 0F | 1A4* | ARITHMETIC EXCEPTION TRAP |

All or equal level

Don't have to enable each trap

* VECTOR LOCATIONS SHARED WITH RTOM INTERRUPTS

PSD [redacted]
REQUIRED ON 1ST RTOM

| INTERRUPT LEVEL | IVL | DESCRIPTION |
|---|---|---|
| 00 | 0F0 | POWER FAIL SAFE INTERRUPT |
| 13 | 0EC | ATTENTION INTERRUPT |
| 27 | 19C | CALL MONITOR INTERRUPT |
| 28 | 1A0 | REAL TIME CLOCK INTERRUPT |
| 7F | 2FC | INTERVAL TIMER INTERRUPT |

## PSW TRAP IMPLEMENTATION ON RTOM

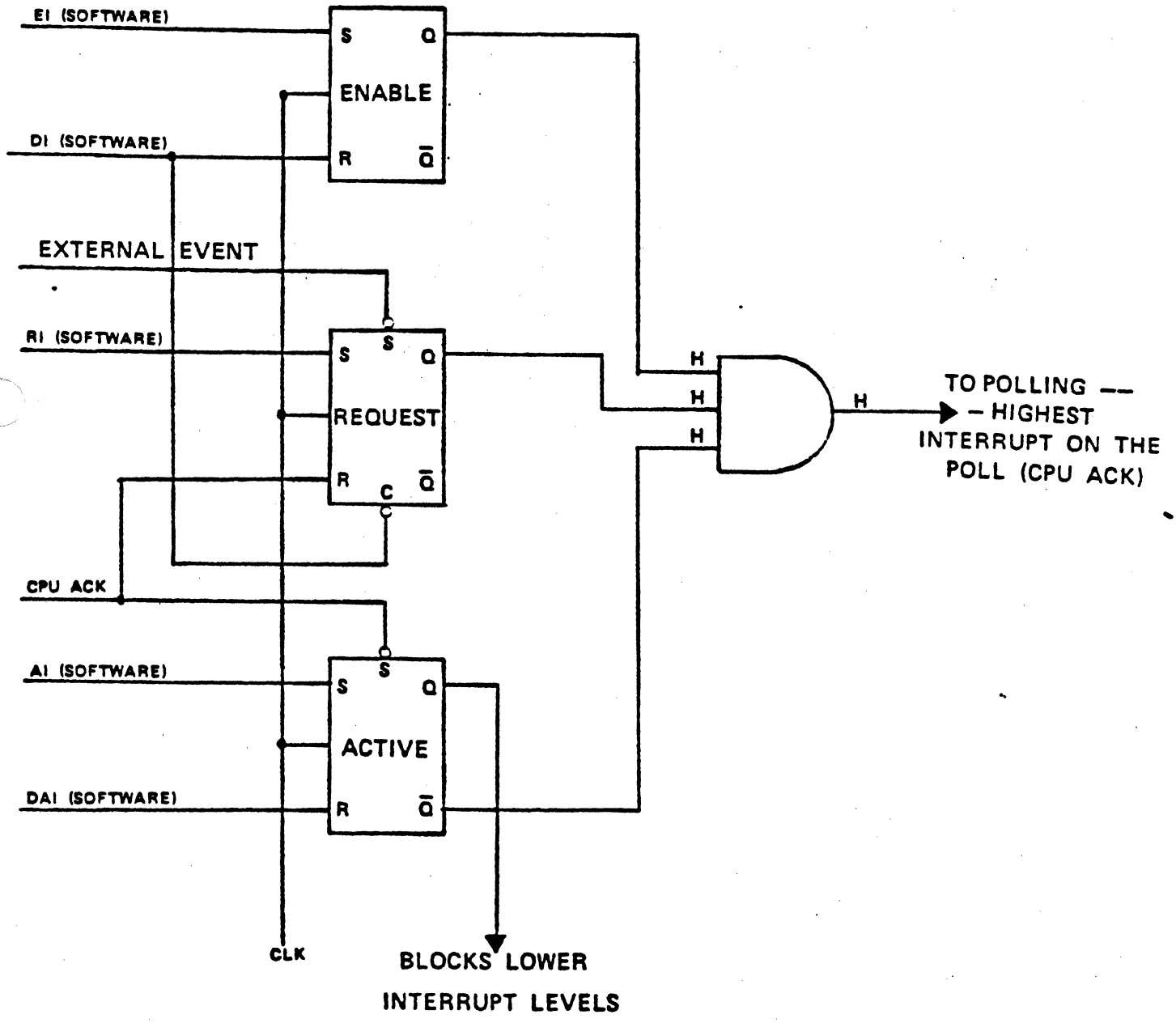
| INTERRUPT LEVEL | IVL | DESCRIPTION |
|---|---|---|
| 00 | 0F4 | POWER FAIL SAFE TRAP |
| 01 | 0FC | SYSTEM OVERRIDE TRAP |
| 12 | 0E8* | MEMORY PARITY TRAP |
| 24 | 190* | NONPRESENT MEMORY TRAP |
| 25 | 194* | UNDEFINED INSTRUCTION TRAP |
| 26 | 198* | PRIVILEGE VIOLATION TRAP |
| 29 | 1A4* | ARITHMETIC EXCEPTION TRAP |

## PSW INTERRUPTS REQUIRED ON 1ST RTOM

| INTERRUPT LEVEL | IVL | DESCRIPTION |
|---|---|---|
| 00 | 0F0 | POWER FAIL SAFE INTERRUPT |
| 01 | 0F8 | SYSTEM OVERRIDE INTERRUPT |
| 12 | 0E8* | MEMORY PARITY INTERRUPT |
| 13 | 0EC | ATTENTION INTERRUPT |
| 24 | 190* | NONPRESENT MEMORY INTERRUPT |
| 25 | 194* | UNDEFINED INSTRUCTION INTERRUPT |
| 26 | 198* | PRIVILEGE VIOLATION INTERRUPT |
| 27 | 19C | CALL MONITOR INTERRUPT |
| 28 | 1A0 | REAL TIME CLOCK INTERRUPT |
| 29 | 1A4* | ARITHMETIC EXCEPTION INTERRUPT |
| 2A | 1A8 | EXT INT (J.WAIT FOR RTM) |
| 2B | 1AC | INTERVAL TIMER (RTM) |
| 2C | 1B0 | EXT INT (TSS FOR RTM) |
| 2D | 1B4 | EXT INT (RESERVED FOR RTM) |
| 2E | 1B8 | EXT INT (RESERVED FOR RTM) |
| 2F | 1BC | EXT INT (RESERVED FOR RTM) |

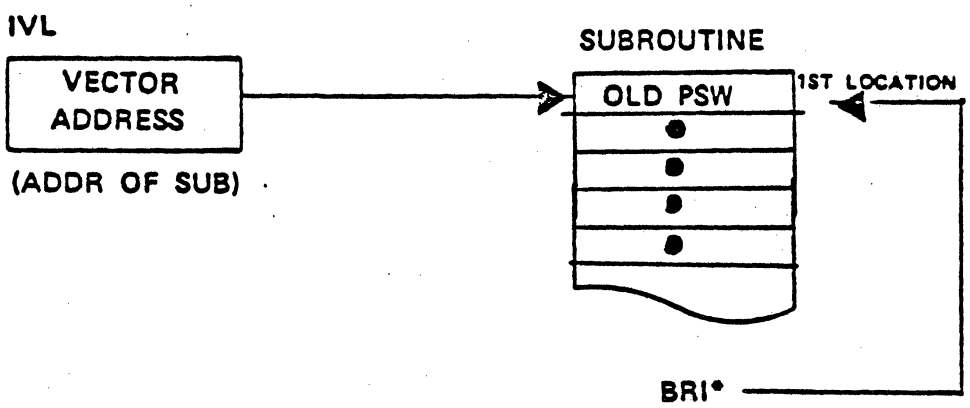
* VECTOR LOCATIONS SHARED WITH PSD TRAPS

# INTERRUPT STATES

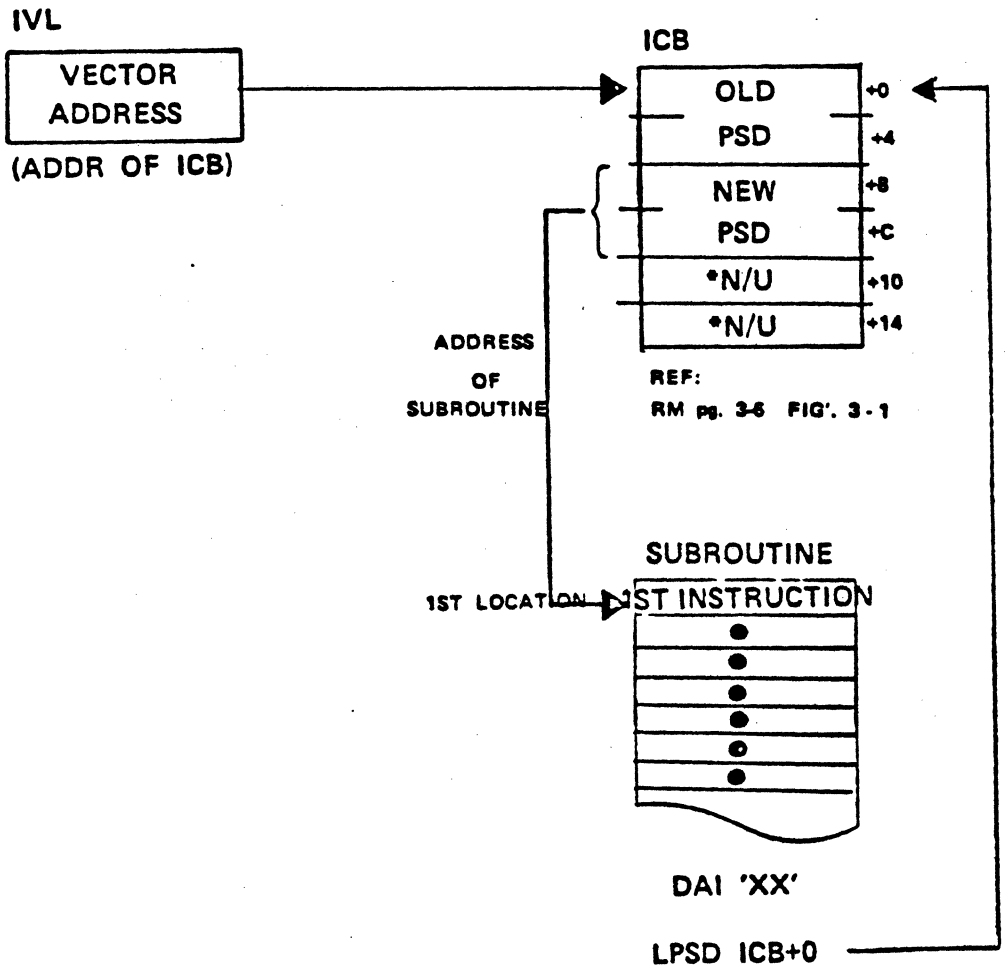


# PSW MODE

# INTERRUPT PROCESSING

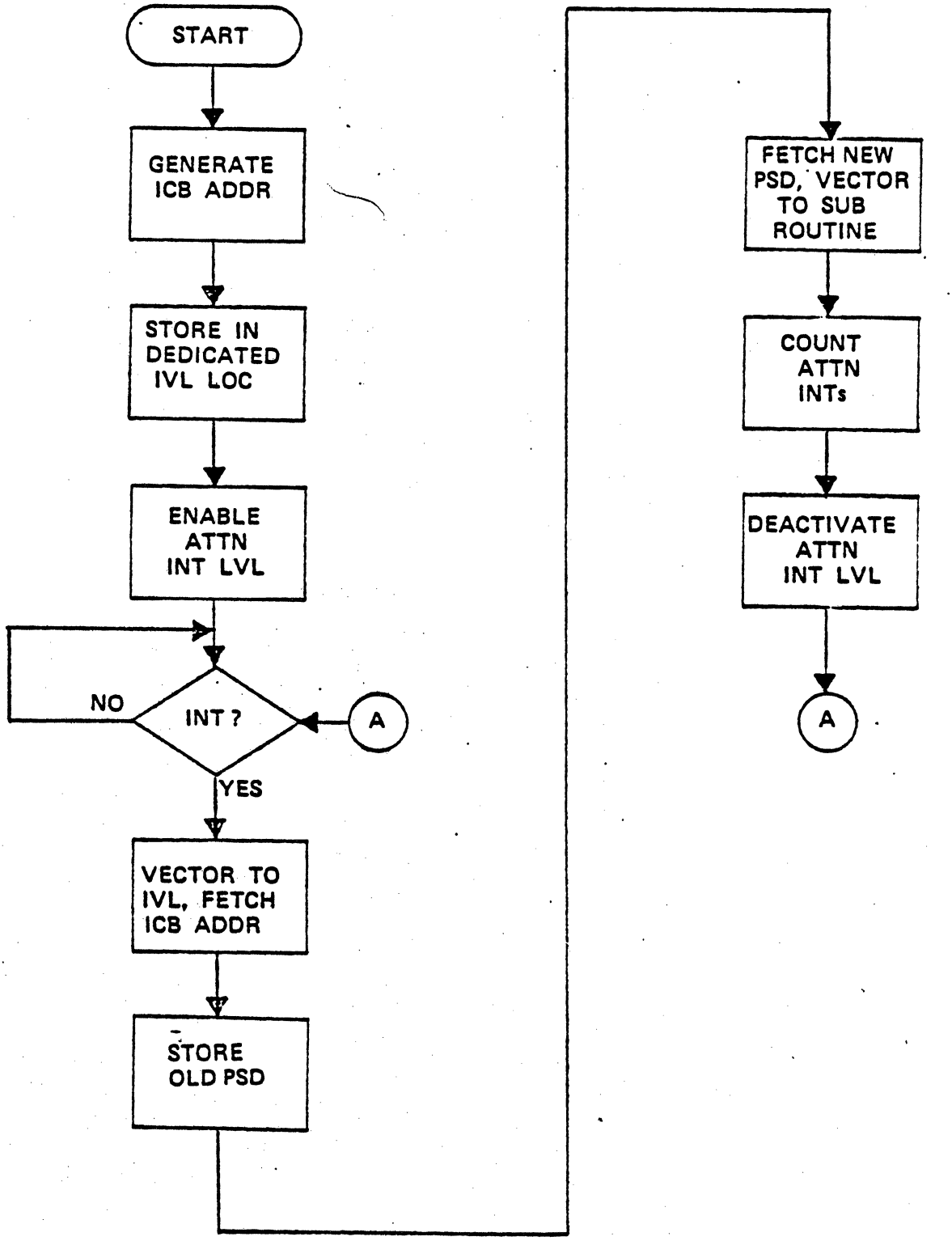


# PSD MODE
# INTERRUPT PROCESSING



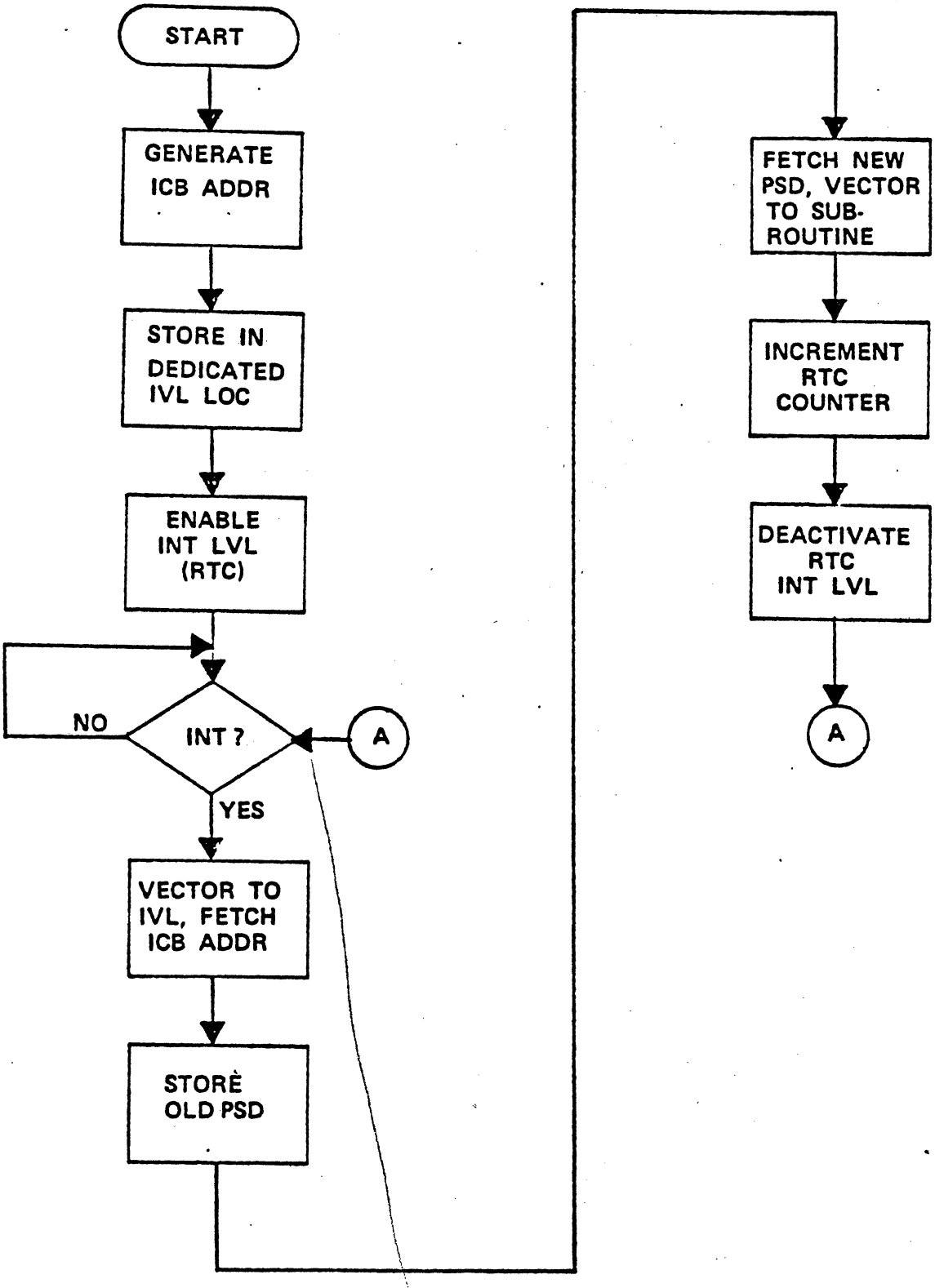
# INTERRUPT PROGRAM FOR ATTENTION BUTTON

RTOM



# INTERRUPT PROGRAM FOR REAL TIME CLOCK CHECKOUT

RTOVL



# System 32/75 Instruction Set Worksession

Introduction:

The purpose of this worksession is to familiarize the student with the instructions used by the 32/75.

The student will be given the information to enter into memory and registers in order to execute the instruction. The student will write down the results observed after execution.

Reference:

SYSTEMS 32/75 Reference Manual

1. Execute a "Supervisor Call" (SVC) instruction as follows:

| Location | Data | Comments |
|---|---|---|
| 00020 | C8060ACE | SVC Instruction |
| 00024 | 00000030 | Secondary Vector |
| 00030 | 00000000 | OLD PSD1 |
| 00034 | 00000000 | OLD PSD2 |
| 00038 | 80800050 | NEW PSD1 |
| 0003C | 00000000 | NEW PSD2 |
| 00040 | 00000000 | Call Field |
| 00050 | EC000050 | SVC Routine |
| 00180 | 00000024 | Primary Vector |

Set P.C. to 00020

Press Instruction Step

PSD1 =

Loc. 00040 =

Why does PSD1 and Loc. 40 appear as they do?

1. Execute RDSTS 0009

2. Execute a "Set CPU" (SETCPU) instruction as follows:

| Location | Data | Comments |
|---|---|---|
| 00060 | 2C090002 | SETCPU Instruction |
| GPR 0 | 00031000 | Mode bits in Reg. 0 |

PSD1 =

3. Execute a "Read CPU Status Word" (RDSTS) instruction as follows:

| Location | Data | Comments |
|---|---|---|
| 00070 | 00090002 | RDSTS Instruction |

Set P.C. to 00070

Press Instruction Step

GPR 0 =

Why does the CPU status word appear as it does?

4. Execute an "Enable Arithmetic Exception Trap" (EAE) instruction.

| Location | Data | Comments |
|---|---|---|
| 00074 | 00080002 | EAE Instruction |

Set P.C. to 00074

Press Instruction Step

PSD1 =

What is the meaning of bit 7 in PSD1?

5. Execute a "Block External Interrupts" (BEI) instruction.

| Location | Data | Comments |
|---|---|---|
| 00078 | 00060000 | BEI Instruction |

Set P.C. to 00078

Press Instruction Step

PSD1 =

Note: Re-execute the RDSTS instruction to look at the CPU status word.

Set P.C. to 00070

Press Instruction Step

GPR 0 =

What is the meaning of bit 26?

DAY 5

INSTRUCTION SET

SCRATCHPAD & INT/TRAP - WORKSESSION

REFERENCES:

32/70 SERIES
REFERENCE MANUAL
SECTIONS III & VI

WORKBOOK

Objective: State the purpose and function of Scratchpad.

Interpret ICL Device Entries and Interrupt Entries in Scratchpad.

Define the TRAP and Interrupt structure of a 32/70 Series computer.

1. Where is scratchpad located? ON CPU BOARDS A AND B LOCATIONS 300-6E0

2. What is the purpose of scratchpad? HAS INFO FOR CPU FIRMWARE GIVING (DEVICE, DEVICE INT PRIORITY INTERRUPT ENTRY) LINK PHYS. TO LOGICAL

3. Given the following ICL card *DEV10=0E18(001) define:

   (A) The device address for CD/TD instructions 10

   (B) The class of IOM E=IOM

   (C) The interrupt level (SI) 18

   (D) The IOM physical address 10

   (E) The device subaddress 01

   (F) At what location in scratchpad will this device entry be located? 10

4. For the above question, where on scratchpad will the interrupt entry be located? IL+80=98

5. What is contained in the RHW of an interrupt entry in scratchpad? The IVL

6. Given the following ICL card *INT26=19 define:

   (A) What is the interrupt level? 26

   (B) The RTOM physical address? 79

   (C) What subaddress on the RTOM is the interrupt connected to? 9

location scratchpad 80+2h = A1

7. In a device interrupt entry in scratchpad, what do bits 8 thru 15 reference? ~~8=RTOM Entry~~

TVL

~~9-11 = 3 LSB's of RTOM A/B ADD~~ 12-15 = 4 MSB's of Sub Address RTOM

8. In a transfer scratchpad to register instruction - in what register is the scratchpad address located? Which bits? Rs bits 8-15 (9-11)

BYTE 1

9. List the indications observed on the System Control Panel when a non-present Memory Trap ocurrs, without traps enabled.

190 Interrupt Active

HALT

PSW → TVL

10. When a trap halt ocurrs, where could further information be obtained to aid in isolating the problem?

530, 540

11. How are traps enabled in the PSD mode? SET CPU

12. How are traps handled when enabled in the PSD mode?

Software subroutines TVL → TCB → SUBROUTINE

13. How are traps enabled in the PSW mode? Enable Interrupt

14. In the PSD mode, what interrupts are required on the 1st RTOM? 00, 13, 27, 28, 7F Power Fail Safe, Attention, Call Monitor, RTC, Interval Timer. - ↓ RTM OS.

15. How is the "Arithmetic Exception Trap" enabled and what bit in the CPU status word is set when the trap is enabled? Enable Arithmetic Exception Bit 24

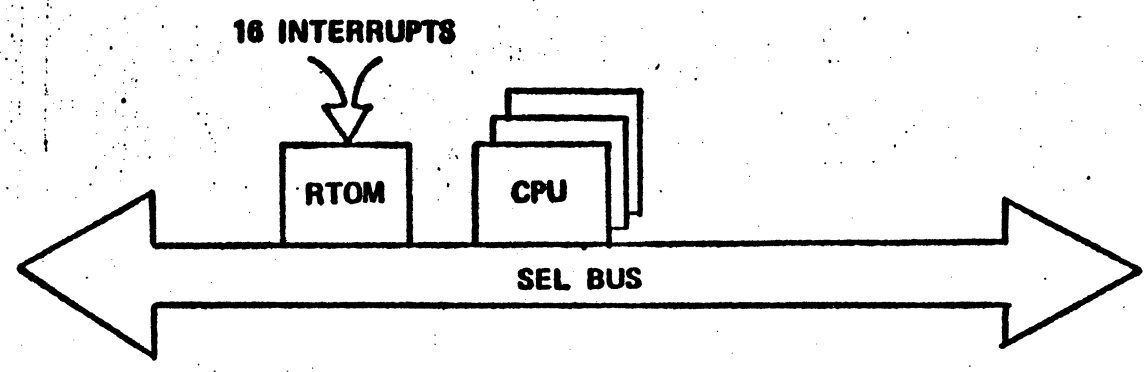
Bit 7 of PSD would be set

# DAY 6

# SECTION 6

# RTOM

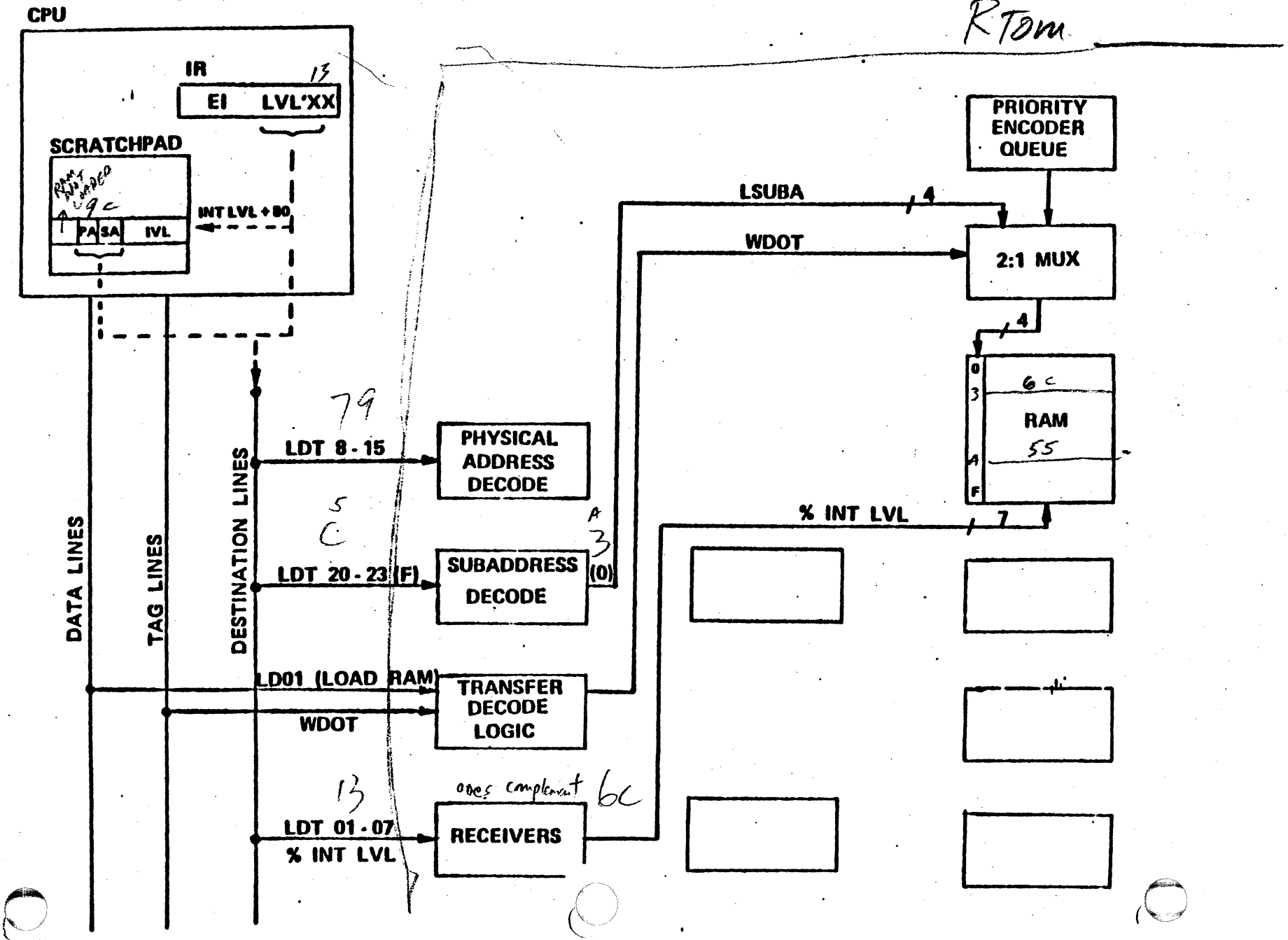
# REAL TIME OPTION MODULE



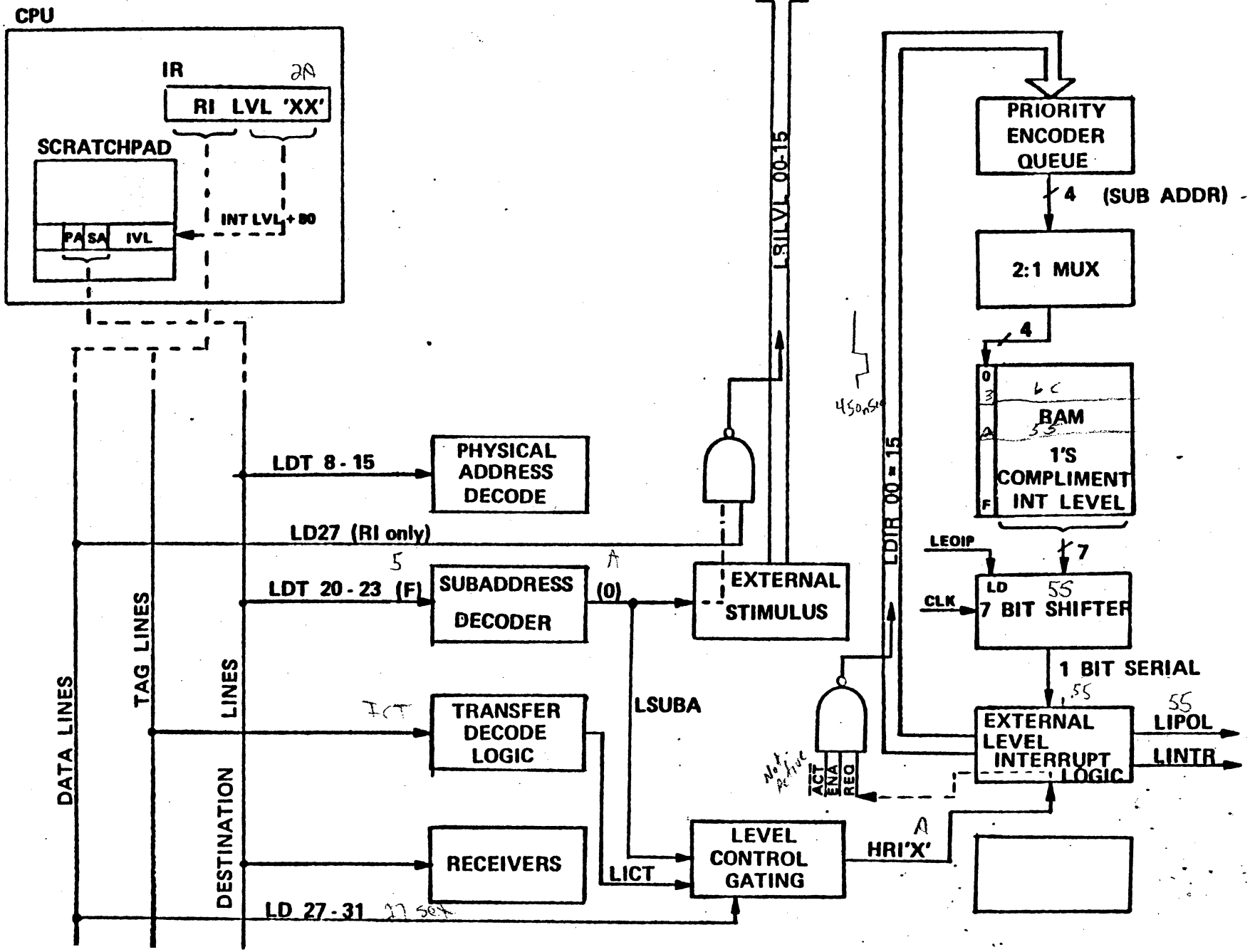

RTOM MODEL 2117 WIRE/WRAP
2118 WIRE/WRAP
2345 COPPER

- SYSTEM REQUIRES AT LEAST ONE AND MAY HAVE 7 MAX
- THE SEL BUS INTERRUPT CONTROL LINES PROVIDE FOR 112 INTERRUPT LEVELS
- [redacted]
- [redacted]
- THE INTERRUPT STRUCTURE IS FULLY PROGRAMMABLE BY THE USER TO MANIPULATE THE PRIORITY OF ANY INTERRUPT EVENT.

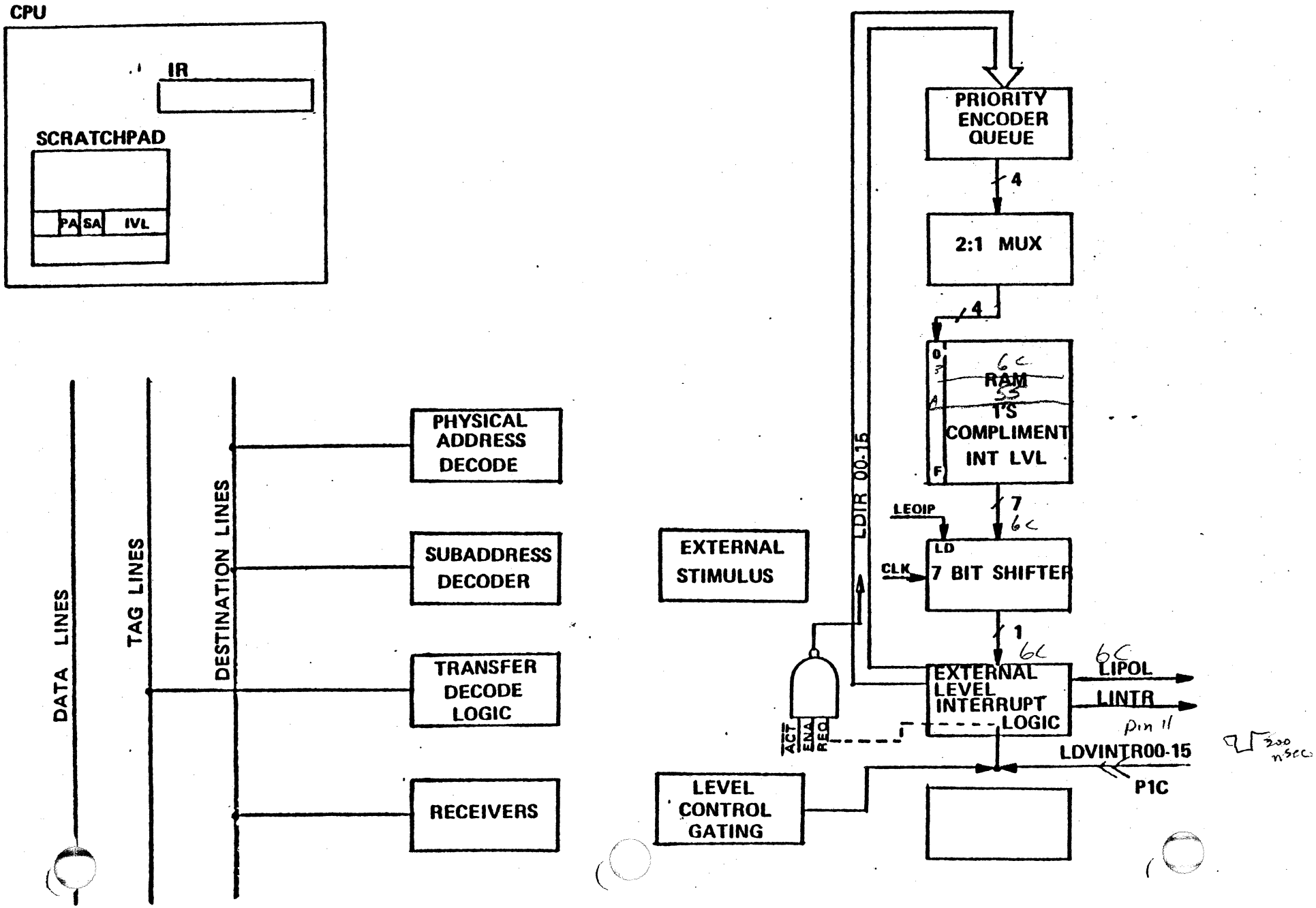
# RTOM 'LOAD RAM' SEQUENCE

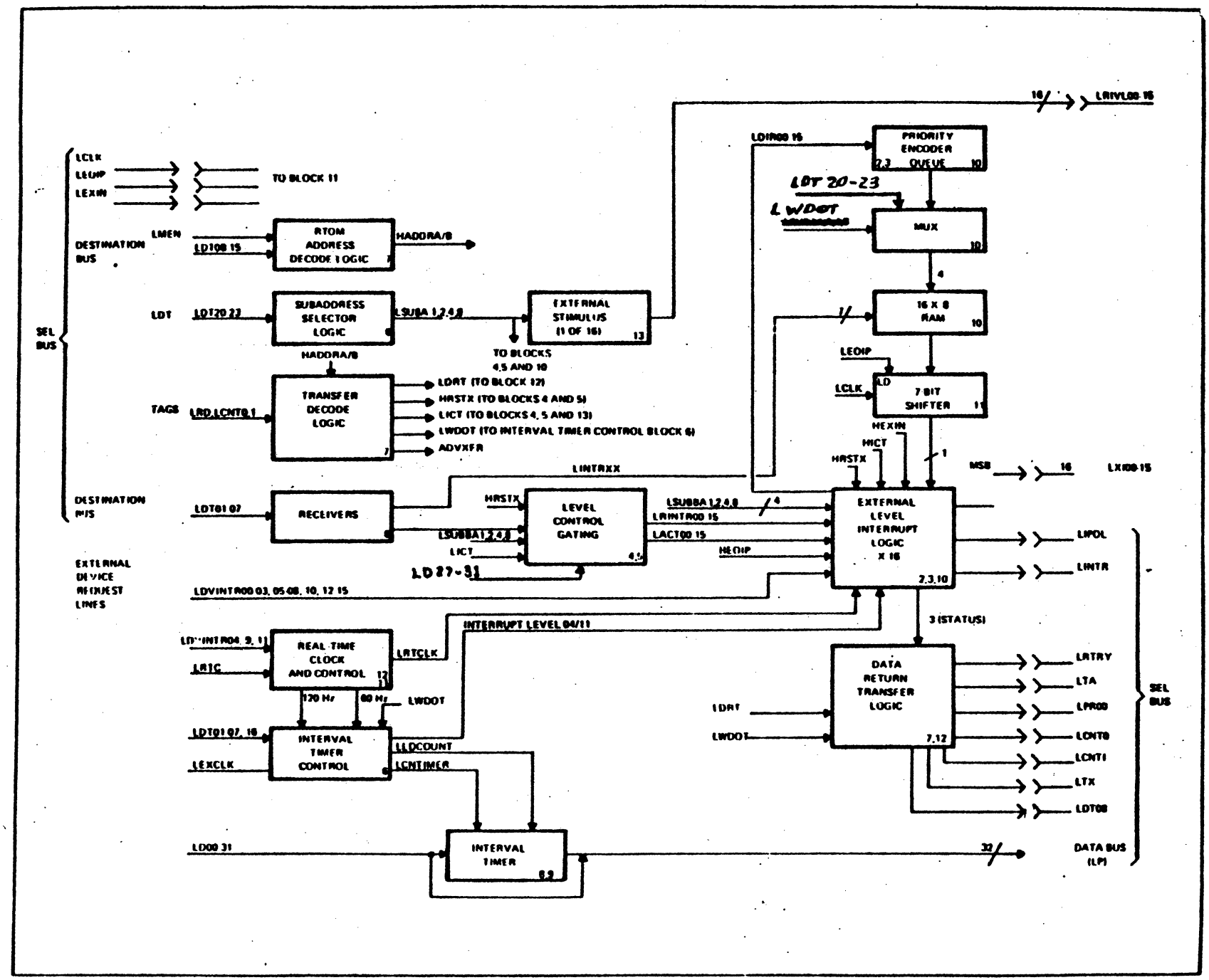


# RTOM SOFTWARE 'REQUEST INTERRUPT' FLOW



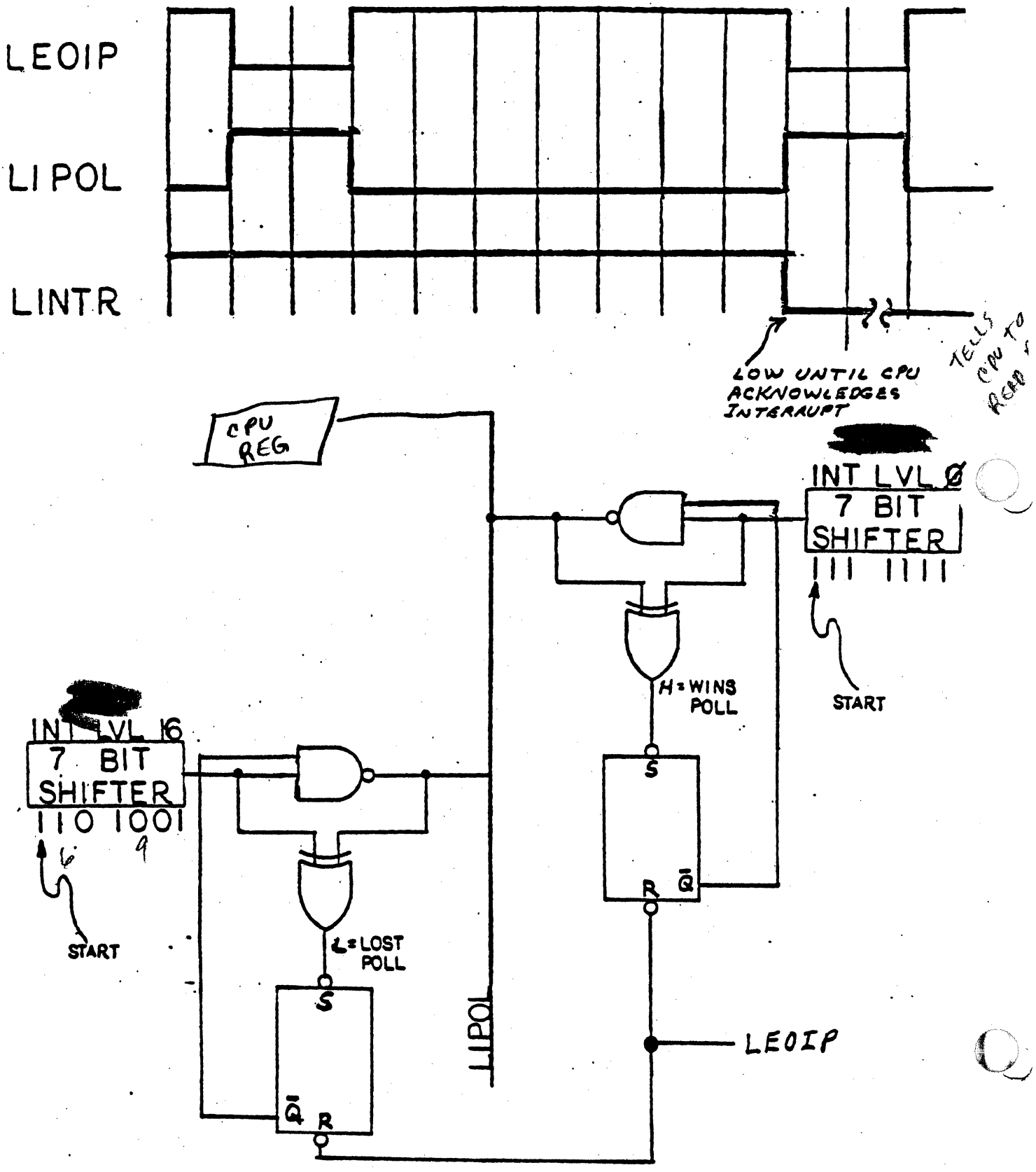
# RTOM 'EXTERNAL INTERRUPT' FLOW SEQUENCE






Block Diagram - RTOM

# SEL 32 INTERRUPT STRUCTURE



## OPERATION OF LIPOL

When LEOIP goes FALSE (HIGH) to enable shifting, the EXCLUSIVE OR'S have high outputs. A zero (1'S compl.) coming out of the shifters is a HIGH, qualifying the NAND gate, driving LIPOL LOW. If, in any bit period a shifter shifts out a "LOW", the NAND is defeated and attempts to drive LIPOL high. However, if a higher priority level is holding LIPOL LOW at this time, the EXCLUSIVE OR on the board that is shifting out a LOW will output a LOW and set its disabling flip-flop, which will inhibit this board from further contention for LIPOL.

The one level which is able to shift out all seven bits will generate LINTR, notifying the CPU that there is a seven bit code to be examined. After CPU firmware knows the identity of the winning interrupt, it acknowledges the interrupt with an RSTX SELBUS Transfer, which causes the interrupt to go active and terminates LINTR, allowing another cycle.

While this level is active, it continues its contention for LIPOL, winning until a higher priority is contending also.

# ICL INTERRUPT ENTRY FORMAT

**Format #2** *INTXX=RS

where:

| | |
|---|---|
| *INT | defines that the record contains an interrupt definition entry. |
| XX | is the hexadecimal interrupt priority level that is to be emulated. |
| = | is a necessary delimiter. Each letter to the right of this delimiter represents one hexadecimal digit (four binary bits). |
| R | is the hexadecimal RTOM board number to which the interrupt XX is assigned. |
| S | is the hexadecimal subaddress on the RTOM board to which the interrupt XX is assigned (in one's complement). |
| NOTE 1: | RTOM physical controller address $79_{16}$ is RTOM board number 1, address $7A_{16}$ is RTOM board number 2, etc. |
| NOTE 2: | Real-Time Clock hardware is connected to subaddress $6_{16}$ on the RTOM board. |
| NOTE 3: | Interval Timer hardware is connected to subaddress $4_{16}$ on the RTOM board. |
| NOTE 4: | RTOM physical controller addresses must be $79_{16}$ or above. This convention allows a maximum of seven RTOM boards to be defined on a single 32 system. Seven RTOM boards will support $112_{16}$ interrupt levels. |

**Format #3** *END

where:

| | |
|---|---|
| *END | is the last record of an Initial Configuration Load (ICL) deck. This record signifies the end of the load process. |

**EXAMPLES OF INITIAL CONFIGURATION LOAD (ICL) RECORDS**

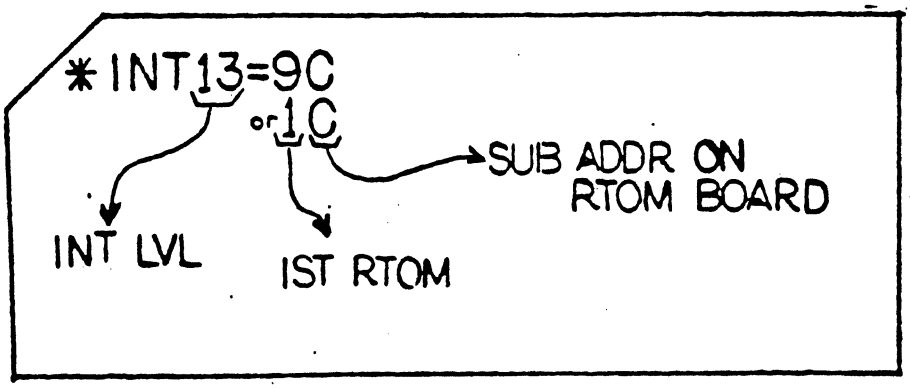
A device entry:

*DEV04=0E140100,04

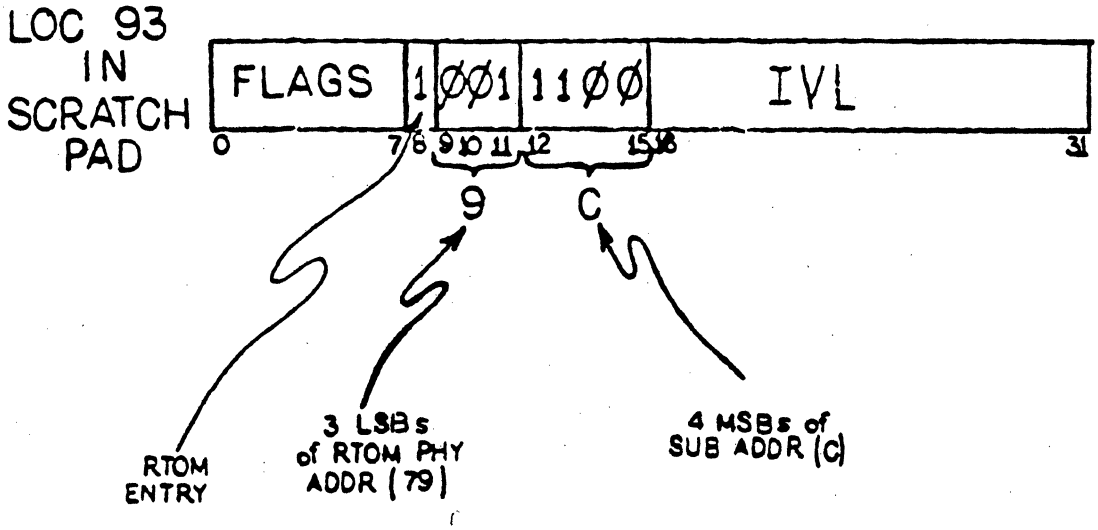
The device entry above specifies the following information:

1. The 32/55 input/output commands will address the controller as $4_{16}$.
2. The ",04" is an optional parameter that specifies that there are $4_{16}$ devices on the controller. There will be four entries defined in the Configuration RAM (CR). The 32/55 input/output commands will address the devices as $4_{16}$, $5_{16}$, $6_{16}$, and $7_{16}$.

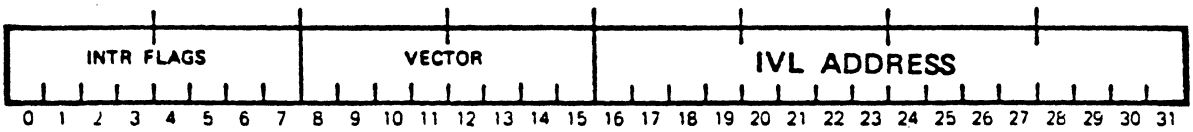
# INTERRUPT ENTRY CARD




## SCRATHPAD INTERRUPT ENTRY (RTOM)



# SCRATCHPAD INTERRUPT TABLE ENTRY




| INTERRUPT FLAGS | |
|---|---|
| BIT | DEFINITION |
| 0 | RAM LOADED |
| 1 | I/O IN PROGRESS |
| 2 | 2 IOCD'S REQUIRED |
| 3 | ARITHMETIC EXCEPTION PENDING |
| 4 | REFETCH WITH IGNORE ADDRESS STOP |
| 5 | ACTIVATE/DEACTIVATE INTERRUPT ISSUED<br>0 = DEACTIVATE ISSUED<br>1 = ACTIVATE ISSUED |
| 6 | REQUEST INTERRUPT ISSUED |
| 7 | ENABLE/DISABLE INTERRUPT ISSUED<br>0 = DISABLE ISSUED<br>1 = ENABLE ISSUED |

**VECTOR**

BIT 08 = 0; BITS 09-15 PROVIDE DEVICE VECTOR (IOM ENTRY)
BIT 08 = 1; BITS 09-15 PROVIDE THE RTOM ENTRY.

**IOM ENTRY:**

IF BIT 08 = 0; BITS 09-15 PROVIDE THE SCRATCHPAD ADDRESS OF THE DEVICE ENTRY (REFER TO FIGURE 5-51 FOR FORMAT) AND ALSO CORRESPOND TO THE COMMAND DEVICE OR TEST DEVICE INSTRUCTION DEVICE ADDRESS FIELD.

**RTOM ENTRY:**

IF BIT 08 = 1; BITS 09 11 PROVIDE THE THREE LEAST SIGNIFICANT BITS OF THE RTOM PHYSICAL ADDRESS. (FOUR MOST SIGNIFICANT BITS OF PHYSICAL ADDRESS ARE (1111

BITS 12-15 PROVIDE THE FOUR BITS OF THE RTOM INTERRUPT LEVEL SUBADDRESS.

NOTES:
1. INTERRUPT TABLE ENTRIES ARE IN SCRATCHPAD ADDRESSES 80-81, AND 92-FF. OF THESE ADDRESSES, ADDRESSES 94-A3 ARE IOM ENTRIES, AND THE REMAINING ADDRESSES ARE RTOM ENTRIES.
2. SCRATCHPAD INTERRUPT TABLE ADDRESSES ARE OBTAINED BY ADDING $80_H$ TO THE INTERRUPT LEVEL.

Format - Scratchpad Interrupt Table Entry

Table 1-2. P1A Connector Assignments

| Pin | Signal |
|---|---|
| 1 | +5V |
| 2 | +5V |
| 3 | LRILVL00 |
| 4 | |
| 5 | LRILVL01 |
| 6 | |
| 7 | LRILVL02 |
| 8 | |
| 9 | GND |
| 10 | |
| 11 | LRILVL03 |
| 12 | |
| 13 | |
| 14 | GND |
| 15 | LRILVL04 |
| 16 | |
| 17 | LRILVL05 |
| 18 | |
| 19 | LRILVL06 |
| 20 | |
| 21 | GND |
| 22 | |
| 23 | LRILVL07 |
| 24 | |
| 25 | |
| 26 | GND |
| 27 | LRILVL08 |
| 28 | |
| 29 | LRILVL09 |
| 30 | |
| 31 | LREALTCLK |
| 32 | GND |
| 33 | LRILVL10 |
| 34 | |
| 35 | LRILVL11 |
| 36 | |
| 37 | LINTVO |
| 38 | GND |
| 39 | LRILVL12 |
| 40 | |
| 41 | LRILVL13 |
| 42 | |
| 43 | |
| 44 | GND |
| 45 | LRILVL14 |
| 46 | |
| 47 | LRILVL15 |
| 48 | |
| 49 | |
| 50 | GND |

Table 1-3. P1C Connector Assignments

| Pin | Signal |
|---|---|
| 1 | |
| 2 | |
| 3 | LDVINTR00 |
| 4 | LXI01 |
| 5 | LDVINTR01 |
| 6 | LXI00 |
| 7 | LDVINTR02 |
| 8 | LXI02 |
| 9 | |
| 10 | LXI03 |
| 11 | LDVINTR03 |
| 12 | LXI04 |
| 13 | LXI05 |
| 14 | |
| 15 | LDVINTR04 |
| 16 | LXI06 |
| 17 | LDVINTR05 |
| 18 | LXI07 |
| 19 | LDVINTR06 |
| 20 | LXI08 |
| 21 | |
| 22 | LXI09 |
| 23 | LDVINTR07 |
| 24 | LXI10 |
| 25 | LXI11 |
| 26 | |
| 27 | LDVINTR08 |
| 28 | LXI12 |
| 29 | LDVINTR09 |
| 30 | LXI13 |
| 31 | LXI14 |
| 32 | |
| 33 | LDVINTR10 |
| 34 | LXI15 |
| 35 | LDVINTR11 |
| 36 | |
| 37 | |
| 38 | |
| 39 | LDVINTR12 |
| 40 | |
| 41 | LDVINTR13 |
| 42 | |
| 43 | LEXCLK |
| 44 | |
| 45 | LDVINTR14 |
| 46 | |
| 47 | LDVINTR15 |
| 48 | |

| Interrupt (Relative Physical Priority and Subaddress On RTOM Board) | RTOM Subaddress On ICL Deck | Interrupt Level (16) | Interrupt Definition |
|---|---|---|---|
| 0 | 15(F) | Level 0 | Power Fail Safe/Auto Start Interrupt |
| 1 | 14(E) | Level 1 | System Override Interrupt |
| 2 | 13(D) | Level 12 | Memory Parity |
| 3 | 12(C) | Level 13 | Attention Interrupt (Console Interrupt |
| 4 | 11(B) | Level 24 | Nonpresent Memory |
| 5 | 10(A) | Level 25 | Undefined Instruction |
| 6 | 9 | Level 26 | Privilege Violation |
| 7 | 8 | Level 27 | Call Monitor |
| 8 | 7 | Level 29 | Arithmetic Exception |
| 9 | 6 | Level 28 | Real-Time Clock |
| 10 | 5 | Level 2A | External Interrupt (J.Wait if RTM 6.1) |
| 11 | 4 | Level 2B | Interval Timer |
| 12 | 3 | Level 2C | External Interrupt (T.S.S.) |
| 13 | 2 | Level 2D | External Interrupt } Reserved for RTM Software |
| 14 | 1 | Level 2E | External Interrupt |
| 15 | 0 | Level 2F | External Interrupt |

First RTOM Internal Priority Interrupt Levels
RTM/PSW Mode

| Interrupt Relative Physical Priority and Subaddress On RTOM Board) | RTOM Subaddress On ICL Deck | Interrupt Level (16) | Interrupt Definition |
|---|---|---|---|
| 0 | 15(F) | | |
| 1 | 14(E) | | |
| 2 | 13(D) | | |
| 3 | 12(C) | Level 13 | Attention Interrupt (Console Interrupt |
| 4 | 11(B) | | |
| 5 | 10(A) | | |
| 6 | 9 | | |
| 7 | 8 | Level 27 | Call Monitor |
| 8 | 7 | | |
| 9 | 6 | Level 28 | Real-Time Clock |
| 10 | 5 | | |
| 11 | 4 | Level 7F | Interval Timer |
| 12 | 3 | | |
| 13 | 2 | | |
| 14 | 1 | | |
| 15 | 0 | | |

1st RTOM assignments for PSD Mode or MPX.

RTOM #1-7 79-7F

ICL

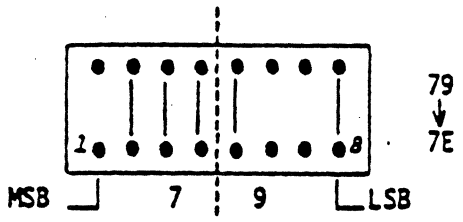
| | Interrupt Level | Sub Address ICL | Logic Mnemonic |
|---|---|---|---|
| | 00 | 1(9) F | 00 |
| | 01 | E | 01 |
| | 12 | D | 02 |
| | 13 | C | 03 |
| | 24 | B | 04 |
| | 25 | A | 05 |
| RTOM #1 (2117) WITH INTERVAL TIMER | 26 | 9 | 06 |
| | 27 | 8 | 07 |
| | 28 | 6 | 09 |
| | 29 | 7 | 08 |
| | 2A | 5 | 10 |
| | 2B | 4 | 11 |
| | 2C | 3 | 12 |
| | 2D | 2 | 13 |
| | 2E | 1 | 14 |
| | 2F | 0 | 15 |
| | 30 | 2(A) F | 00 |
| | 31 | E | 01 |
| | 32 | D | 02 |
| | 33 | C | 03 |
| | 34 | B | 04 |
| | 35 | A | 05 |
| RTOM #2 (2118) WITH INTERVAL TIMER | 36 | 9 | 06 |
| | 37 | 8 | 07 |
| | 38 | 7 | 08 |
| | 39 | 6 | 09 |
| | 3A | 5 | 10 |
| | 3B | 4 | 11 |
| | 3C | 3 | 12 |
| | 3D | 2 | 13 |
| | 3E | 1 | 14 |
| | 3F | 0 | 15 |
| RTOM 3, 4, 5, etc. | 40 | 3(B) F | 00 |
| | 41 | E | 01 |
| | ↓ | ↓ | ↓ |
| | 50 | 4(C) F | 00 |
| | 51* | E | 01 |
| | ↓ | | |

Page 6-11

## RTOM

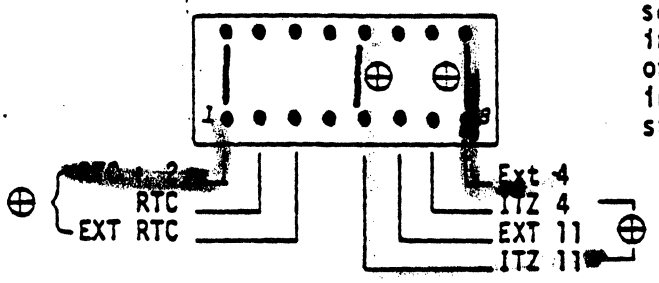
2345/2117 Phy Addr

U13/D10



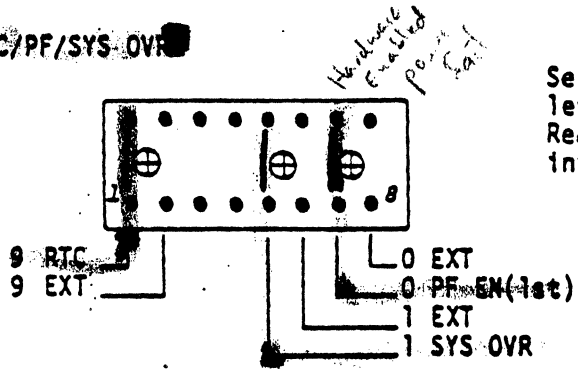

RTOM address selection (1st RTOM = $79_{16}$). Subsequent RTOM's are addressed 7A through 7E.

U107/K14 Interval Timer/RTC Freq



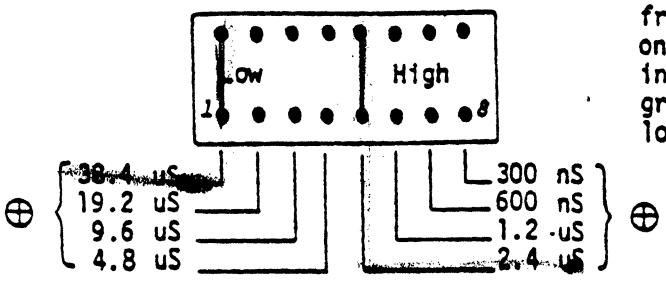

Real time Clock rate Selection, selection of Interval Timer to interrupt level sub-address 4 or 11 and selection of external interrupt to interrupt level sub-address 4 or 11.

U133/D24 RTC/PF/SYS OVR




Selection of enable to RTOM levels 00 & 01, selection of Real Time Clock or external interrupt to subaddress 09.

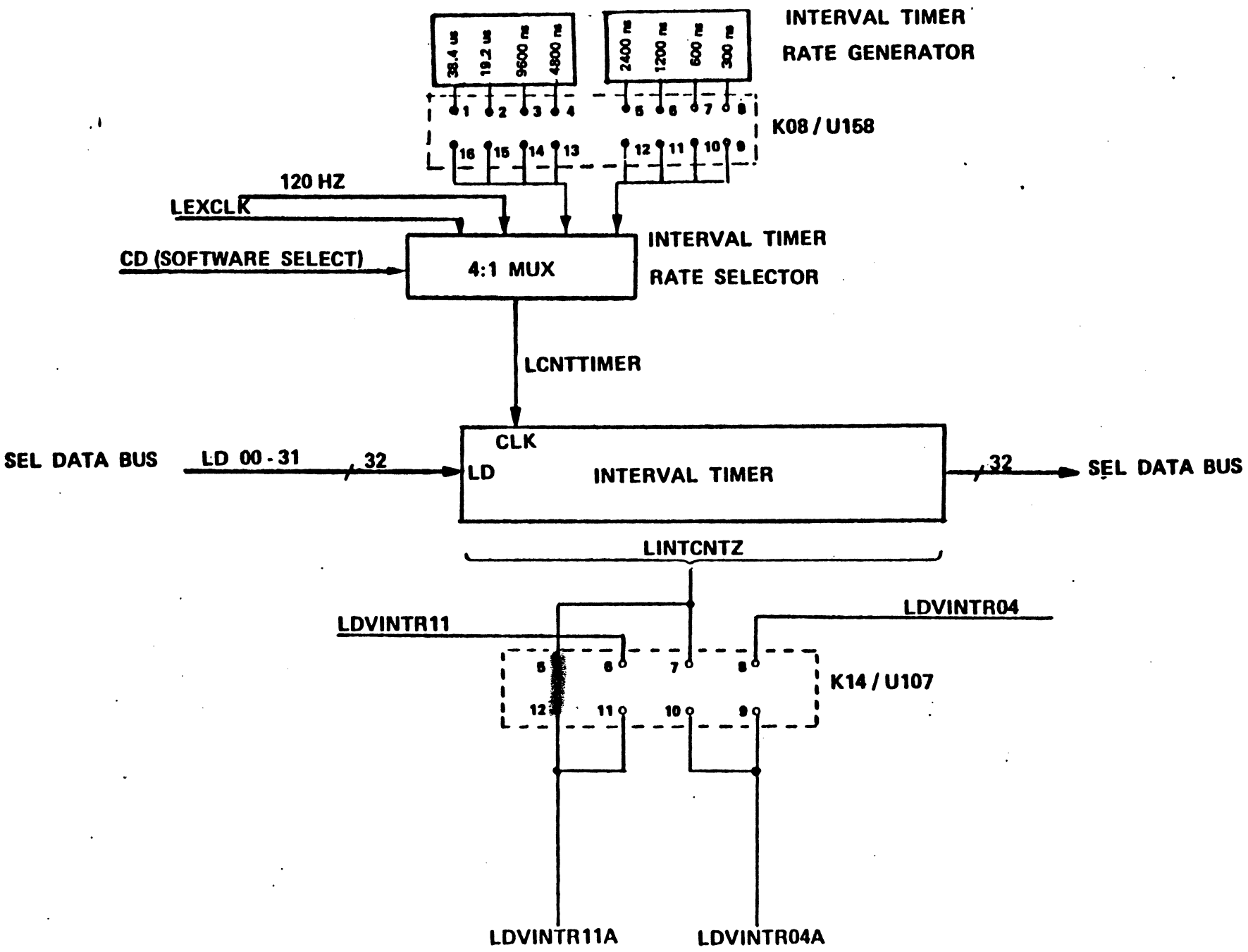
U158/K08 High/Low Freq Select



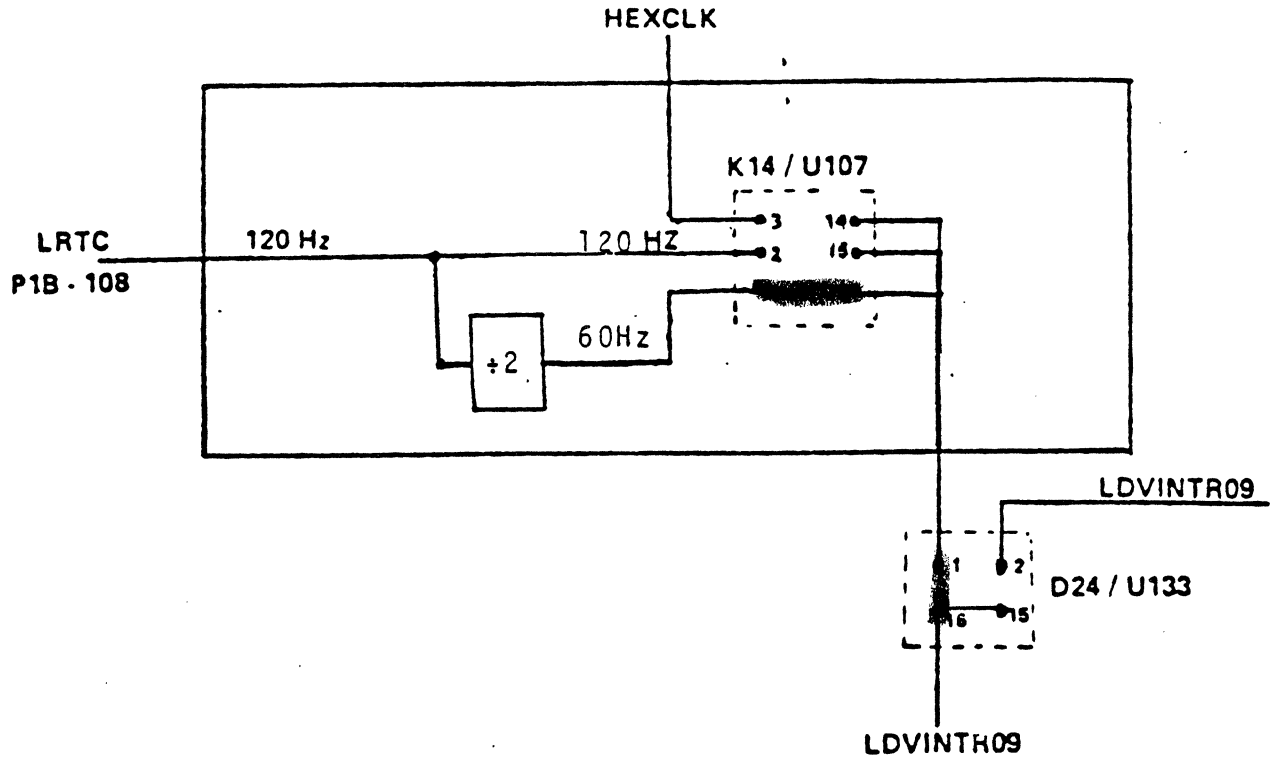

Interval Timer, high & low frequency selection. Note: only one jumper may be installed in each of the two groups (high frequency and low frequency).

⊕ Mutually Exclusive

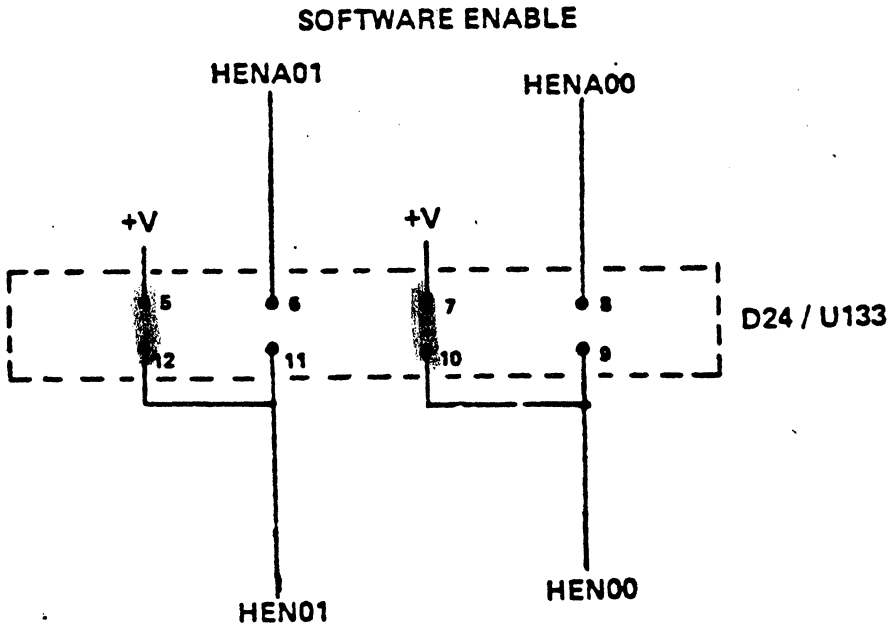
# RTOM 'INTERVAL TI[illegible]' JUMPERING



# RTOM 'REAL TIME CLOCK' JUMPERING




# HARDWARE ENABLING JUMPERS



SYSTEMS ENGINEERING LABORATORIES

# Technical Support Bulletin

| TITLE RTOM, Model No. 2345, Jumper Configurations | | TSB Number 0014 |
|---|---|---|
| Product SEL 32 | Model Number 2345 | Date 11/28/77 |

This bulletin contains the jumpering required to configure the new copper RTOM either as the first or subsequent RTOM in the system. Page 2 indicates the location and purpose of all jumpers contained on the board and Pages 3 through 5 give detailed jumpering information for the RTOM. The new RTOM performs the same functions as the Model 2117 & 2118 wire wrap RTOM's.

| TITLE RTOM, Model No. 2345, Jumper Configurations | TSB NO. 0014 |
|---|---|

Jumper configurations for the new Printed Wire Board Assembly, Real Time Option Module (RTOM), Model No. 2345.

| Jumper Switch Locations | Purpose |
|---|---|
| U13 | RTOM address selection (1st RTOM = $79_{16}$). Subsequent RTOM's are addressed 7A through 7E. |
| U107 | Real time Clock rate Selection, selection of Interval Timer to interrupt level sub-address 4 or 11 and selection of external interrupt to interrupt level sub-address 4 or 11. |
| U133 | Selection of enable to RTOM levels 00 & 01, selection of Real Time Clock or external interrupt to lever 09. |
| U158 | Interval Timer, high & low frequency selection. Note: only one jumper may be installed in each of the two groups (high frequency and low frequency). |

| First RTOM | Second or Subsequent RTOM | To Assign | Remove Jumper | Add Jumper | Logic Sheet Location |
|---|---|---|---|---|---|
| X | | Real-Time Clock to Interrupt Level 09 | U133-2 To U133-15 | U133-1 To U133-16 | 11 |
| | X | External Level 09 to RTOM Interrupt Level 09 (Disable Real-Time Clock) | U133-1 To U33-16 | U133-2 To U133-15 | 11 |
| X | | Constant Enable to RTOM Interrupt Level 00 (Power Fail Interrupt Level for the First RTOM in the System) | U133-8 To U133-9 | U133-7 To U133-10 | 11 |
| | X | Software Enable to RTOM Interrupt Level 00 (For the Second or Subsequent RTOM in the System) | U133-7 To U133-10 | U133-8 To U133-9 | 11 |
| X | | Constant Enable to RTOM Interrupt Level 01 (System Override Interrupt for the First RTOM in the System) | U133-6 To U133-11 | U133-5 To U133-12 | 11 |
| | X | Software Enable to RTOM Interrupt Level 01 (For the Second or Subsequent RTOM in the System) | U133-5 To U133-12 | U133-6 To U133-11 | 11 |

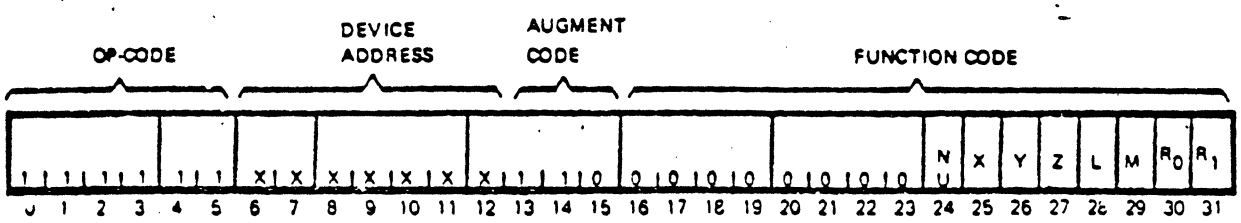
| First RTOM | Second or Subsequent RTOM | To Assign | Remove Jumper | Add Jumper | Logic Sheet Location |
|---|---|---|---|---|---|
| X | | Real-Time Clock to an External Clock | U107-2 To U 107-15<br>U107-1 To U107-16 | U107-3 To U107-14 | 11 |
| X | | Real-Time Clock to 120 Hz (Assuming 120 Hz at P1B-108) | U107-3 To U107-14<br>U107-1 To U107-16 | U107-2 To U107-15 | 11 |
| X | | Real-Time Clock to 60 Hz (Assuming 120 Hz at P1B-108) | U107-3 To U107-14<br>U107-2 To U107-15 | U107-1 To U107-16 | 11 |
| X | X | Interval Timer Count Zero Interrupt to RTOM Interrupt Level 04 | U107-7 To U107-9<br>U107-5 To U107-12 | U107-7 To U107-10 | 11 |
| X | X | External Level to RTOM Interrupt Level 04 | U107-7 TO U107-10 | U107-8 To U107-9 | 11 |
| X | X | Interval Timer Count Zero Interrupt to RTOM Interrupt Level 11 ($B_{16}$) | U107-6 To U107-11<br>U107-7 To U107-10 | U107-5 To U107-12 | 11 |
| X | X | External Level to RTOM Interrupt Level 11 ($B_{16}$) | U107-5 To U107-12 | U107-6 To U107-11 | 11 |
| To Select Interval Timer Count Rate, Install Only One Jumper in Each of the two Groups (Low Frequency and High Frequency) | | | | | |

Model #2345 RTOM Jumper Assignments

TSB Number 0014

| First RTOM | Second or Subsequent RTOM | To Assign | Remove Jumper | Add Jumper | Logic Sheet Location |
|---|---|---|---|---|---|
| | | Low Frequency Rate | | | |
| X | X | 4.8 Microseconds | ----- | U158 Pin 4 to 13 | Sheet 5 |
| X | X | 9.6 Microseconds | ----- | U158 Pin 3 to 14 | Sheet 5 |
| X | X | 19.2 Microseconds | ----- | U158 Pin 2 to 15 | Sheet 5 |
| | | 38.4 Microseconds | ----- | U158 Pin 1 to 16 | Sheet 5 |
| | | High Frequency Rate | | | |
| X | X | 300 Nanoseconds | ----- | U158 Pin 8 to 9 | Sheet 5 |
| X | X | 600 Nanoseconds | ----- | U158 Pin 7 to 10 | Sheet 5 |
| X | X | 1.2 Microseconds | ----- | U158 Pin 6 to 11 | Sheet 5 |
| X | X | 2.4 Microseconds | ----- | U158 Pin 5 to 12 | Sheet 5 |

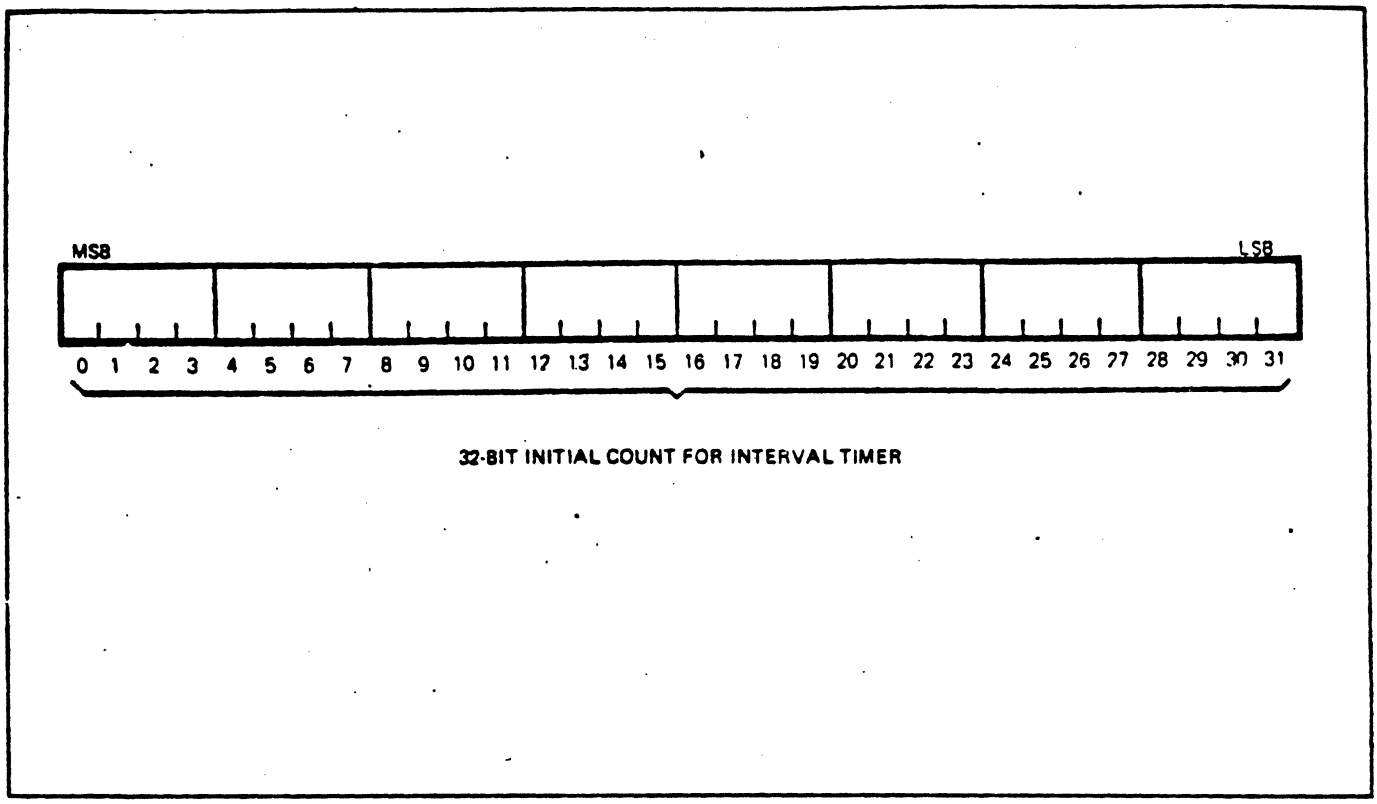
INSTRUCTION WORD



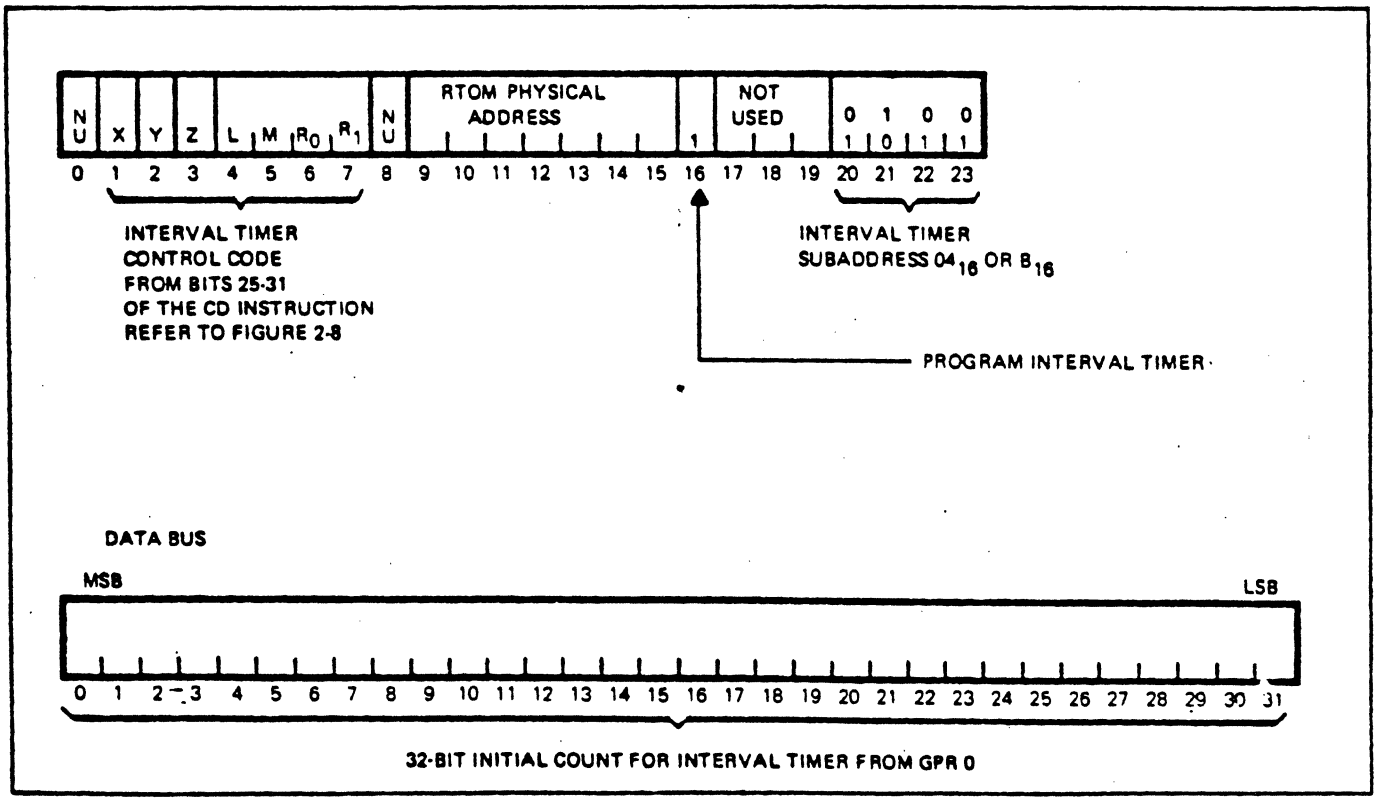

| | |
|---|---|
| X (BIT 25) = 1 | SPECIFIES READ TIMER, CAUSING THE 32-BIT CONTENTS OF THE TIMER TO BE LOADED INTO GPR 0 AS ILLUSTRATED IN FIGURE 2-11. |
| Y (BIT 26) = 1 | SPECIFIES PROGRAM INTERVAL TIMER AND THAT BITS 27-31 ARE VALID. |
| Z (BIT 27) = 1 | ENABLES (START) INTERVAL TIMER |
| Z (BIT 27) = 0 | DISABLES (STOP) INTERVAL TIMER |
| L (BIT 28) = 1 | LOADS BITS 00-31 FROM THE GPR 0 INTO THE INTERVAL TIMER BITS 00-31. |
| L (BIT 28) = 0 | DOES NOT ALTER THE STORE COUNT. |
| M (BIT 29) = 1 | GENERATES MULTIPLE INTERRUPT. WHEN COUNT ZERO IS REACHED, GENERATE INTERRUPT, RELOAD INITIAL COUNT, AND CONTINUE COUNTING. |
| M (BIT 29) = 0 | GENERATES SINGLE INTERRUPT ON COUNT ZERO AND THEN COUNT NEGATIVE. |

| $R_0$ (BIT 30) | $R_1$ (BIT 31) | SELECT COUNT RATE |
|---|---|---|
| 0 | 0 | SELECT HIGH FREQUENCY PER TABLE 2-1 |
| 0 | 1 | SELECT LOW FREQUENCY PER TABLE 2-1 |
| 1 | 0 | SELECT 120 Hz |
| 1 | 1 | SELECT EXTERNAL CLOCK |

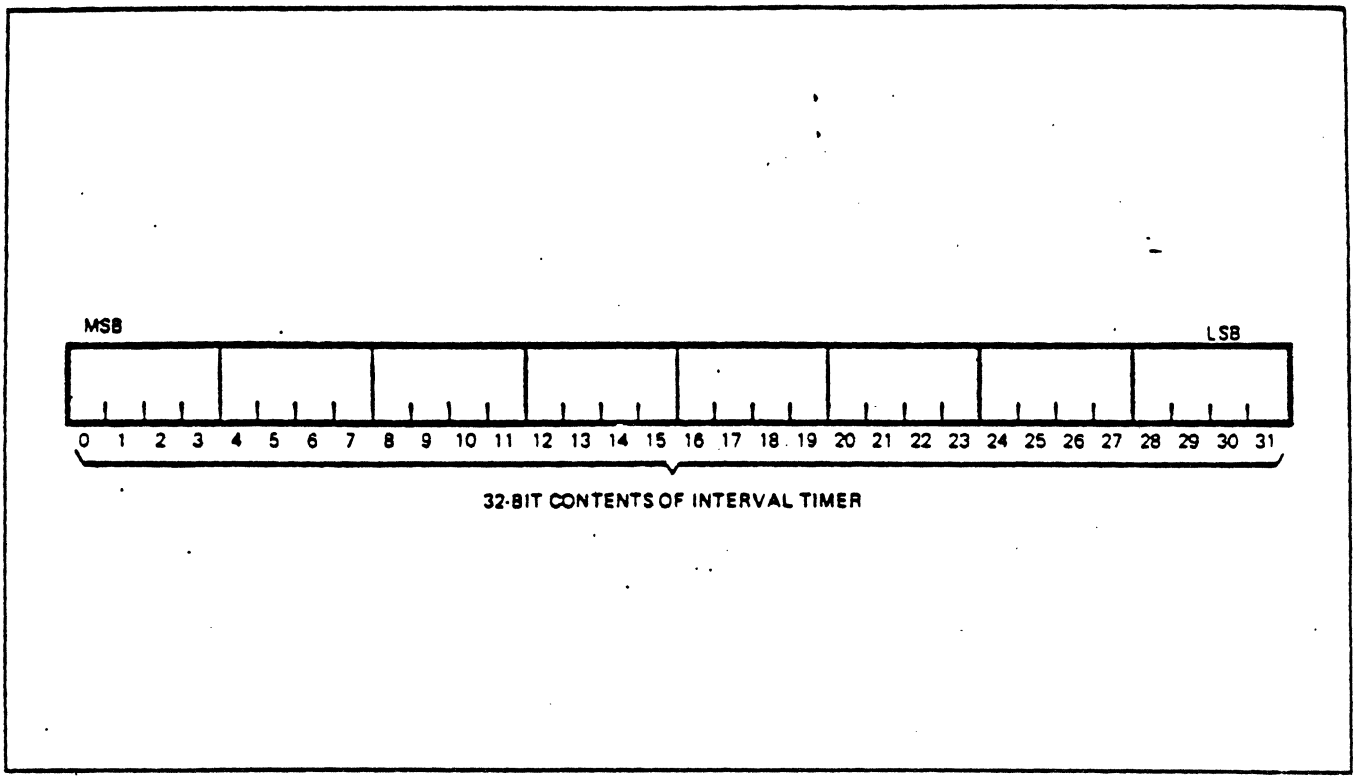
RTOM Interval Timer Command Device Instruction Format



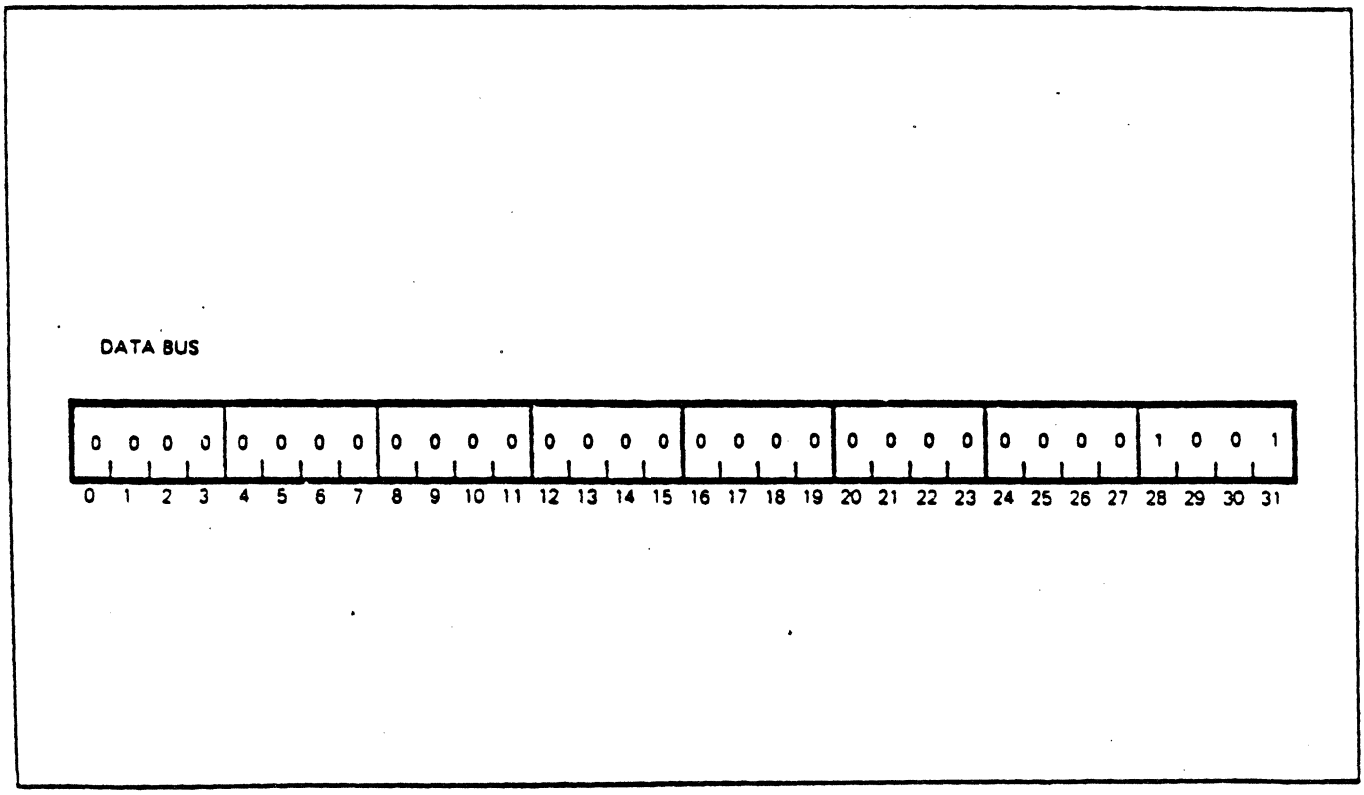

CPU/GPRO Load Count Format




WDOT Control Interval Timer SEL Bus Format




Read Interval Timer, CPU/GPRO Format




RSTX Read Interval Timer SEL Bus Format

| *DEV XX = | 03 | YY | 7Z | 04 OR 08 |
|---|---|---|---|---|

| | |
|---|---|
| XX | THE CD SOFTWARE ADDRESS OF THE INTERVAL TIMER IN THE RANGE OF $00_{16}$ THROUGH $7F_{16}$ |
| =03 | DEFINES THE INTERVAL TIMER AS A CLASS 3 DEVICE |
| YY | YY IS THE INTERRUPT LEVEL OF THE INTERVAL TIMER IN THE RANGE OF $14_{16}$ THROUGH $7F_{16}$ |
| 7Z | Z IS THE 4 LEAST SIGNIFICANT BITS OF THE RTOM PHYSICAL ADDRESS IN RANGE OF $9_{16}$ THROUGH $F_{16}$ |
| 04 OR 08 | DEFINES THE SUBADDRESS OF THE INTERVAL TIMER ON THE RTOM |

Interval Timer ICL Card Format

## "BLACK BOX" AIDS IN DIAGNOSING INTERRUPT ACTIVE PROBLEMS

Several users have asked for assistance in troubleshooting interrupt active problems they have been experiencing. The schematic below can be used to build a diagnostic aid for this condition. All components are readily available and inexpensive.

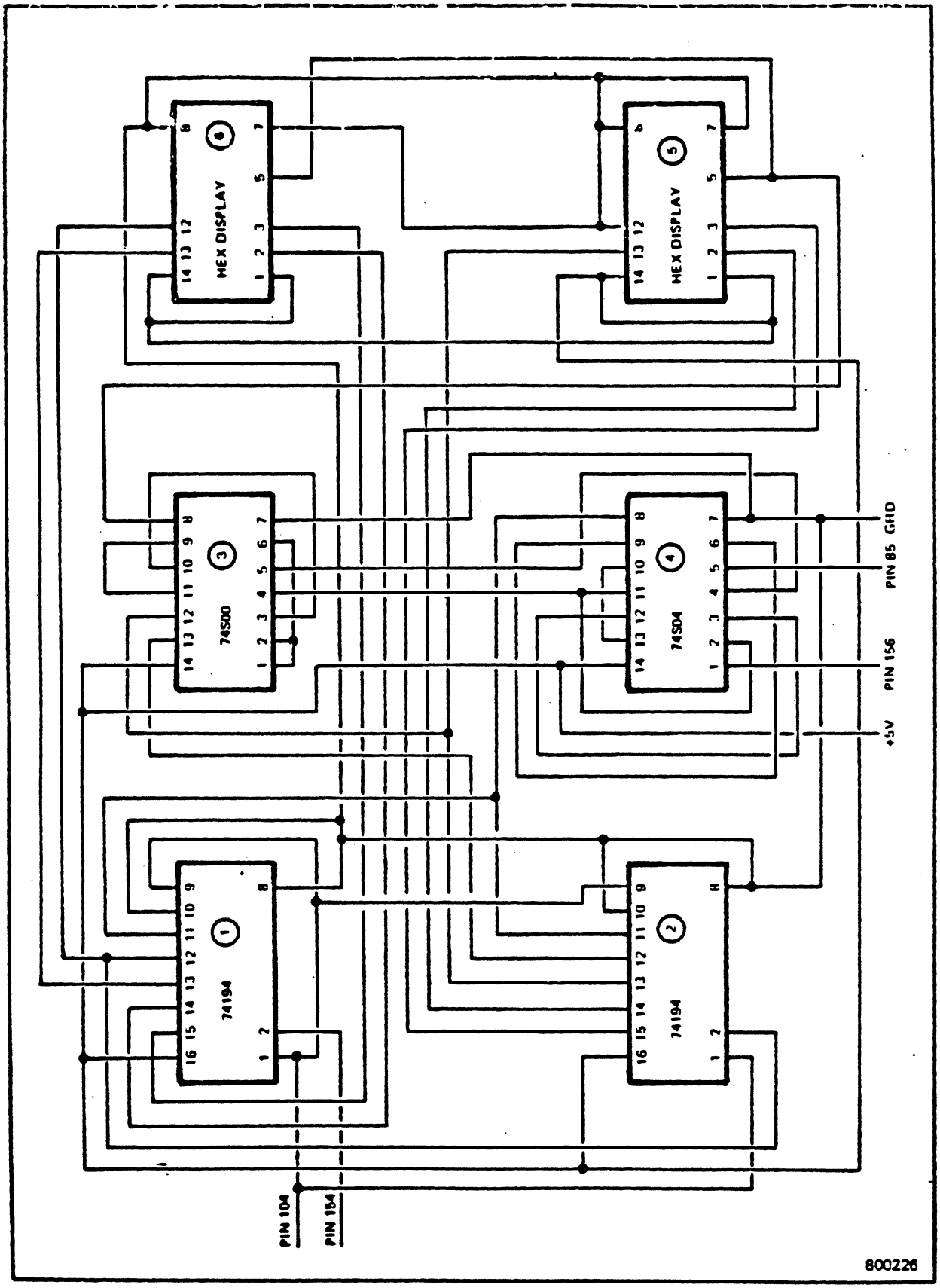
When this "black box" is hooked onto the SelBUS as indicated, it will display the hexadecimal number of the last active interrupt. As the interrupt level corresponds to a particular controller, this narrows the range of the problem rather quickly.

### Wire List

| From | To | From | To | From | To |
|---|---|---|---|---|---|
| 1/1 | 1/9 | 2/1 | 1/1 | 3/1 | 3/6 |
| 1/8 | 6/8 | 2/8 | 2/10 | 3/1 | 3/2 |
| 1/8 | 2/8 | 2/9 | 1/9 | 3/3 | 3/10 |
| 1/8 | 1/10 | 2/11 | 4/8 | 3/4 | 4/11 |
| 1/11 | 2/11 | 2/12 | 3/13 | 3/5 | 4/4 |
| 1/12 | 2/2 | 2/13 | 5/13 | 3/7 | 4/7 |
| 1/12 | 6/12 | 2/13 | 3/12 | 3/8 | 5/5 |
| 1/13 | 6/13 | 2/14 | 5/2 | 3/9 | 3/11 |
| 1/14 | 6/2 | 2/15 | 5/3 | 3/14 | 4/14 |
| 1/15 | 6/3 | 2/16 | 5/14 | 3/14 | 4/14 |
| 1/16 | 3/14 | | | | |
| 1/16 | 2/16 | | | | |
| 4/2 | 4/11 | 5/1 | 5/14 | 6/1 | 5/14 |
| 4/3 | 4/12 | 5/5 | 6/5 | 6/1 | 6/14 |
| 4/6 | 4/9 | 5/7 | 6/7 | | |
| 4/7 | 2/8 | 5/7 | 5/8 | | |
| 4/10 | 4/13 | 5/8 | 6/8 | | |
| | | 5/12 | 6/7 | | |

| From | To | From | To |
|---|---|---|---|
| 1/1 | Pin 104 Backplane | 4/5 | Pin 85 Backplane |
| 1/2 | Pin 154 Backplane | 4/14 | +5v Backplane (Pin 181 or 182) |
| 4/1 | Pin 156 Backplane | 4/7 | Ground Backplane (Pin 1 or 2) |

| Parts List | Quantity | Description |
|---|---|---|
| 74194 | 2 | Standard 74 Series TTL Integrated Circuit |
| 74S00 | 1 | Standard 74 Series TTL Integrated Circuit |
| 74S04 | 1 | Standard 74 Series TTL Integrated Circuit |
| Hex Display | 2 | Standard 74 Series TTL Integrated Circuit |




LAST INTERRUPT ACTIVE
MONITOR

# Real-Time Option Module (RTOM) Worksession

Objectives:

This worksession will reinforce the classroom presentation given by the Instructor to insure that you understand the purpose and characteristics of the RTOM and how it is used.

Reference Reading:

A. SEL 32 Architecture Course Student Workbook, Section on RTOM.

B. 32/70 Series Computer Technical Manual, Page 1-8.

1. What basic function does an RTOM perform?
   32 bit interval timer Provides [illegible] for the system
   Real Time Clock

2. How many Priority Interrupts can be assigned to an RTOM?
   16

3. What would be an example of an External Interrupt?
   Pushbutton or Switch, Timer
   Relay Contact

4. Explain what information is available to the CPU from this ICL card. *INT28=96
   INTERRUPT PRIORITY LEVEL=28
   Scratchpad Location Inter. → $A8 = 80_{16} + 28_{16}$ ←
   1ST RTOM = 9 Physical Address 19
   SubAddress = 6

5. Does the RTOM ever do transfers on the SEL Bus to Memory? If so, why?
   No

6. What could the external stimulus output from an RTOM be used for?
   As interrupt for another RTOM

7. What does the RTOM's RAM contain?
   I[illegible]ls
   Priority
   (one's complement

8. What function does the 1st RTOM perform in the system?

RTM Call Monitor Attention RTC Interval Timer

9. How many External Interrupts are available on the 1st RTOM? 2nd RTOM?

PSD PSW 1ST RTOM Ø 2ND RTOM 16

10. What could the interval timer on the RTOM be used for?

32 bit down counter

11. What is the Physical SEL Bus Address of the 1st RTOM?

79

12. How many RTOM's could be in a System?

7 (1 required)

13. What Sub Address could be assigned to the Interval Timer?

4 or 11 (B)

14. What difference does it make which Sub Address is assigned? How is the Sub Address assignment made?

The subaddress determines the address on the RTOM board the interrupt is assigned.

15. What Sel Bus Priority is used on each RTOM?

Doesn't

16. Which General Purpose Register in the CPU is used with the Interval Timer?

GPR Ø

17. Write a short program to check out an interrupt level on the RTOM.

EI

18. If you connected an external input line to LDVINTR12 on P1C, what interrupt level could you assign to this line and what Sub Address must you assign to it? 2C - IL

3.5A

19. What precaution should be taken when assigning INTERRUPT LEVELS to EXTERNAL INPUTS? Verify pin is unused

20. List the 4 rates at which the interval timer can be clocked.

| Hi | Lo | External | RTC |
|---|---|---|---|
| 38.4μS | 2.4μSec | | |

# DAY 7

# SECTION 7

# CENTRAL PROCESSING UNIT

# SEL BUS

# CENTRAL PROCESSOR UNIT (CPU)

WORD LENGTH IS 32 BITS

- OVERLAPPED INSTRUCTION EXECUTION

  1.2 MICROSECOND INSTRUCTION EXECUTION (TYPICAL)

- OVERLAPPED INSTRUCTION OPERAND FETCH

  .8 MIPS (STRAIGHT LINE CODE)

  545K WHETS

- MICROPROGRAMMED (FIRMWARE)

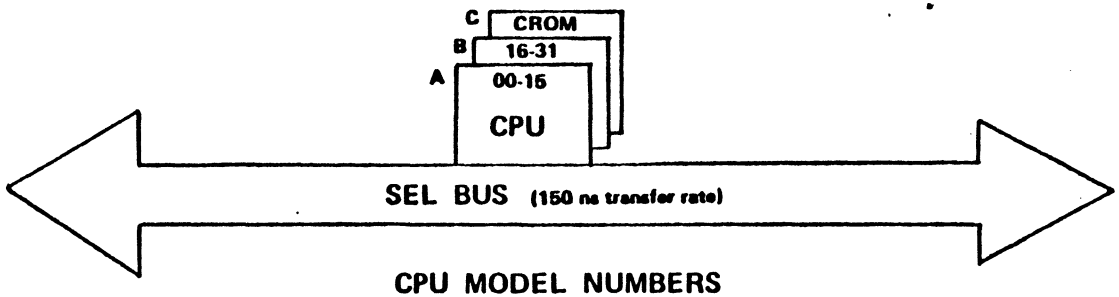
  150 NS PER MICRO-INSTRUCTION

  4K X 64 BIT ROM (000-FFF)

- STANDARD FLOATING POINT (FIRMWARE)
- HAS 8 GENERAL PURPOSE REGISTERS
- HARDWARE/FIRMWARE MEMORY MANAGEMENT (MAPPING)
- OPERATES IN TWO MODES:

| PSW | PSD | |
|---|---|---|
| RTM OPERATING SYSTEM ONLY | RTM OPERATING SYSTEM | MPX OPERATING SYSTEM |
| 161 INSTRUCTIONS | 161 + INSTRUCTIONS | 187 INSTRUCTIONS |
| SYSTEM INTERRUPTS ON RTOM | SYSTEM INTEGRITY TRAPS | SYSTEM INTEGRITY TRAPS |
| DIRECT ADDRESS 128 KW | DIRECT ADDRESS 128 KW | MAPPING (ADDRESS 16 MB) |
| NO CLASS 'F' I/O | CLASS 'F' I/O | CLASS 'F' I/O |
| I/O (ADDRESS 128 KW ONLY) | I/O (ADDRESS 16 MB) | I/O (ADDRESS 16 MB) |

# CENTRAL PROCESSING UNIT



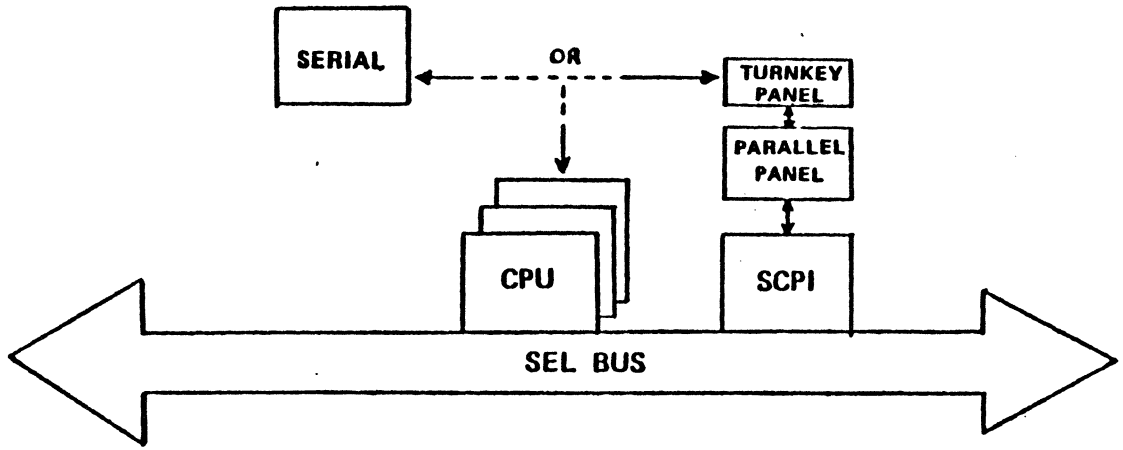

CPU MODEL NUMBERS

2000 WIRE/WRAP 3255

2003 COPPER

2005 CPU-IPU 3277

# SYSTEM CONTROL PANELS

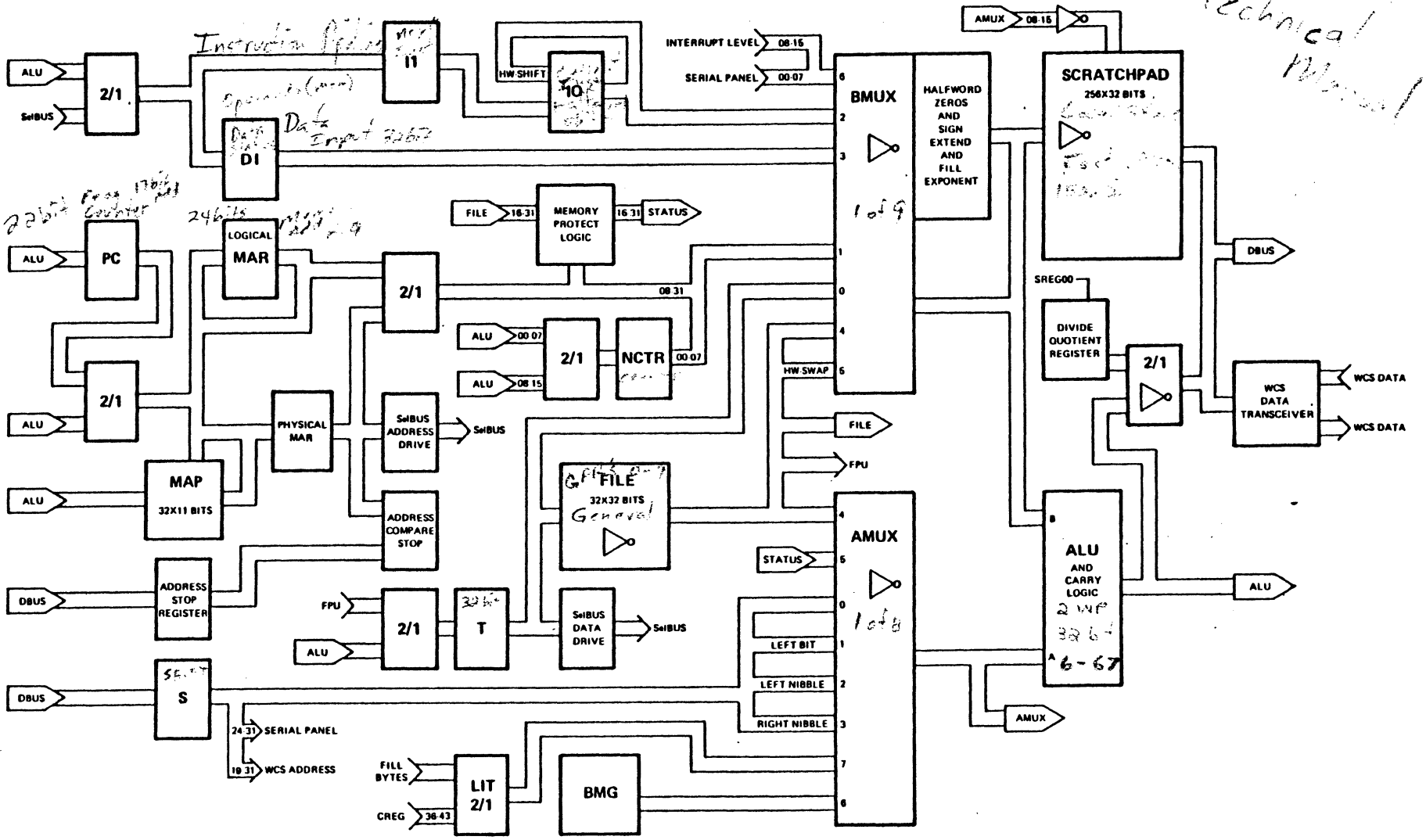



SERIAL CONTROL PANEL – MODEL 2146

PARALLEL CONTROL PANEL WITH SCPI – MODEL 2346

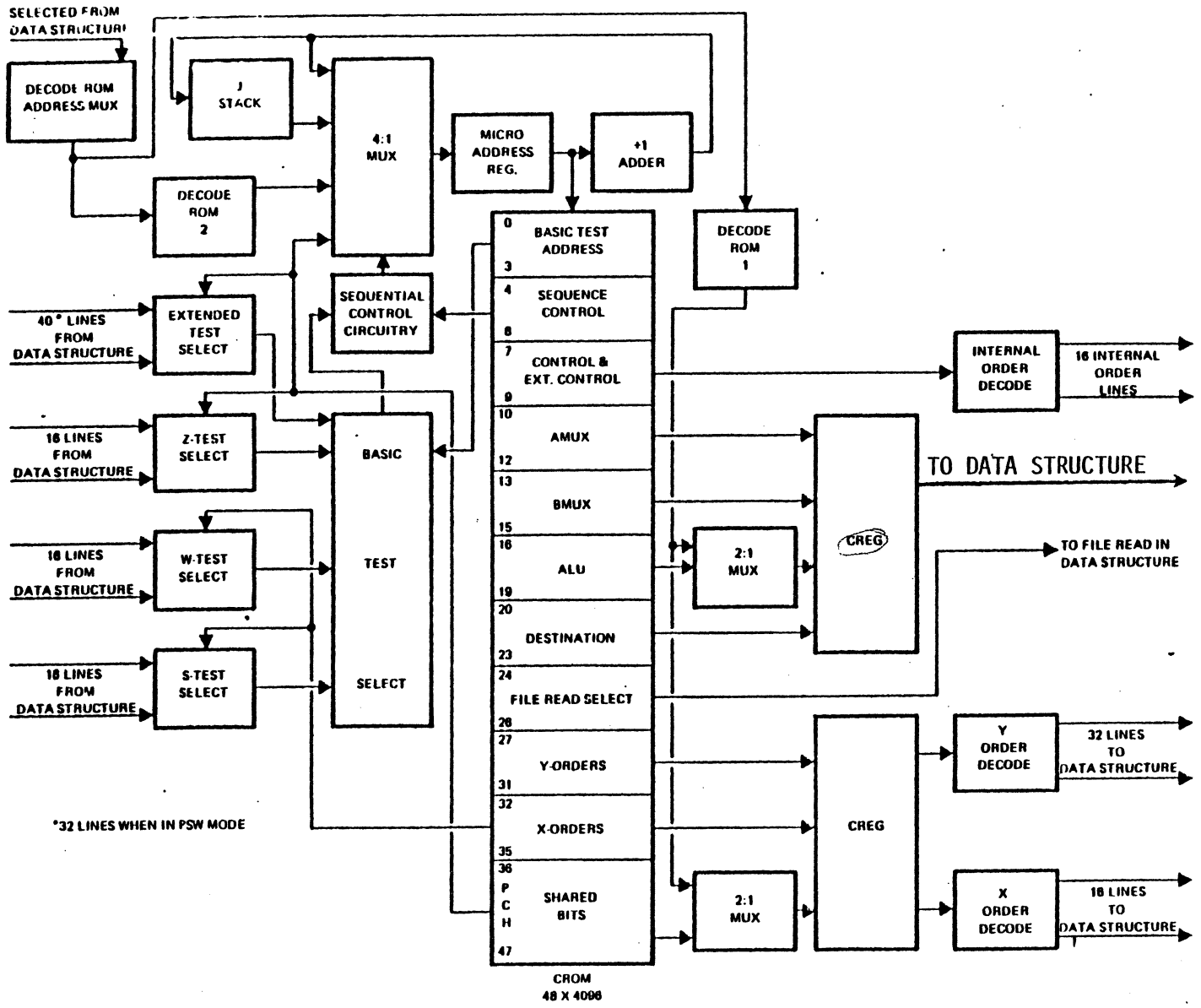
SCPI – MODEL 2142

# CPU DATA STRUCTURE




32/75 Data Structure

# CPU MICRO-ENGINE



SYSTEMS ENGINEERING LABORATORIES

# TECHNICAL SUPPORT BULLETIN

| TITLE CONFIGURATION JUMPERS | | TSB No. 0107A |
|---|---|---|
| Product 32/75A CPU/IPU | Model No. ~~2005~~ 3277 | Date 6-11-80 |

The following is a definition of the jumper settings at position L24 on the "C" board (160-103438) of the CPU/IPU.

| POSITION | IN | OUT |
|---|---|---|
| 1-16 | Enable High Baud Rate | Disable High Baud Rate |
| 2-15 | Enable Low Baud Rate | Disable Low Baud Rate |
| 3-14 | Enable Standard Baud Rate | Disable Standard Baud Rate |
| 4-13 | Disable Trap | Enable Trap |
| 5-12 | IPU For IPU (w/ PSD) | CPU |
| 6-11 | PSD/PSW Mode | PSW Mode Only 3255 |
| 7-10 | Parallel Panel | Serial Panel |
| 8-9 | N/U | N/U |

## Jumper Position 4-13

This jumper is used only in the CPU and only in a configuration under the RTM operating system. When in this configuration, it is required to have SELBUS P1B-33 hard wired to the desired RTOM EXTERNAL INTERRUPT. Level X'2A' is recommended (RTOM P1C-33).

## Jumper Position 5-12

When this jumper is in, the 3 board processor becomes the IPU (INTERNAL PROCESSING UNIT). When this jumper is out, the processor becomes the CPU (CENTRAL PROCESSING UNIT). When both processors are on the SELBUS, one must be a CPU and the other an IPU.

## Jumper Position 6-11

This jumper must be in both the CPU and the IPU, enabling PSD mode.

## Jumper Position 7-10

This jumper, when out, enables the Serial Panel and must be out in the IPU.

| TITLE<br>CONFIGURATION JUMPERS | TSB No.<br>0107A |
|---|---|

[redacted] configured with a CPU and an IPU, SELBUS priority 22 is dedicated to the IPU and must be ena[redacted]e., priority 22 jumper [redacted] the SELBUS [redacted] terminator.

[redacted] previous systems with a parallel panel, priority 22 was normal[redacted]ated to the SCSI [redacted] in the [redacted]here a parallel panel is to [redacted]ation, it is recommended that the SCSI priority be set one the TLC 20.

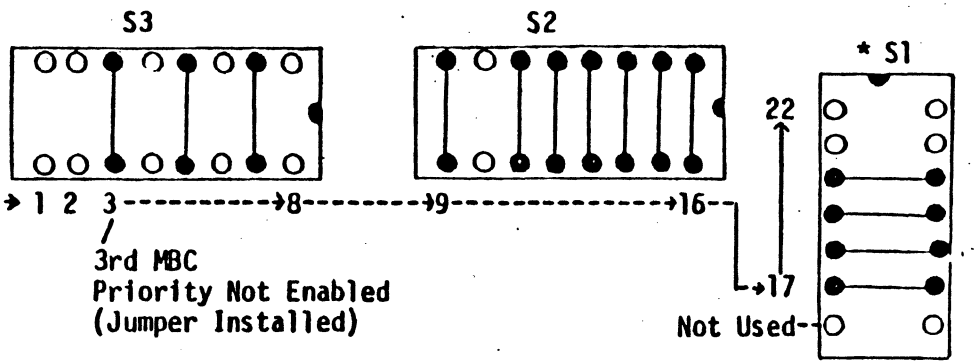
## SEL BUS PRIORITY ENABLE JUMPER CONNECTIONS

The sockets that are to be jumpered for SEL Bus priority are located on the SEL Bus terminator circuit card with coordinates: S1, S2, S3. When an IOM or MBC is installed, the appropriate jumper points are left OPEN (NOT JUMPERED).

As an example, the following assignments are made for a given system; and the jumpered sockets would result, as is illustrated for S1, S2, and S3.

Example of Assignment of SEL Bus Priority

a. First MBC----------> 1
b. Second MBC----------> 2
c. FHD IOM ----------> 4
d. CD IOM ----------> 6
e. MT IOM ----------> 8
f. ADS IOM ----------> 10
g. TLC IOM ----------> 21
h. SCP IOM ----------> 22

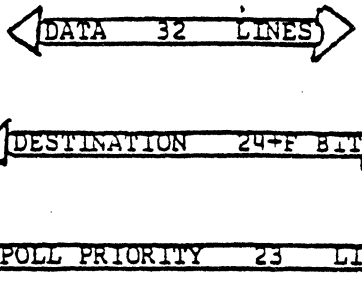



* S1 is a 14 pin DIP socket; S2 & S3 are 16 pin DIP sockets.

PICTORIAL ILLUSTRATION OF S1, S2, S3 JUMPER SOCKETS ON SEL BUS TERMINATOR CIRCUIT CARD

NOTE: EXAMPLE ONLY

SEL BUS LINES



## Useful Test Programs

Using the skills you have learned thus far, try these programs:

A. Clear Memory - This program will write the contents of GPR0 into all memory locations except location 4.

Procedure: Enter the following program.

| Location | Hex | Instruction |
|---|---|---|
| 0000 | 000D0002 | SEA, NOP |
| 0004 | D420000C | STW R0, C, I1 |
| 0008 | F4C00004 | BIW R1, 4 |

1. SYSTEM RESET
2. RUN

Observe program run for about .5 seconds.

3. Read several locations to verify they are cleared.

B. Address of address - This program writes the address of the address through memory. This is useful for troubleshooting suspected address errors in memory.

Procedure: 1. Enter the following program.

| Location | Hex | Instruction |
|---|---|---|
| 0000 | 000D0002 | SEA, NOP |
| 0004 | D4A00000 | STW 1,0,1 |
| 0008 | F4C00004 | BIW 1,0 |

2. System Reset
3. Put X'C' in Reg1. (KB-ENTER C-WRITE-1/REG)
4. Press and release run.

Observe program run and halt.

5. Read several locations to see that the data in a location equals its own location address.

Example: location 100, data = 100
location 3F4, data = 3F4

C. Add bit in memory - This program is useful in checking for a gross failure in the CPU. It is also a building block for future programs because it may be used as a subroutine to count things. The program adds bit 31 to location 100.

Procedure: 1. Enter the following program.

| Location | Hex | Instruction |
|---|---|---|
| 0000 | A3880103 | ABM 31, 100 |
| 0004 | EC000000 | BU 0 |

C. Add bit in memory (Continued):

2. System Reset
3. Run

Observe program running continuously.

The only indication the program is running is the RUN indicator is lit. But we can see things happen by monitoring the ABM instruction executing on location 100.

4. Use extended function #1
   KB - enter 100
   WRITE - MA
   EXTENDED FUNCTION - #1

Now observe 'A' display has memory address 100 displayed and the 'B' display is constantly being updated.

## SelBUS Pin Assignments

| Pin | Signal | Description |
|---|---|---|
| 1 | GND | Ground |
| 2 | GND | Ground |
| 3 | +5vA | Positive 5 volts DC from power supply |
| 4 | +5vA | Positive 5 volts DC from power supply |
| 5 | LD01 | Data bit 01 |
| 6 | LD00 | Data bit 00 |
| 7 | LD03 | Data bit 03 |
| 8 | LD02 | Data bit 02 |
| 9 | LD04 | Data bit 04 |
| 10 | GND | Ground |
| 11 | LD06 | Data bit 06 |
| 12 | LD05 | Data bit 05 |
| 13 | +5vB | Positive 5 volts DC from battery backup |
| 14 | LD07 | Data bit 07 |
| 15 | LD09 | Data bit 09 |
| 16 | LD08 | Data bit 08 |
| 17 | LD11 | Data bit 11 |
| 18 | LD10 | Data bit 10 |
| 19 | GND | Ground |
| 20 | LD12 | Data bit 12 |
| 21 | LD14 | Data bit 14 |
| 22 | LD13 | Data bit 13 |
| 23 | +5vB | Positive 5 volts DC from battery backup |
| 24 | LD15 | Data bit 15 |
| 25 | LD17 | Data bit 17 |
| 26 | LD16 | Data bit 16 |
| 27 | LD19 | Data bit 19 |
| 28 | LD18 | Data bit 18 |
| 29 | LD20 | Data bit 20 |
| 30 | GND | Ground |
| 31 | LD22 | Data bit 22 |
| 32 | LD21 | Data bit 21 |
| 33 | LCPUTRAP | CPU trap |
| 34 | LD23 | Data bit 23 |
| 35 | LD25 | Data bit 25 |
| 36 | LD24 | Data bit 24 |
| 37 | LD27 | Data bit 27 |
| 38 | LD26 | Data bit 26 |
| 39 | GND | Ground |
| 40 | LD28 | Data bit 28 |
| 41 | LD30 | Data bit 30 |
| 42 | LD29 | Data bit 29 |
| 43 | +5vB | Positive 5 volts DC from battery backup |
| 44 | LD31 | Data bit 31 |
| 45 | GND | Ground |
| 46 | GND | Ground |
| 47 | +5vA | Positive 5 volts DC from power supply |
| 48 | +5vA | Positive 5 volts DC from power supply |
| 49 | LDT01 | Destination bit 01 |
| 50 | LDT00 | Destination bit 00 |

## SelBUS Pin Assignments (continued)

| Pin | Signal | Description |
|---|---|---|
| 51 | LDT03 | Destination bit 03 |
| 52 | LDT02 | Destination bit 02 |
| 53 | LDT04 | Destination bit 04 |
| 54 | GND | Ground |
| 55 | LDT06 | Destination bit 06 |
| 56 | LDT05 | Destination bit 05 |
| 57 | LDT08 | Dsetination bit 08 |
| 58 | LDT07 | Destination bit 07 |
| 59 | LDT10 | Destination bit 10 |
| 60 | LDT09 | Destination bit 09 |
| 61 | LDT12 | Destination bit 12 |
| 62 | LDT11 | Destination bit 11 |
| 63 | GND | Ground |
| 64 | LDT13 | Destination bit 13 |
| 65 | LDT15 | Destination bit 15 |
| 66 | LDT14 | Destination bit 14 |
| 67 | LDT17 | Destination bit 17 |
| 68 | LDT16 | Destination bit 16 |
| 69 | LDT19 | Destination bit 19 |
| 70 | LDT18 | Destination bit 18 |
| 71 | LDT21 | Destination bit 21 |
| 72 | LDT20 | Destination bit 20 |
| 73 | LDT22 | Destination bit 22 |
| 74 | GND | Ground |
| 75 | LDTF | Byte transfer tag signal |
| 76 | LDT23 | Destination bit 23 |
| 77 | LREADY | Ready signal |
| 78 | GND | Ground |
| 79 | GND | Ground |
| 80 | LSYNC | Sync interrupt poll |
| 81 | LCLKE | System clock early |
| 82 | GND | Ground |
| 83 | GND | Ground |
| 84 | LCLK | System clock |
| 85 | LCLKL | System clock late |
| 86 | GND | Ground |
| 87 | GND | Ground |
| 88 | LSTSC | Stop system clock |
| 89 | +15v | Positive 15 volts DC |
| 90 | +15v | Positive 15 volts DC |
| 91 | GND | Ground |
| 92 | GND | Ground |
| 93 | GND | Ground |
| 94 | GND | Ground |
| 95 | -15v | Negative 15 volts DC |
| 96 | -15v | Negative 15 volts DC |
| 97 | LCPUSC | CPU stop clock |
| 98 | GND | Ground |
| 99 | GND | Ground |
| 100 | LCLP | Clock problem |

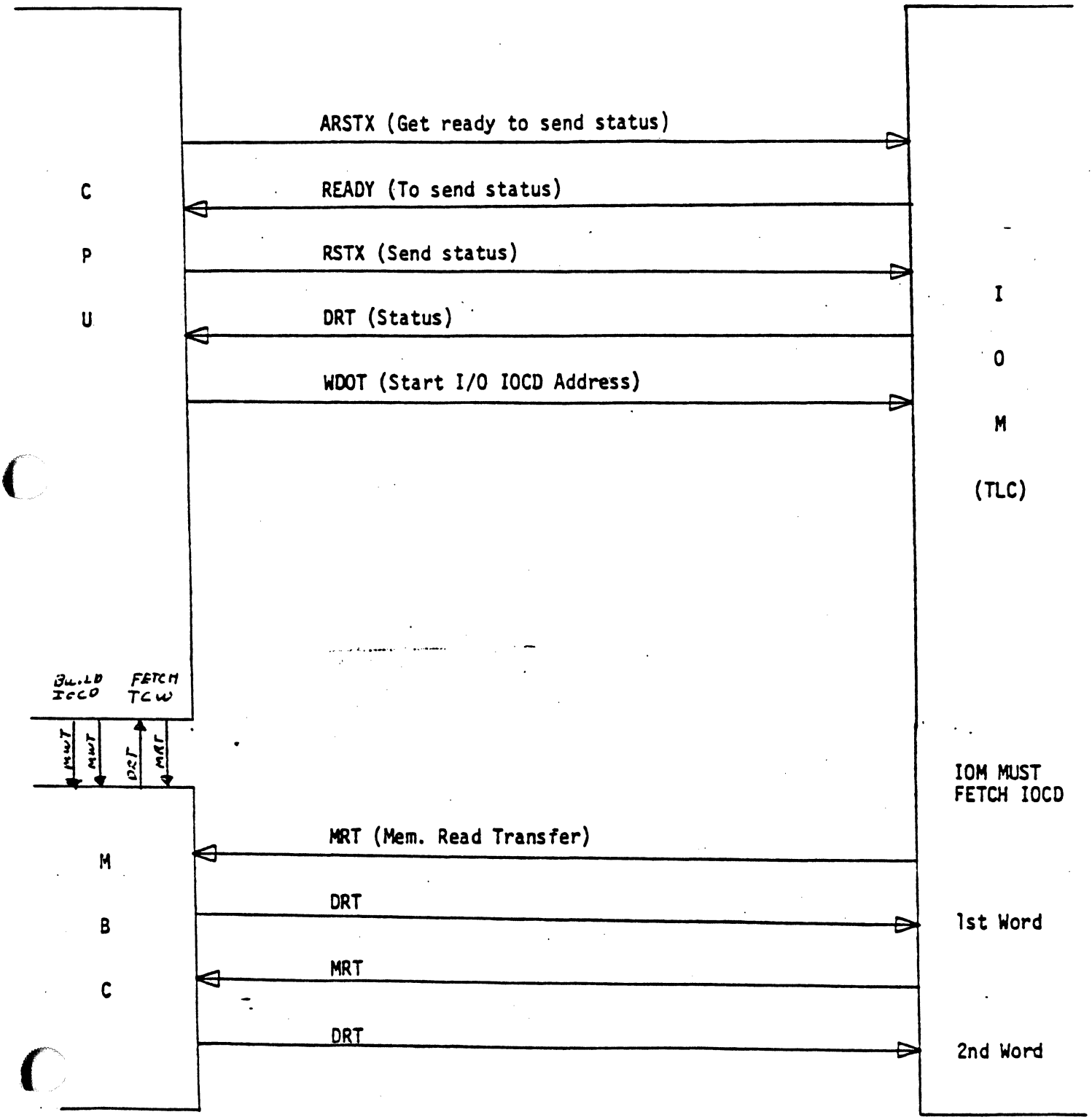
## SelBUS Pin Assignment (continued)

| Pin | Signal | Description |
|---|---|---|
| 101 | LINH00 | Memory busy inhibit 00 |
| 102 | GND | Ground |
| 103 | LINH01 | Memory busy inhibit 01 |
| 104 | LRESET | I/O reset |
| 105 | LINH02 | Memory busy inhibit 02 |
| 106 | LCLKOV | Clock override |
| 107 | LINH03 | Memory busy inhibit 03 |
| 108 | LRTC | Real time clock |
| 109 | +12v margin | Margin MOS Memories (DSS test stand only) |
| 110 | -5v margin | Margin MOS Memories (DSS test stand only) |
| 111 | GND | Ground |
| 112 | LPF | Power fail |
| 113 | LPFMEM | Power fail memory |
| 114 | Ext +5v | External positive 5 volt DC |
| 115 | LTRC | Transmitting C |
| 116 | LMUNLK | Unlock memory |
| 117 | LREFM | Refresh memory |
| 118 | LERROR | Memory error |
| 119 | LECK0 | Echo bit 0 |
| 120 | LRTRY | Retry |
| 121 | LECK1 | Echo bit 1 |
| 122 | GND | Ground |
| 123 | LD32/P0 | Parity bit byte 0 |
| 124 | LCHBSY | Channel busy |
| 125 | LD33/P1 | Parity bit byte 1 |
| 126 | LTX | Transfer tag bit |
| 127 | LD34/P2 | Parity bit byte 2 |
| 128 | LCPUSTART | CPU start |
| 129 | LDP0/P3 | Parity bit byte 3 |
| 130 | LTA | Transfer acknowledge |
| 131 | GND | Ground |
| 132 | LCNT0 | Control 0 tag signal |
| 133 | LSCPATTN | System control panel attention |
| 134 | LCNT1 | Control 1 tag signal |
| 135 | LPEF | Pre-refresh memory |
| 136 | LCPU | CPU bit line signal |
| 137 | +5vA | Positive 5 volts DC from power supply |
| 138 | +5vA | Positive 5 volts DC from power supply |
| 139 | GND | Ground |
| 140 | GND | Ground |
| 141 | LPR00 | Poll priority bit 00 |
| 142 | LRD | Read tag signal |
| 143 | HPR01 | Poll priority bit 01 |
| 144 | LMEM | Memory tag signal |
| 145 | HPR02 | Poll priority bit 02 |
| 146 | GND | Ground |
| 147 | HPR03 | Poll priority bit 03 |
| 148 | LUS | Memory unsuccessful tag bit |
| 149 | HPR04 | Poll priority bit 04 |

## SelBUS Pin Assignment (continued)

| Pin | Signal | Description |
|---|---|---|
| 150 | +5vB | Positive 5 volts DC from battery backup |
| 151 | HPR05 | Poll priority bit 05 |
| 152 | LMLK | Memory lock tag bit |
| 153 | HPR06 | Poll priority bit 06 |
| 154 | LIPOL | Interrupt poll |
| 155 | GND | Ground |
| 156 | LEOIP | End of interrupt poll |
| 157 | HPR07 | Poll priority bit 07 |
| 158 | LINTR | Interrupt request |
| 159 | HPR08 | Poll priority bit 08 |
| 160 | LIOIN | I/O interrupt inhibit |
| 161 | HPR09 | Poll priority bit 09 |
| 162 | LEXIN | External interrupt |
| 163 | HPR10 | Poll priority bit 10 |
| 164 | +5vB | Positive 5 volts DC from battery backup |
| 165 | HPR11 | Poll priority bit 11 |
| 166 | GND | Ground |
| 167 | HPR12 | Poll priority bit 12 |
| 168 | LERRED | Parity bit toggle signal |
| 169 | HPR13 | Poll priority bit 13 |
| 170 | LIORST | I/O reset |
| 171 | HPR15 | Poll priority bit 15 |
| 172 | HPR14 | Poll priority bit 14 |
| 173 | HPR17 | Poll priority bit 17 |
| 174 | HPR16 | Poll priority bit 16 |
| 175 | GND | Ground |
| 176 | HPR18 | Poll priority bit 18 |
| 177 | HPR20 | Poll priority bit 20 |
| 178 | HPR19 | Poll priority bit 19 |
| 179 | HPR22 | Poll priority bit 22 |
| 180 | HPR21 | Poll priority bit 21 |
| 181 | +5vA | Positive 5 volts DC from power supply |
| 182 | +5vA | Positive 5 volts DC from power supply |
| 183 | GND | Ground |
| 184 | GND | Ground |

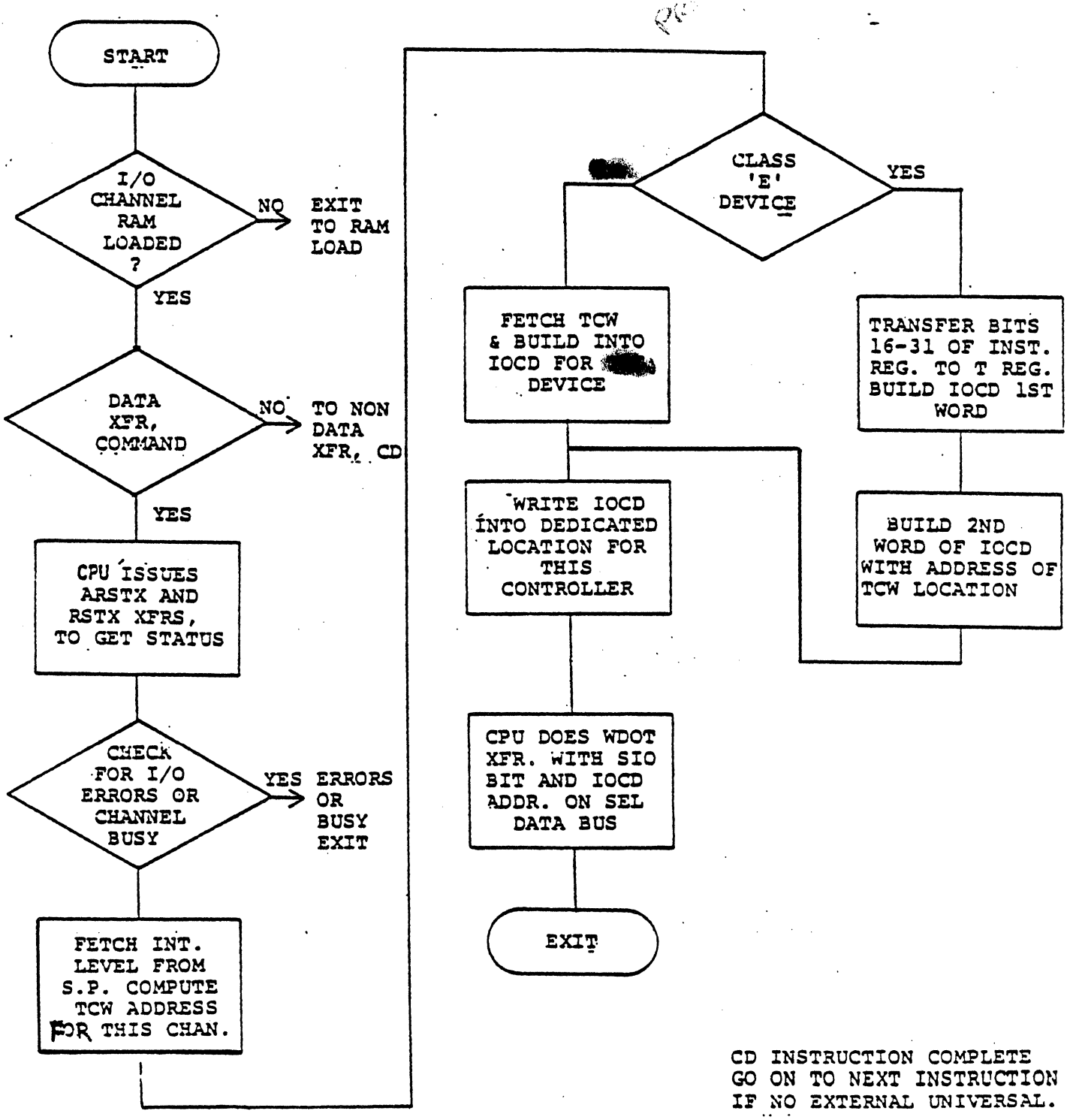
BASIC SEL BUS SEQUENCE EXECUTING A CD




NOTE: Tag Line, Response Lines, Priority Interrupt Lines and Misc. Lines not shown

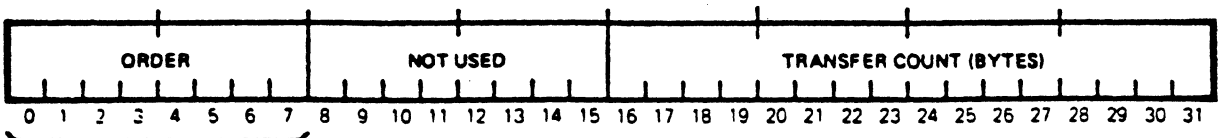
BASIC CD INSTRUCTION FIRMWARE FLOW

DATA TRANSFER COMMAND




CD INSTRUCTION COMPLETE
GO ON TO NEXT INSTRUCTION
IF NO EXTERNAL UNIVERSAL.

**FORMAT A**
**DEVICE CLASS 0, 1, OR 2**
**INPUT/OUTPUT COMMAND DOUBLEWORD - WORD 0**

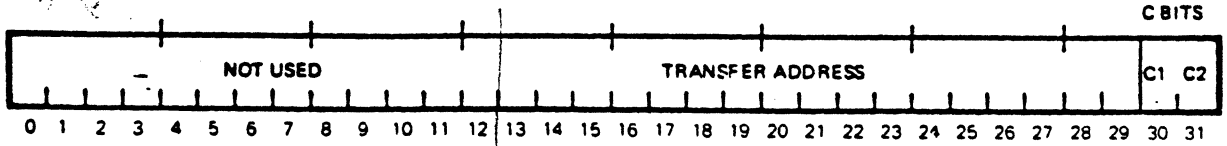



ORDER

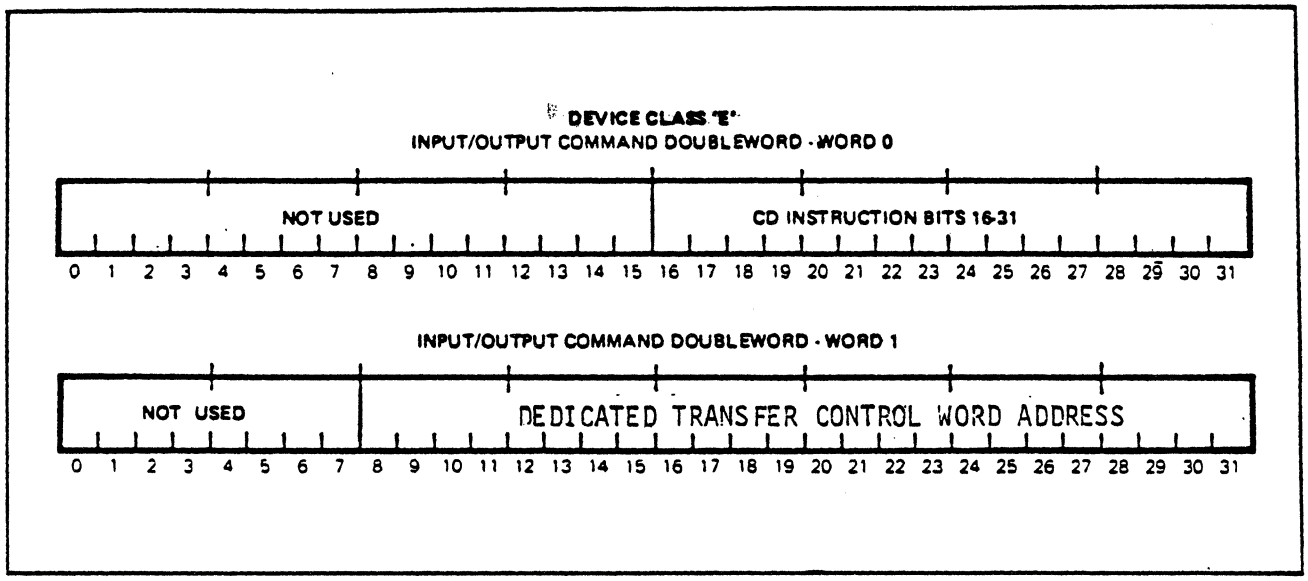
| | IOCD WORD 0 BITS | | | | | | | | FUNCTION |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| DEVICE CLASS 0 (LINE PRINTER) | M | M | M | M | M | 0 | 0 | 1 | BASIC WRITE - CD INITIALIZE DATA OUTPUT |
| | M | M | M | M | M | 0 | 1 | 0 | BASIC READ - CD INITIALIZE DATA INPUT |
| | M | M | M | M | M | 1 | 1 | 0 | CD CONTROL (NON-DATA TRANSFER) |
| | 20 | NU | 21 | 22 | 23 | X | X | X | CD INSTRUCTION BITS TRANSLATED INTO IOCD |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | WRITE (PRINT DATA WITH NO PAPER ADVANCE) |
| | 0 | 0 | 21 | 22 | 23 | 0 | 0 | 1 | ADVANCE PAPER THE NUMBER OF LINES SPECIFIED BY CD BITS 21, 22, AND 23 THEN PRINT DATA |
| | 1 | 0 | 21 | 22 | 23 | 0 | 0 | 1 | ADVANCE PAPER TO FORMAT LOOP COLUMN SPECIFIED BY CD BITS 21, 22, AND 23 THEN PRINT DATA |
| | 0 | 0 | 21 | 22 | 23 | 1 | 1 | 0. | ADVANCE PAPER THE NUMBER OF LINES SPECIFIED BY CD INSTRUCTION BITS 21, 22, AND 23 (NO-PRINT) |
| | 1 | 0 | 21 | 22 | 23 | 1 | 1 | 0 | ADVANCE PAPER TO THE FORMAT LOOP COLUMN SPECIFIED BY CD INSTRUCTION BITS 21, 22, AND 23 (NO-PRINT) |
| DEVICE CLASS 1 (CARD READER) | X | NU | $\overline{20}\cdot\overline{21}$ | NU | 21 | 0 | X | 0 | CD INSTRUCTION BITS TRANSLATED INTO IOCD |
| | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | READ IN FULL ASCII MODE (ONLY USED IN CPU INITIALIZATION-IPL SEQUENCE) |
| | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | READ IN HALF ASCII MODE (TRANSLATE MODE) |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | READ IN AUTOMATIC MODE |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | READ IN BINARY MODE |
| DEVICE CLASS 2 (TELETYPEWRITER) | 17 | 17 | 17 | 17 | 20 | 0 | X | X | CD INSTRUCTION BITS TRANSLATED INTO IOCD |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | WRITE TO PRINTER |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | READ FROM KEYBOARD |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | READ FROM KEYBOARD THEN WRITE TO PRINTER (ECHO MODE) |
| | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | ILLEGAL COMMAND CAUSED BY CD BIT 17 |

KEY:
- 'M' = MODIFY BIT USED TO MODIFY THE BASIC WRITE, READ, OR CD CONTROL IOCD ORDERS.
- 'X' = BITS AVAILABLE FOR DEFINING THE BASIC ORDER FUNCTION (WRITE, READ OR CD CONTROL)
- 17,20,21,22, & 23 = CD INSTRUCTION BITS THAT ARE TRANSLATED INTO IOCD ORDER 'M' BITS.
- $\overline{20}\cdot\overline{21}$ = FALSE CONDITION OF CD INSTRUCTION BITS 20 AND 21 ARE REQUIRED TO GENERATE THIS BIT IN THE IOCD ORDER.
- NU = BIT NOT USED.

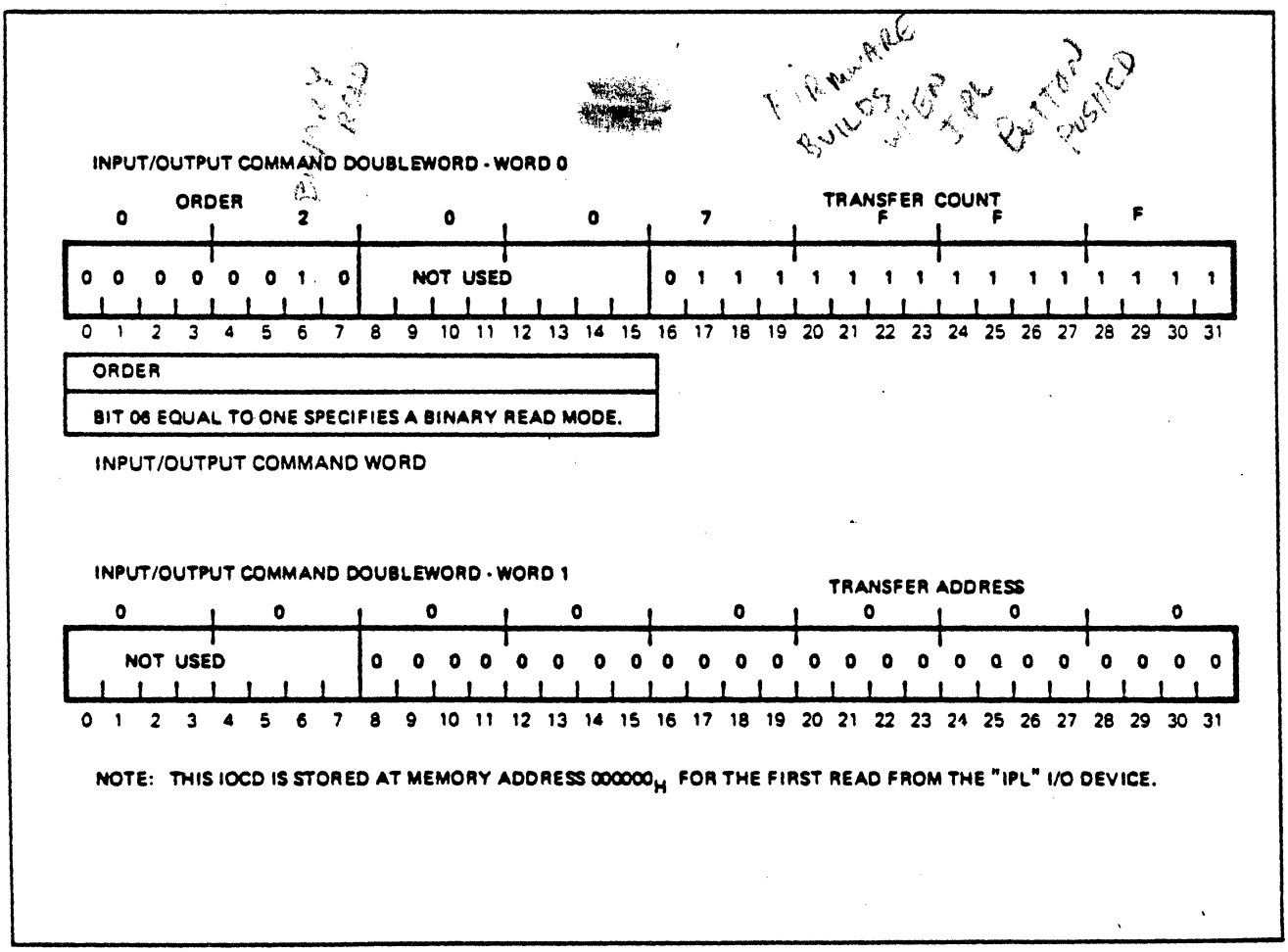
**INPUT/OUTPUT COMMAND DOUBLEWORD - WORD 1**




IOCD Format - Class 0, 1, and 2 Devices




IOCD Format - Class E Devices



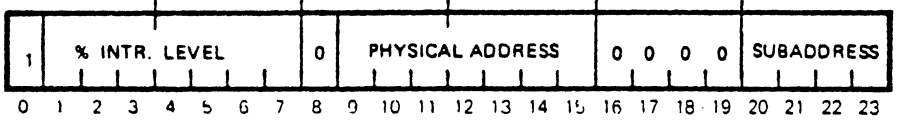

NOTE: THIS IOCD IS STORED AT MEMORY ADDRESS $000000_H$ FOR THE FIRST READ FROM THE "IPL" I/O DEVICE.

IOCD Format - IPL Device

SEL BUS FORMATS

## AICT/ICT SEL BUS TRANSFER

DESTINATION BUS

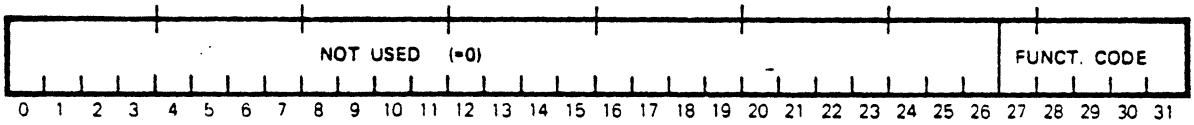



| INTERRUPT LEVEL |
|---|
| BITS 01-07 PROVIDE THE ONES COMPLEMENT OF THE INTERRUPT LEVEL TO BE CONTROLLED. |

| PHYSICAL ADDRESS |
|---|
| BITS 09-15 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER OR RTOM TO WHICH THE INTERRUPT LEVEL IS ASSOCIATED. |

| SUBADDRESS |
|---|
| BITS 20-23 PROVIDE THE I/O CONTROLLER OR RTOM SUBADDRESS WITH WHICH THE INTERRUPT LEVEL IS ASSOCIATED. |

DATA BUS




| FUNCTION CODE | | | | | |
|---|---|---|---|---|---|
| BITS | | | | | DEFINITION |
| 27 | 28 | 29 | 30 | 31 | |
| 0 | 0 | 0 | 0 | 1 | ENABLE INTERRUPT |
| 0 | 0 | 0 | 1 | 0 | DISABLE INTERRUPT |
| 0 | 0 | 1 | 0 | 0 | ACTIVATE INTERRUPT |
| 0 | 1 | 0 | 0 | 0 | DEACTIVATE INTERRUPT |
| 1 | 0 | 0 | 0 | 0 | REQUEST INTERRUPT |

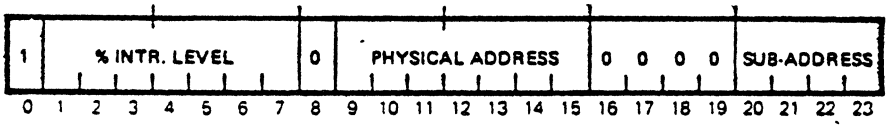
NOTES:
1. THE INTERRUPT CONTROL TRANSFER (AICT OR ICT) IS SENT FROM THE CPU TO THE I/O CONTROLLER OR RTOM DURING INTERRUPT CONTROL INSTRUCTIONS, BRANCH AND RESET INTERRUPT INSTRUCTIONS, AND INTERRUPT SEQUENCES.
2. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY THE TRANSFER AS AN AICT OR ICT.
3. THE I/O CONTROLLER OR RTOM RESPONDS TO THE AICT AND ICT WITH A DATA RETURN TRANSFER (DRT). THE CONTENTS OF THE DRT IS NOT USED.

NOTES:
1. THE INTERRUPT CONTROL TRANSFER (AICT OR ICT) IS SENT FROM THE CPU TO THE I/O CONTROLLER OR RTOM DURING INTERRUPT CONTROL INSTRUCTIONS, BRANCH AND RESET INTERRUPT INSTRUCTIONS, AND INTERRUPT SEQUENCES.
2. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY THE TRANSFER AS AN AICT OR ICT.
3. THE I/O CONTROLLER OR RTOM RESPONDS TO THE AICT AND ICT WITH A DATA RETURN TRANSFER (DRT). THE CONTENTS OF THE DRT IS NOT USED.

Interrupt Control Transfers (AICT or ICT)

ARSTX/RSTX SEL BUS TRANSFER

DESTINATION BUS

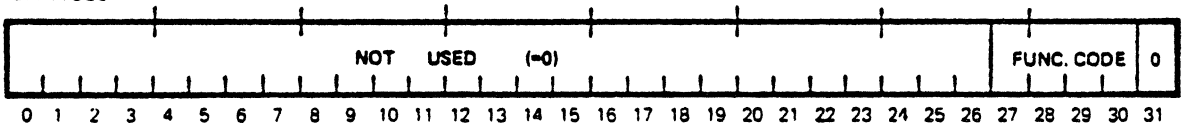



| INTERRUPT LEVEL |
|---|
| BITS 01-07 PROVIDE THE ONES COMPLEMENT OF THE I/O CONTROLLER INTERRUPT LEVEL |

| PHYSICAL ADDRESS |
|---|
| BITS 09-15 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER |

| SUBADDRESS |
|---|
| BITS 20-23 PROVIDE THE I/O CONTROLLER DEVICE SUBADDRESS |

DATA BUS



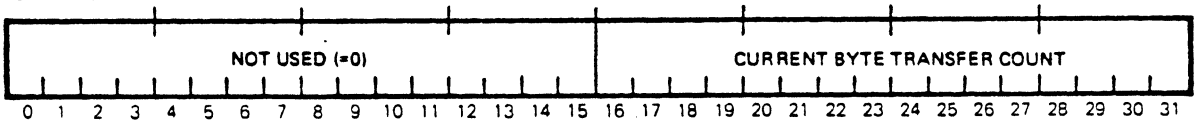

| FUNCTION CODE | | | | |
|---|---|---|---|---|
| BITS | | | | DEFINITION |
| 27 | 28 | 29 | 30 | |
| 0 | 0 | 0 | 1 | TRANSFER CURRENT WORD ADDRESS (TCA) |
| 0 | 0 | 1 | 0 | DEVICE STATUS (TD2000 STATUS) |
| 0 | 1 | 0 | 0 | CONTROLLER STATUS (TD8000 AND 4000 STATUS) |
| 1 | 0 | 0 | 0 | ACKNOWLEDGE INTERRUPT (ACTIVATE INTERRUPT LEVEL) |

NOTES:
1. THE READ STATUS TRANSFER (ARSTX OR RSTX) IS SENT FROM THE CPU TO THE I/O CONTROLLER DURING ANY CPU-I/O COMMUNICATION SEQUENCE.
2. THE I/O CONTROLLER RESPONDS TO THE ARSTX AND RSTX SEQUENCE WITH A DRT THAT CONTAINS THE REQUESTED STATUS. REFER TO FIGURE 5-57 FOR THE VARIOUS DRT FORMATS.
3. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY THE TRANSFER AS A RSTX OR RSTX.

Read Status Transfer (ARSTX and RSTX)

**DRT SEL BUS TRANSFER**
**FORMAT F**

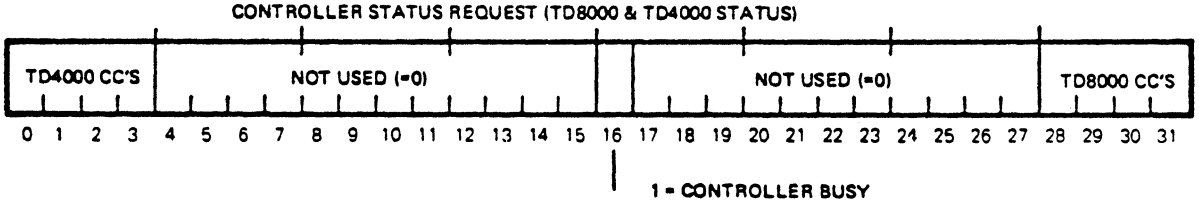
DATA BUS FOR 'TLC' CONTROLLER RESPONSE TO AN ARSTX/RSTX TRANSFER CURRENT WORD ADDRESS (TCWA) REQUEST




| CURRENT BYTE TRANSFER COUNT |
|---|
| BITS 16-31 PROVIDE THE CURRENT BYTE TRANSFER COUNT. |

**FORMAT G**

DATA BUS FOR STANDARD I/O CONTROLLER RESPONSE TO A ARSTX/RSTX CONTROLLER STATUS REQUEST (TD8000 & TD4000 STATUS)

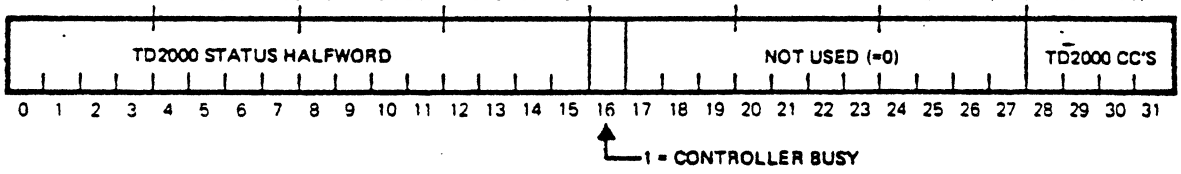



| TD4000 CONDITION CODES |
|---|
| BITS 00, 01, 02, AND 03 PROVIDE THE TD4000 CONDITION CODES 1, 2, 3, AND 4, RESPECTIVELY. REFER TO FIGURE 4-3 FOR CONDITION CODE BIT DEFINITION. |

Format - Data Return Transfer (DRT) (Sheet 3 of 4)

DRT SEL BUS TRANSFER
FORMAT H

DATA BUS FOR STANDARD I/O CONTROLLER RESPONSE TO AN ARSTX/RSTX DEVICE STATUS REQUEST (TD2000 STATUS)

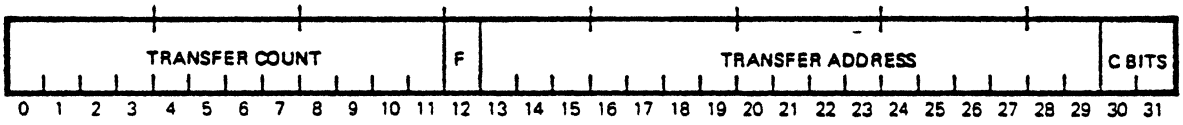



| TD2000 STATUS HALFWORD |
|---|
| BITS 00-15 PROVIDE THE TD2000 STATUS HALFWORD. REFER TO THE DEVICE CONTROLLER TECHNICAL MANUAL FOR BIT DEFINITIONS. |

| TD2000 CONDITION CODES |
|---|
| BITS 28, 29, 30, AND 31 PROVIDE THE TD2000 CONDITION CODES 1, 2, 3, AND 4 RESPECTIVELY. REFER TO FIGURE 4-3 FOR CONDITION CODE BIT DEFINITION. |

FORMAT I

DATA BUS FOR STANDARD I/O CONTROLLER RESPONSE TO AN ARSTX/RSTX TRANSFER CURRENT WORD ADDRESS (TCWA) REQUEST

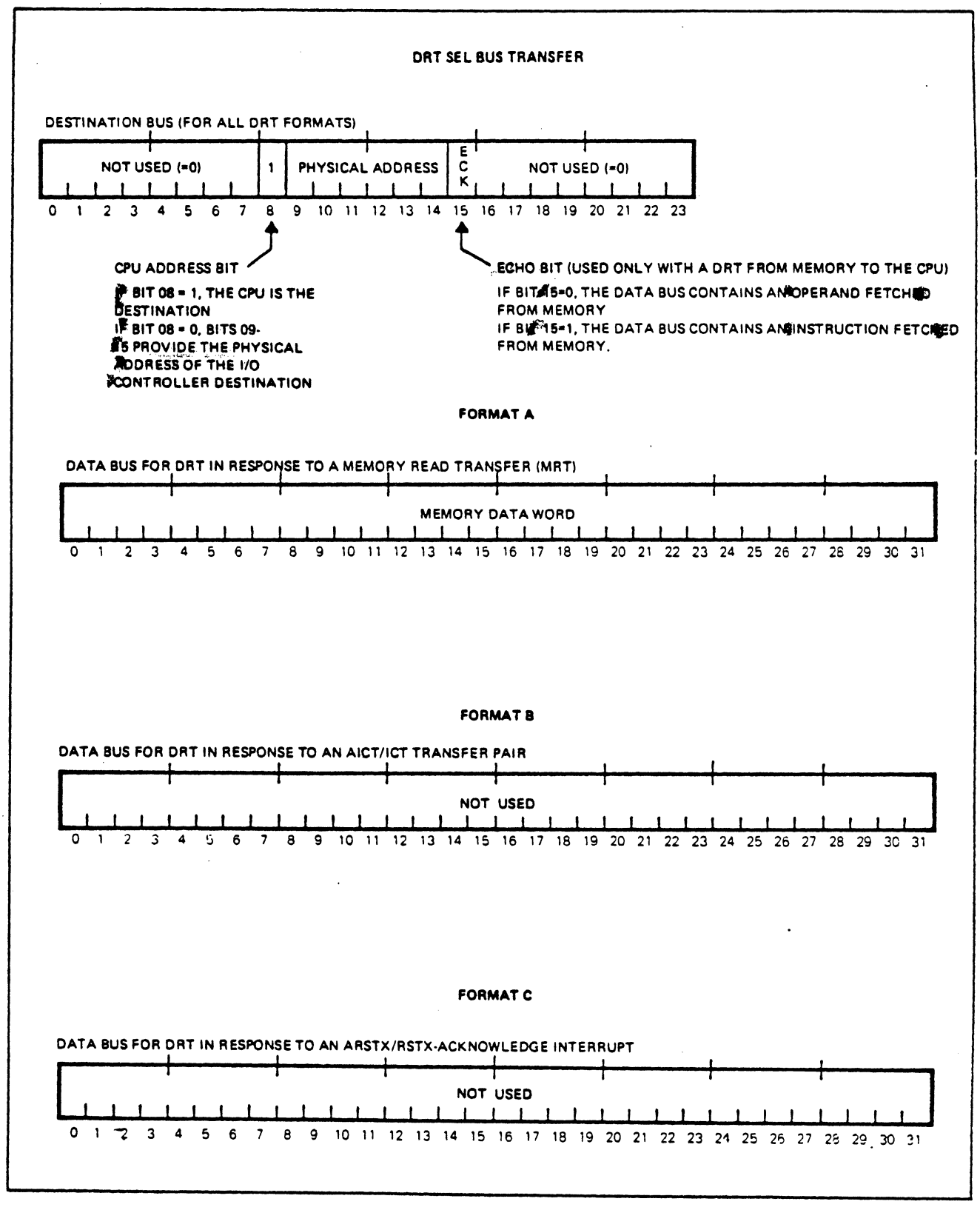



| TRANSFER COUNT |
|---|
| BITS 00-11 PROVIDE THE CURRENT TRANSFER COUNT. THE COUNT CAN BE IN WORDS, HALFWORDS, OR BYTES, ACCORDING TO THE 'F' AND 'C' BITS. |

| TRANSFER ADDRESS |
|---|
| BITS 13-31 PROVIDE THE CURRENT MEMORY DATA ADDRESS. THE ADDRESS CAN BE A WORD, HALFWORD, OR BYTE ADDRESS, ACCORDING TO THE 'F' AND 'C' BITS. REFER TO FIGURE 5-62 FOR THE DEFINITION OF THE 'F' AND 'C' BITS. |

NOTES:
1. A DATA RETURN TRANSFER (DRT) IS SENT FROM MEMORY OR A I/O CONTROLLER TO THE CPU UNDER THE FOLLOWING CONDITIONS:
   A. FROM THE MEMORY TO THE CPU IN RESPONSE TO A MEMORY READ TRANSFER (MRT). REFER TO FIGURE 5-59 FOR A MRT FORMAT.
   B. FROM THE I/O CONTROLLER TO THE CPU IN RESPONSE TO EITHER AN AICT/ICT OR AN ARSTX/RSTX TRANSFER PAIR. REFER TO FIGURE 5-55 FOR THE AICT/ICT FORMAT AND TO FIGURE 5-56 FOR THE ARSTX/RSTX FORMAT.
2. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY THE TRANSFER AS A DRT.

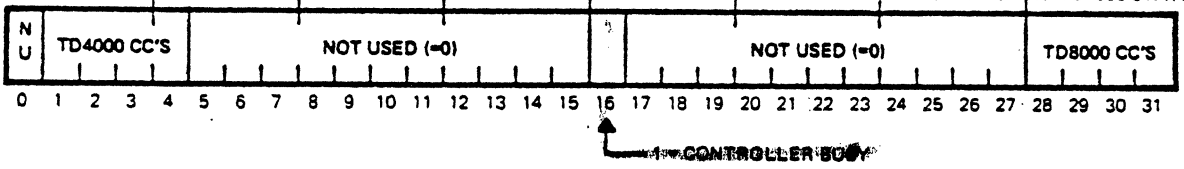
Format - Data Return Transfer (DRT) (Sheet 4 of 4)




Format - Data Return Transfer (DRT) (Sheet 1 of 4)

**DRT SEL BUS TRANSFER**
**FORMAT D**

DATA BUS FOR 'TLC' CONTROLLER RESPONSE TO A ARSTX/RSTX CONTROLLER STATUS REQUEST (TD8000 AND TD4000 STATUS)

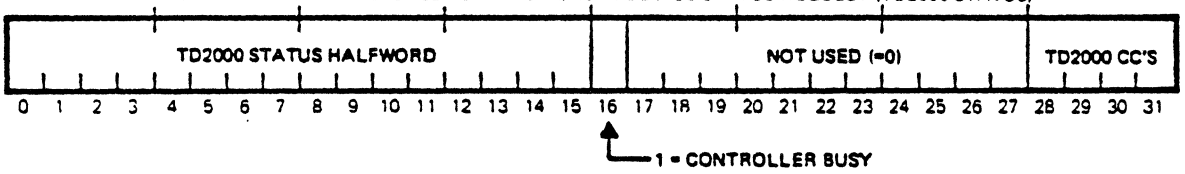



| TD4000 CONDITION CODES |
|---|
| BITS 01, 02, 03 AND 04 PROVIDE THE TD4000 CONDITION CODES 1, 2, 3, AND 4, RESPECTIVELY. REFER TO FIGURE 4-3 FOR CONDITION CODE BIT DEFINITION. |

| TD8000 CONDITION CODES |
|---|
| BITS 28, 29, 30, AND 31 PROVIDE THE TD8000 CONDITION CODES 1, 2, 3, AND 4, RESPECTIVELY. REFER TO FIGURE 4-3 FOR CONDITION CODE BIT DEFINITION. |

**FORMAT E**

DATA BUS FOR 'TLC' CONTROLLER RESPONSE TO A ARSTX/RSTX DEVICE STATUS REQUEST (TD2000 STATUS)



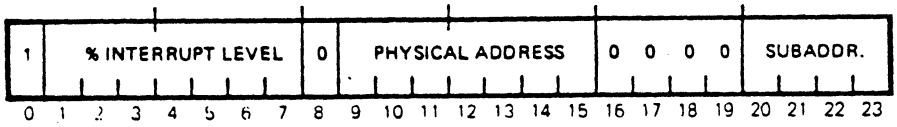

| TD2000 STATUS HALFWORD |
|---|
| BITS 00-15 PROVIDE THE TD2000 STATUS HALFWORD. REFER TO THE 'TLC' TECHNICAL MANUAL FOR BIT DEFINITION. |

| TD2000 CONDITION CODES |
|---|
| BITS 28, 29, 30, AND 31 PROVIDE THE TD2000 CONDITION CODES 1, 2, 3, AND 4, RESPECTIVELY. REFER TO FIGURE 4-3 FOR CONDITION CODE BIT DEFINITION. |

Format - Data Return Transfer (DRT) (Sheet 2 of 4)

## WDOT SEL BUS TRANSFER

DESTINATION BUS (FOR ALL FORMATS)



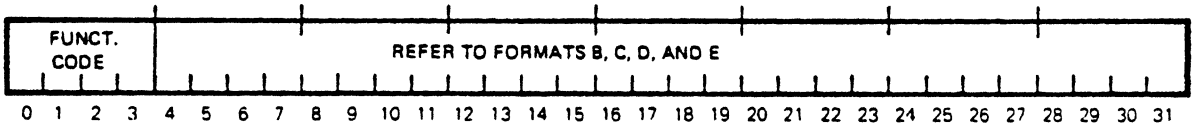

| INTERRUPT LEVEL |
|---|
| BITS 01-07 PROVIDE THE ONES COMPLEMENT OF THE I/O CONTROLLER INTERRUPT LEVEL. |

| PHYSICAL ADDRESS |
|---|
| BITS 09-15 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER. |

| SUBADDRESS |
|---|
| BITS 20-23 PROVIDE THE I/O CONTROLLER DEVICE SUBADDRESS. |

## FORMAT A

DATA BUS FOR WDOT TRANSFER (GENERAL FORMAT)



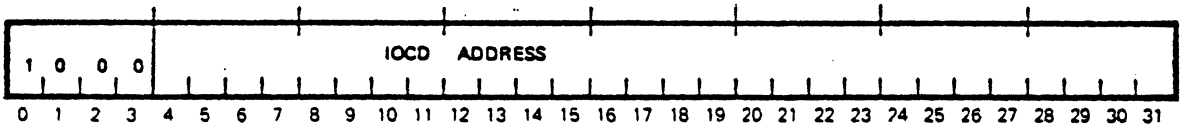

| FUNCTION CODE BITS | | | | |
|---|---|---|---|---|
| 00 | 01 | 02 | 03 | DEFINITION |
| 1 | 0 | 0 | 0 | START I/O. REFER TO FORMAT B FOR THE CONTENTS OF BITS 04-31. |
| 0 | 1 | 0 | 0 | LOAD RAM. REFER TO FORMAT C FOR THE CONTENTS OF BITS 04-31. |
| 0 | 0 | 1 | 0 | HALT I/O. SEE NOTE 2. |
| 1 | 0 | 0 | 1 | INITIAL PROGRAM LOAD AND START I/O. REFER TO FORMAT E FOR THE CONTENTS OF BITS 04-31. |

Format - Write Data or Order Transfer (WDOT) (Sheet 1 of 3)

WDOT SEL BUS TRANSFER

FORMAT B

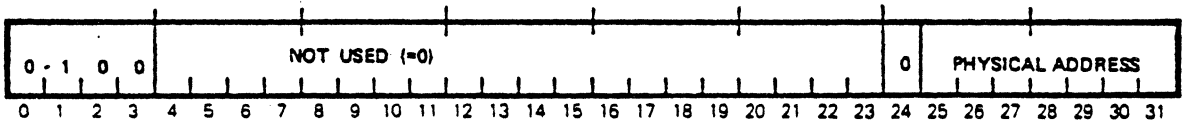
DATA BUS FOR WDOT - START I/O




| IOCD ADDRESS |
| --- |
| BITS 04-31 PROVIDE THE MEMORY ADDRESS OF THE INPUT/OUTPUT COMMAND DOUBLEWORD (IOCD). |

FORMAT C

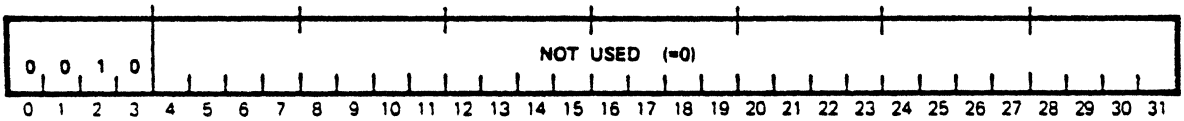
DATA BUS FOR WDOT - LOAD RAM




| PHYSICAL ADDRESS |
| --- |
| BITS 25-31 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER. |

FORMAT D

DATA BUS FOR WDOT - HALT I/O

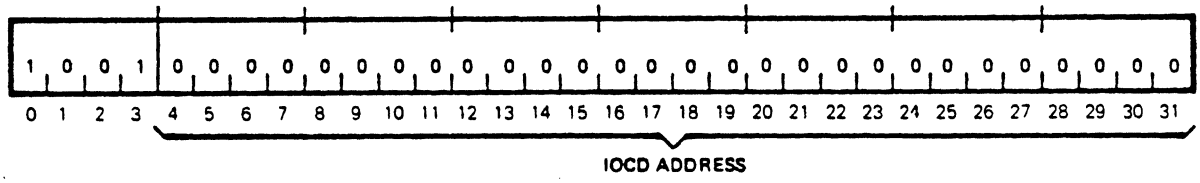



Format - Write Data or Order Transfer (WDOT) (Sheet 2 of 3)

WDOT SEL BUS TRANSFER

FORMAT E

DATA BUS FOR WDOT - INITIAL PROGRAM LOAD START I/O




| IOCD ADDRESS |
|---|
| BITS 03-31 PROVIDE THE MEMORY ADDRESS OF THE INPUT/OUTPUT COMMAND DOUBLEWORD (IOCD), WHICH MUST BE 00 00 00 $00_H$ DURING IPL. |

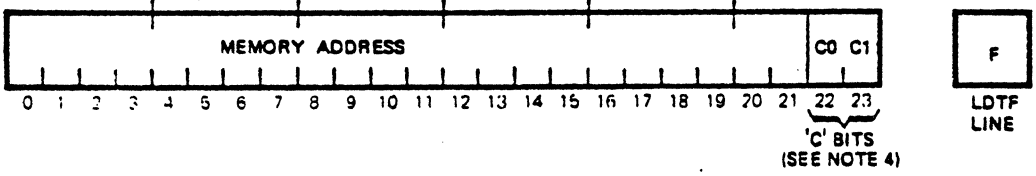
NOTES:

1. THE WRITE DATA OR ORDER TRANSFER (WDOT) IS SENT FROM THE CPU TO THE I/O CONTROLLER OR RTOM.
2. THE WDOT IS NORMALLY PRECEDED BY AN ARSTX, RSTX, AND DRT SEQUENCE TO DETERMINE THE AVAILABILITY AND OPERABILITY OF THE I/O CONTROLLER OR RTOM. THE WDOT-HALT I/O IS NOT PRECEDED BY THE ARSTX, RSTX, AND DRT SEQUENCE, SINCE THE PURPOSE OF THE WDOT HALT I/O IS TO CLEAR A BUSY I/O CONTROLLER.
3. THE I/O CONTROLLER OR RTOM DOES NOT EXECUTE A BUS TRANSFER RESPONSE TO THE WDOT.
4. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY THE TRANSFER AS A WDOT.

Format - Write Data or Order Transfer (WDOT) (Sheet 3 of 3)

MRT SEL BUS TRANSFER

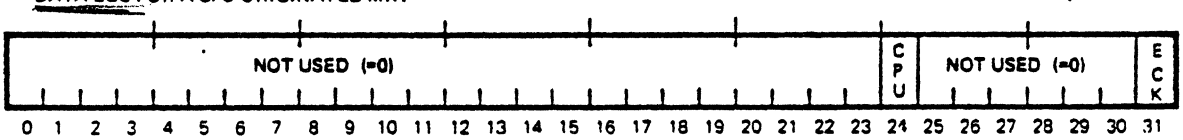
DESTINATION BUS (FOR ALL FORMATS)




| MEMORY ADDRESS |
| --- |
| BITS 00-23 PROVIDE THE MEMORY ADDRESS OF THE LOCATION TO BE READ. |

FORMAT A

DATA BUS FOR A CPU ORIGINATED MRT

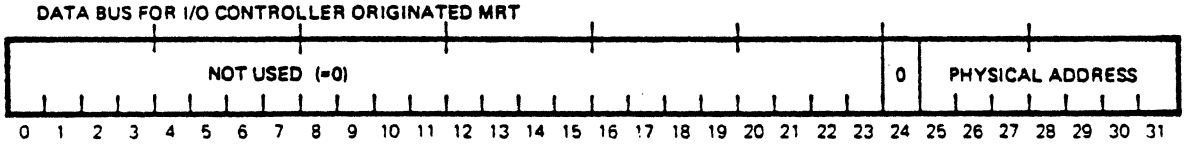



| CPU |
| --- |
| BIT 24 EQUAL TO ONE SPECIFIES THAT THE CPU IS THE SOURCE OF THE MRT AND THAT THE DATA READ FROM THE ADDRESSED LOCATION IS TO BE RETURNED TO THE CPU. |

| ECHO |
| --- |
| BIT 31 EQUAL TO ZERO SPECIFIES THAT AN OPERAND IS TO BE READ FROM MEMORY; BIT 31 EQUAL TO ONE SPECIFIES THAT AN INSTRUCTION IS TO BE READ FROM MEMORY. |

FORMAT B

DATA BUS FOR I/O CONTROLLER ORIGINATED MRT




| PHYSICAL ADDRESS |
| --- |
| BITS 25 THRU 31 PROVIDE THE PHYSICAL ADDRESS OF THE I/O CONTROLLER THAT ORIGINATED THE MRT AND SPECIFIES THAT THE DATA READ FROM THE ADDRESSED MEMORY LOCATION IS TO BE SENT TO THIS I/O CONTROLLER ADDRESS. |

Format - Memory Read Transfer (MRT) (Sheet 1 of 2)

NOTES:

1. THE MEMORY READ TRANSFER (MRT) CAN BE ORIGINATED BY EITHER AN I/O CONTROLLER OR THE CPU. THE DESTINATION OF THE TRANSFER IS THE MEMORY BUS CONTROLLER (MBC).

2. THE MEMORY BUS CONTROLLER RESPONDS TO THE MRT WITH A DATA RETURN TRANSFER (DRT) CONTAINING THE CONTENTS OF THE MEMORY LOCATION ADDRESSED BY THE MRT. THE DESTINATION OF THE DRT IS THE CPU OR IOM THAT ORIGINATED THE MRT.

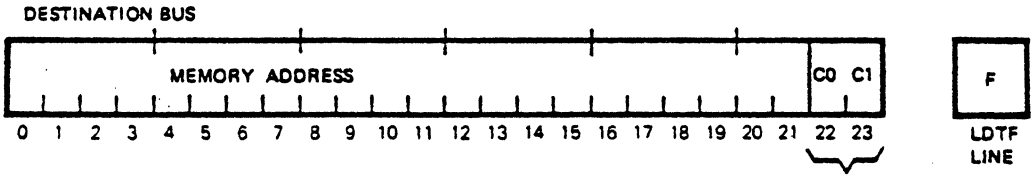
3. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY A TRANSFER AS AN MRT.

4. THE MRT CAN BE USED TO SPECIFY A WORD, HALFWORD, OR BYTE READ FUNCTION. THE DATA READ FROM MEMORY IS RETURNED ON THE DATA BUS, RIGHT JUSTIFIED IN A DRT TRANSFER. THE 'F' BIT (THE TAG BUS LDTF SIGNAL) AND THE DESTINATION BUS 'C' BITS ARE USED TO SPECIFY ANY OF THESE MODES AS FOLLOWS:

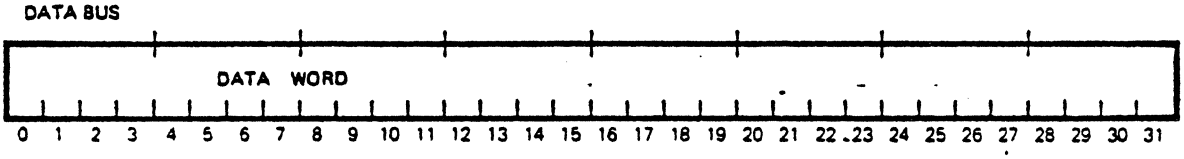
| F BITS (LDTF SIGNAL) | C BITS DESTINATION BUS BITS | | TRANSFER FUNCTION |
|---|---|---|---|
| | 22 | 23 | |
| 0 (HIGH) | 0 | 0 | WORD TRANSFER |
| 0 (HIGH) | 0 | 1 | HALFWORD TRANSFER (LEFT HALFWORD) |
| 0 (HIGH) | 1 | 1 | HALFWORD TRANSFER (RIGHT HALFWORD) |
| 1 (LOW) | 0 | 0 | BYTE TRANSFER TO BYTE 0 (BITS 00-07) |
| 1 (LOW) | 0 | 1 | BYTE TRANSFER TO BYTE 1 (BITS 08-15) |
| 1 (LOW) | 1 | 0 | BYTE TRANSFER TO BYTE 2 (BITS 16-23) |
| 1 (LOW) | 1 | 1 | BYTE TRANSFER TO BYTE 3 (BITS 24-31) |

Format - Memory Read Transfer (MRT) (Sheet 2 of 2)

MWT SEL BUS TRANSFER




| MEMORY ADDRESS |
|---|
| BITS 00-23 PROVIDE THE MEMORY ADDRESS FOR THE WRITE MEMORY FUNCTION. |




| DATA WORD |
|---|
| BITS 00-31 PROVIDE THE DATA WORD TO BE STORED AT THE ADDRESS SPECIFIED BY THE DESTINATION BUS. |

NOTES:

1. THE MEMORY WRITE TRANSFER (MWT) CAN BE ORIGINATED BY EITHER AN I/O CONTROLLER OR THE CPU. THE DESTINATION OF THE TRANSFER IS THE MEMORY BUS CONTROLLER.

2. REFER TO TABLE 5-12 FOR THE SEL BUS TAG SIGNAL CONFIGURATIONS THAT IDENTIFY A TRANSFER AS A MRT.

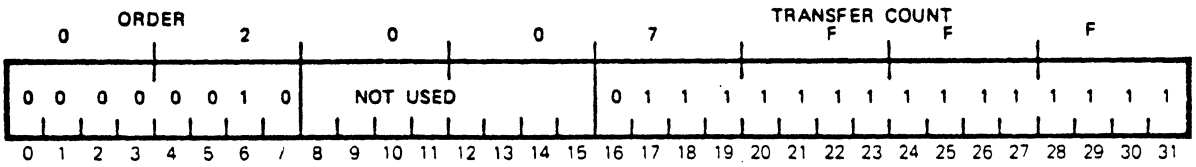
3. THE MWT CAN BE USED TO SPECIFY A WORD, HALFWORD, OR BYTE WRITE FUNCTION. IN ANY OF THESE CASES, THE DATA TO BE STORED IN MEMORY MUST BE RIGHT-JUSTIFIED ON THE DATA BUS AND THE SEL BUS TAG BUS SIGNAL 'LDTF' ('F' BIT) AND THE DESTINATION BUS 'C' BITS ARE USED AS FOLLOWS:

| F | 22 | 23 | |
|---|---|---|---|
| 0 | 0 | 0 | WORD |
| 0 | 0 | 1 | LHW |
| 0 | 1 | 1 | RHW |
| 1 | 0 | 0 | BYTE 0 |
| 1 | 0 | 1 | BYTE 1 |
| 1 | 1 | 0 | BYTE 2 |
| 1 | 1 | 1 | BYTE 3 |

Format - Memory Write Transfer (MWT)

IPL

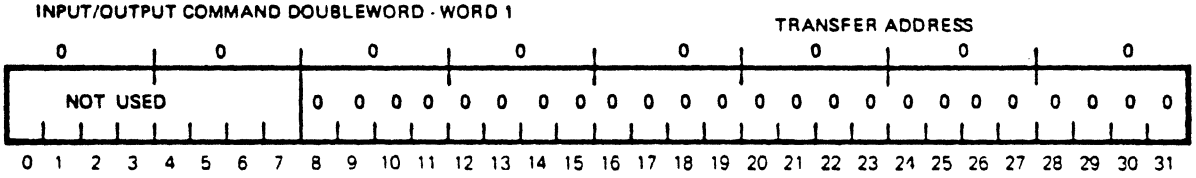
INPUT/OUTPUT COMMAND DOUBLEWORD - WORD 0




| ORDER |
|---|
| BIT 06 EQUAL TO ONE SPECIFIES A BINARY READ MODE. |

INPUT/OUTPUT COMMAND W

INPUT/OUTPUT COMMAND DOUBLEWORD - WORD 1




NOTE: THIS IOCD IS STORED AT MEMORY ADDRESS $000000_H$ FOR THE FIRST READ FROM THE "IPL" I/O DEVICE.

NOTES:
1. THIS IOCD IS STORED AT MEMORY ADDRESS $000000_H$ FOR THE FIRST READ FROM THE "IPL" I/O DEVICE.
2. THE ORDER BYTE SPECIFIES A BINARY READ FROM THE IPL DEVICE.

Format - IPL Basic Input/Output Command Doubleword (IOCD)

WORKSESSION

Central Processing Unit

Objectives:

After completing the referenced reading sections of the technical manual, completing general information questions of this section, and completing the data structure block diagram of the CPU; the student will be capable of identifying the following, concerning the 32/70 Series Central Processing Unit:

1. The fundamental architectural characteristics of the CPU.
2. The component parts of the CPU data structure.
3. Relationship and purpose of the CPU in a total system.

Referenced Reading:

1. 32/70 Series Computer, Technical Manual;

   A. Functional Description; Pgs. 1-3 thru 1-6
   B. CPU Modules Interconnection; Pg. 1-10

Course # 340 - 32/75 Architecture

CPU - WORKSESSION (GENERAL QUESTIONS)

Directions: Answer the following statements as true or false (true = 1, false = 0). Also, correct all false statements to make them true.

1 1. One instruction can be decoded while another is being fetched from memory.

1 2. High-speed instruction decoding is accomplished by using ROM's (read-only-memories).

1 3. The CPU is on three plug-in circuit boards. One board is for the micro control unit and the other two boards are for micro-arithmetic logic units.

0 4. A 20-bit address field is provided for directly addressing 512K bytes of memory in memory reference instructions. Only bits, bytes, halfwords, and words are directly addressable.

1 5. When a halfword instruction is followed by a fullword instruction, a no operation (NOP) is placed in the second half of the halfword instruction.

1 6. During the execution of Input/Output operations, the CPU uses the highest SEL Bus priority.

1 7. The CPU firmware, which is implanted in the control memory, uses a 32-bit elementary operation (EO) micro-command word format.

WORKSESSION

## SEL Bus

It is to your advantage to learn all you can about the SEL Bus since it is the communications link between all elements of the system.

Objectives:

This worksession will help you to understand the physical characteristics of the SEL Bus. The SEL Bus protocol will be better understood if all questions are answered correctly. (If you cannot answer a question, please ask the instructor to reword the question or to explain it to you.)

Reference Reading:

1. Technical Manual, SYSTEMS 32/70 Series Computer

Overview:

1. Read the Series 32 Computer Tech. Manual. Pgs. 6-1 thru 6-32

Worksession

## SEL Bus

1. What is the transfer rate on the SEL Bus? 1 Son Src

2. The SEL Bus is physically a part of the Logic Chassis P1B

3. List the types of transfers that occur on the SEL Bus between the CPU, IOM's, Memory and RTOM's.

   1. AICT ✓
   2. ARSTX ✓ from CPU to Iom
   3. DRT ✓ Data Return Transfer
   4. WDOT ✓
   5. MRT ✓
   6. MWT ✓
   8. ICT ✓
   9. RSTX ✓
   10. MRLT ✓
   11. ET ✓
   12. MWUT ✓

4. During which type of transfers is the CPU's SEL Bus Priority Ø?

   1. AICT
   2. ICT
   3. ARSTX
   4. RSTX
   5. WDOT

5. What Sequence of transfers takes place on the SEL Bus between the CPU and IOM when executing a Command Device Instruction?

   1. ARSTX
   2. READY
   3. RSTX
   4. DRT
   5. WDOT

6. During which of the above sequences is the IOM's Priority Ø? DRT

7. What sequence of transfers takes place on the SEL Bus when the CPU is executing an Enable Interrupt Instruction to the IOM or RTOM?

   1. AICT
   2. READY
   3. ICT
   4. DRT

8. Which SEL Bus lines are the IOM's physical address transmitted on by the CPU?

   Destination Bus 24 lines +F 6,4

   Pins 59, 60, 61, 62, 64, 65, 66 on SELBUS P1B

## SEL Bus

9. What bit on the Destination Bus tells the CPU that the DRT from the MBC is for an Instruction or an Operand?

   Bit 15 Echo bit

10. Most signals on the SEL Bus are driven low true. What group of signals are driven high true? Priority Lines 1-22

11. When an IOM is removed from the logic chassis what must be done to the SEL Bus terminator before the system can be used again? A jumper installed for Xmit Priority

12. What indication is observed on the System Control Panel when a Test Device Instruction is executed to a non-present IOM?

    For PSW $CC_1$-$CC_4$ = F

13. The logic chassis backplane is a copper etched board divided into three sigments. The middle section (184 lines) is the SEL Bus. The outer 2 segments are how many lines? Are they interconnected? What are they used for?

    A. 50 lines

    B. No

    C. I/O connections

14. What is the difference between SEL Bus Priority and Interrupt Priority?

    Interrupt Priority is for the RTOM (Software handshaking)
    SEL Bus Priority is for Terms/Controllers (hardware control)

15. When the SEL Bus response line "READY" is generated, what does it indicate?

    The device being requested to send status is ready to "Send Status" No previous instruction, not waiting for Data/Info

16. How many lines on the SEL Bus are associated with Priority Interrupts? 3 lines

17. In reference to the preceding question, name the lines and explain what functions they perform.

    Upon [illegible] to [illegible] serial polling

    [illegible] provides signal to CPU to tell CPU there is a [illegible] code to be examined

    EOP Enables [illegible] Timing

# DAY 8

# SECTION 8

# MEMORY SUBSYSTEM

# MAPPING

# MEMORY SUBSYSTEM

CONTROLLED BY MEMORY BUS CONTROLLER (MBC)

MBC INTERFACES MEMORY MODULE TO SEL BUS
CHECKS AND GENERATES PARITY
MBC CONTROLS UP TO 16 MEMORY MODULES

MBC CAN ADDRESS:

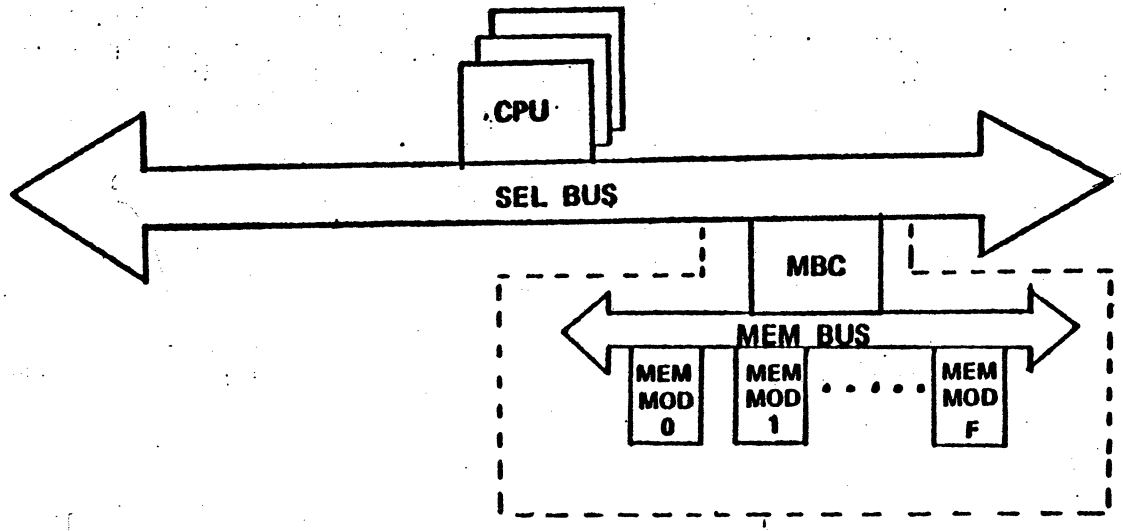
BIT: SMALLEST MEMORY VALUE ADDRESS BY INSTRUCTION
BYTE: 8 BITS OR 2 HEX CHARACTERS
HALFWORD: 16 BITS (LEFT OR RIGHT)
FULLWORD: 32 BITS - 2 HW - 4 BYTES/8 HEX CHARACTERS
DOUBLEWORD: 64 BITS (2 WORD FETCH)

MEMORY MODULES:

8/16 KW (32/64 KB) CORE MODULES, BYTE PARITY

32/64 KW (128/256 KB) MOS MODULES, WORD ECC

# MEMORY SUBSYSTEM




CORE MBC'S

2150 WIRE/WRAP
2162 COPPER - 4 PORT, 600/900NS
2164 WIRE/WRAP - 2 PORT, 600NS

MOS MBC'S

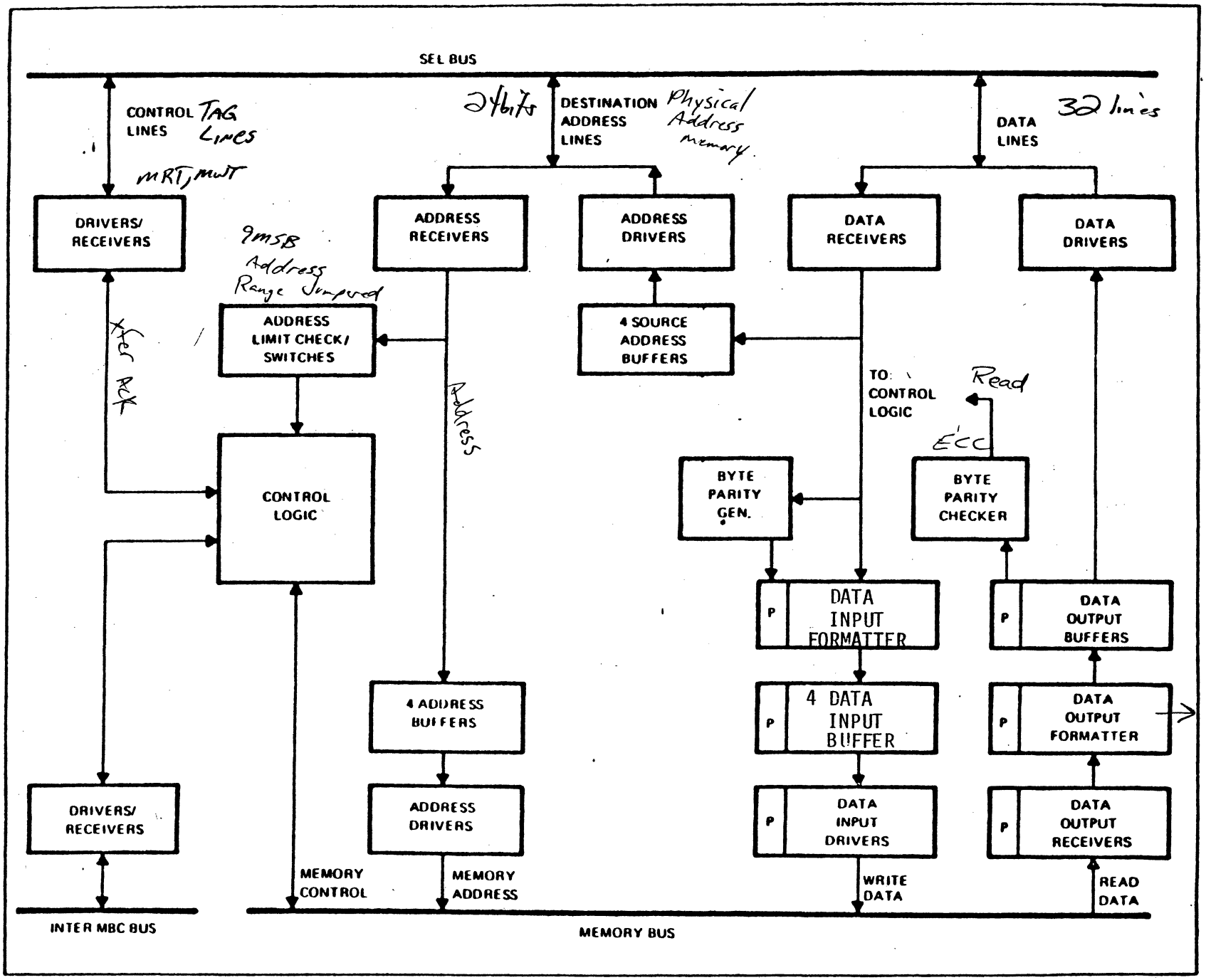
2377 32KW/900NS
2382 COPPER

CORE MEMORY MODULES

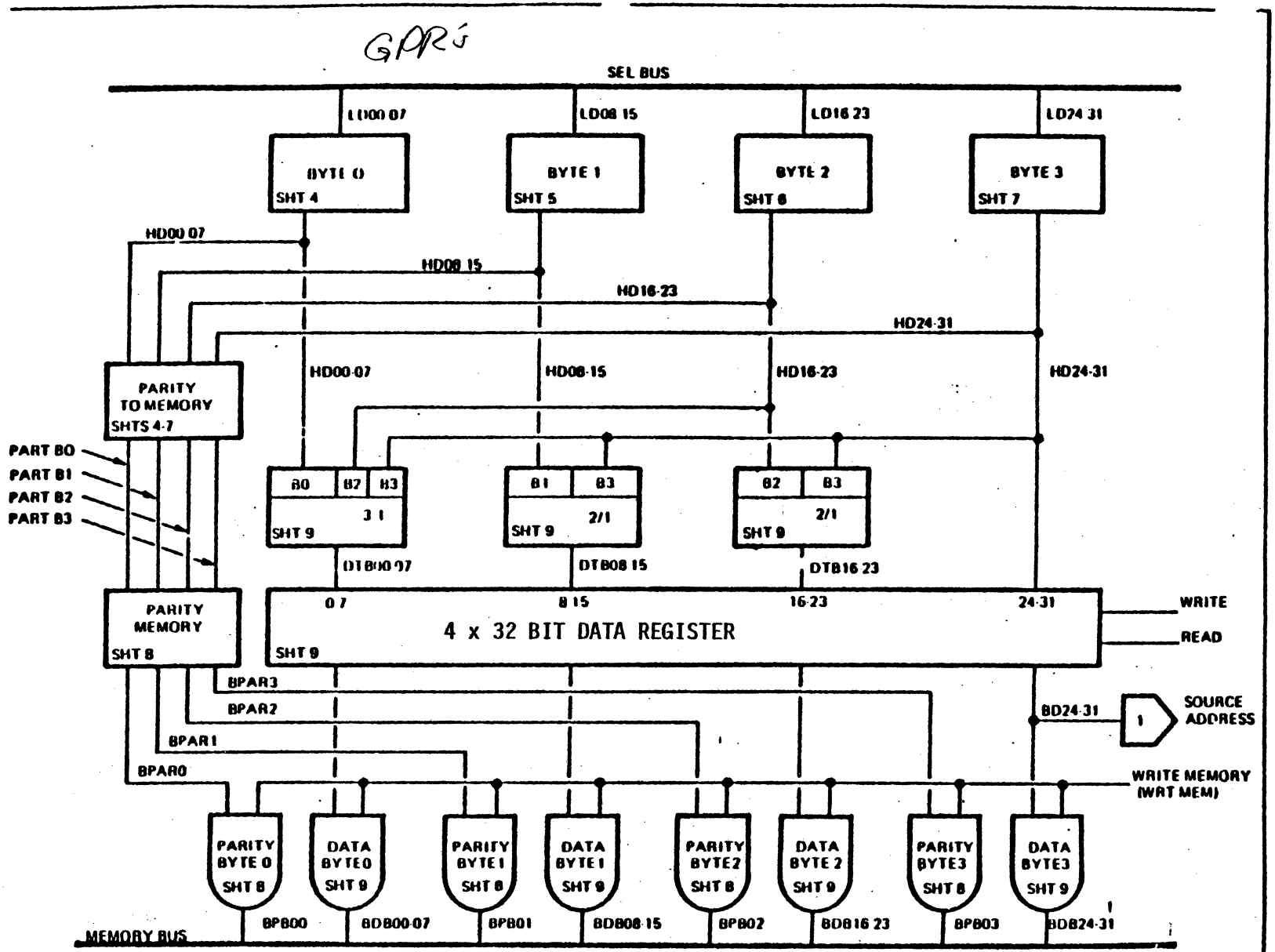
2152 8KW/600 NS
2153 16KW/900 NS

MOS MEMORY MODULES

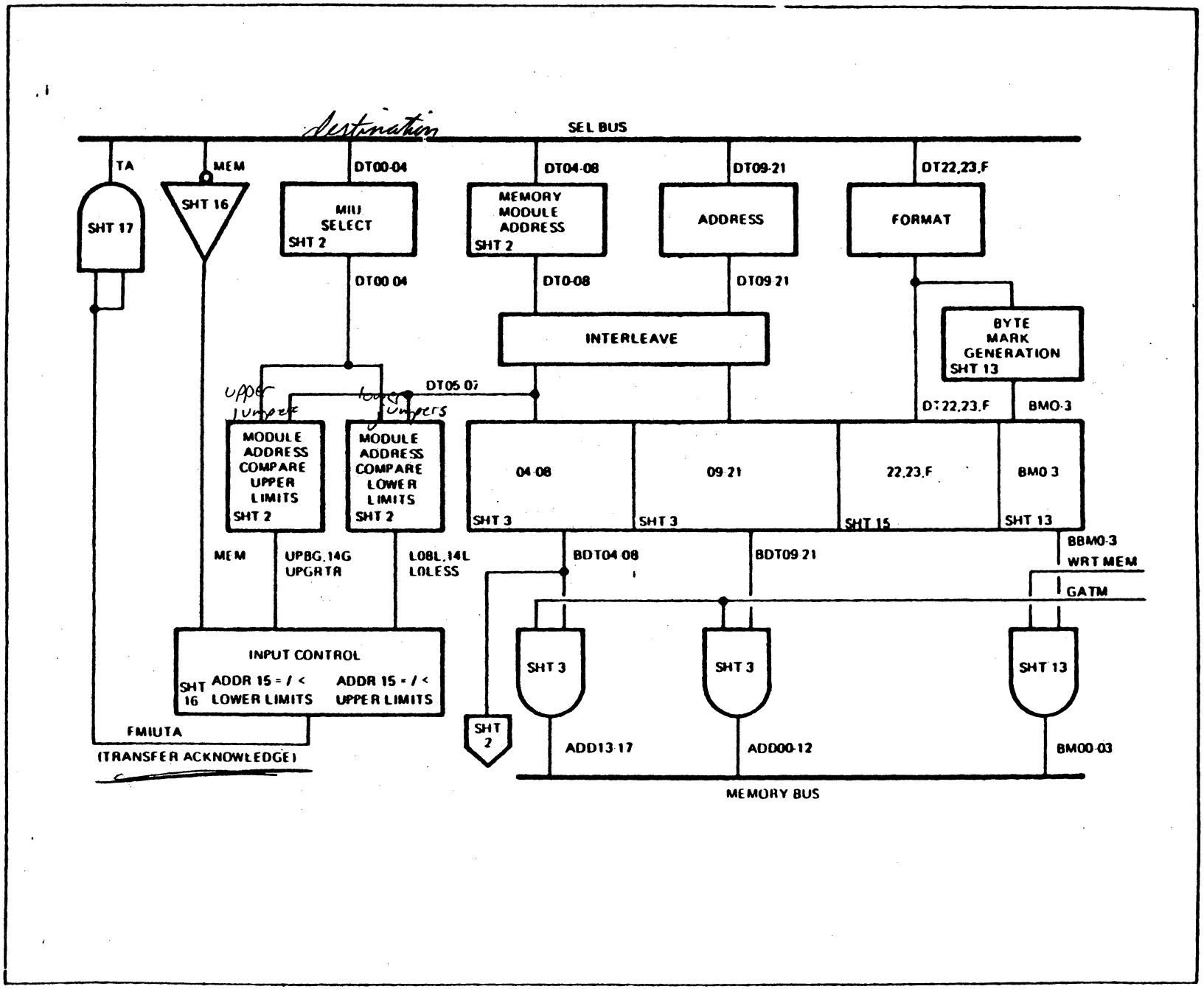
2378 32KW/600 NS
2376 32KW/900 NS
2379 64KW/600 NS
2381 64KW/900 NS



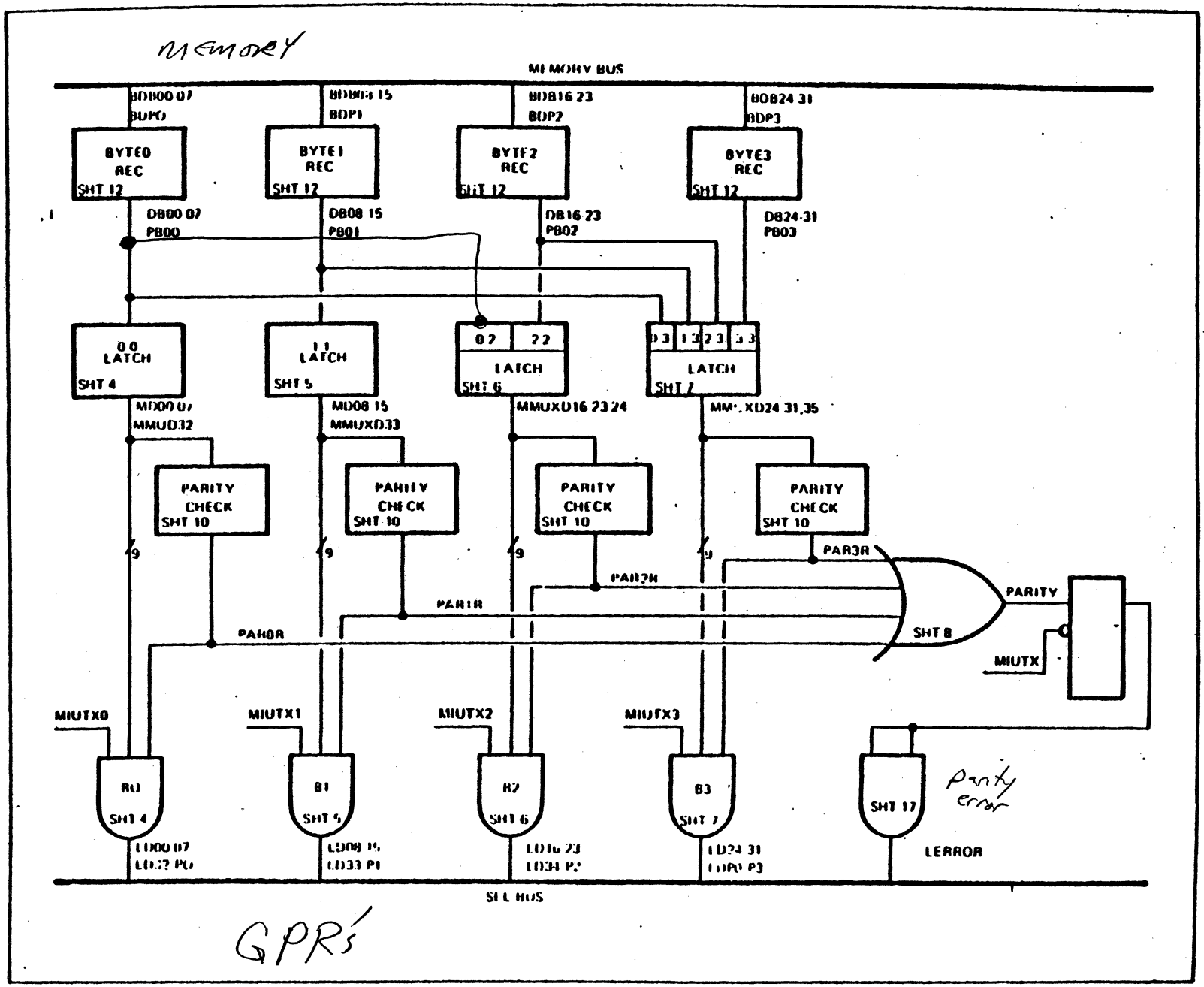

Block Diagram - Memory Bus Controller



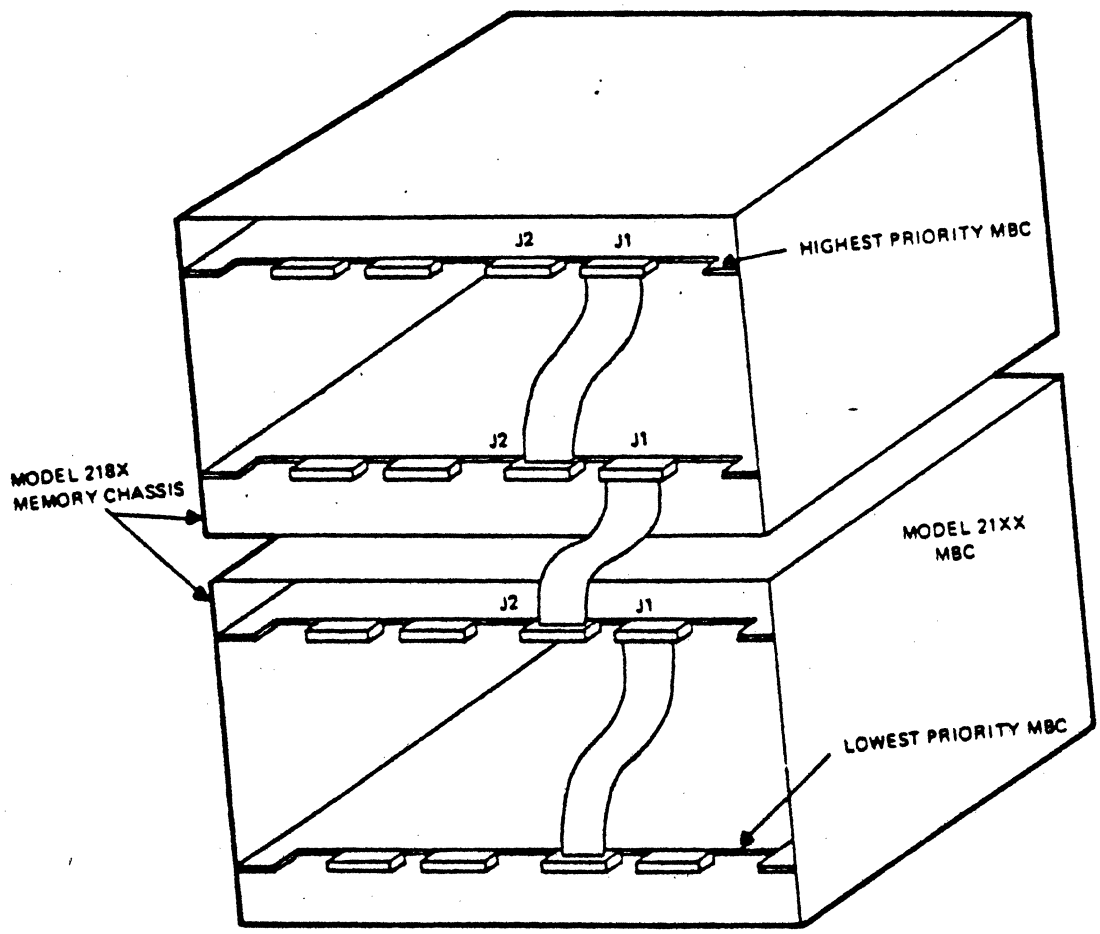

Detailed Block Diagram - Memory Bus Controller (Sheet 1 of 4)




Detailed Block Diagram - Memory Bus Controller (Sheet 2 of 4)

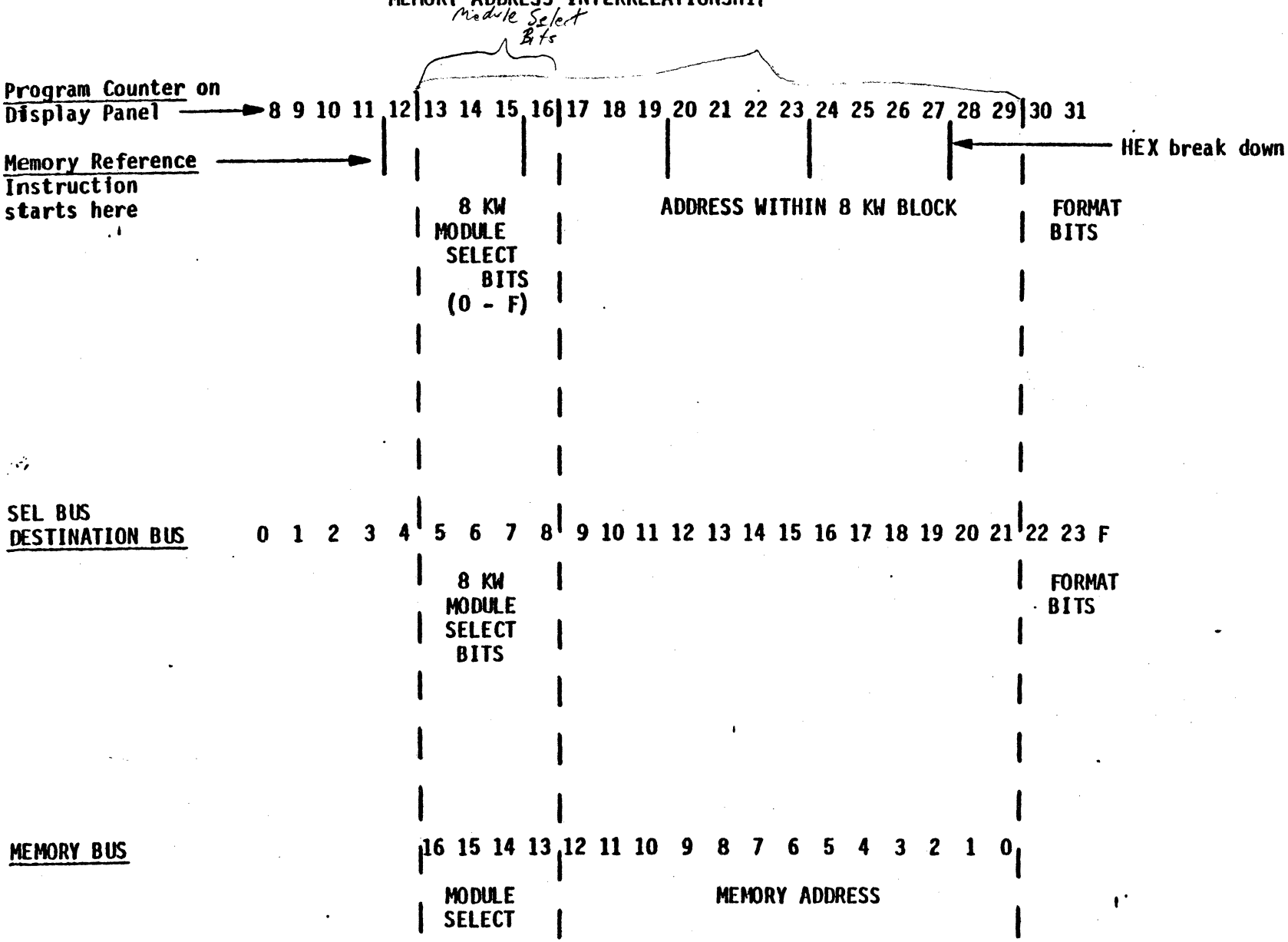



Detailed Block Diagram - Memory Bus Controller (Sheet 3 of 4)




Memory Bus Controller (MBC) Bus Connection

## MEMORY ADDRESS INTERRELATIONSHIP



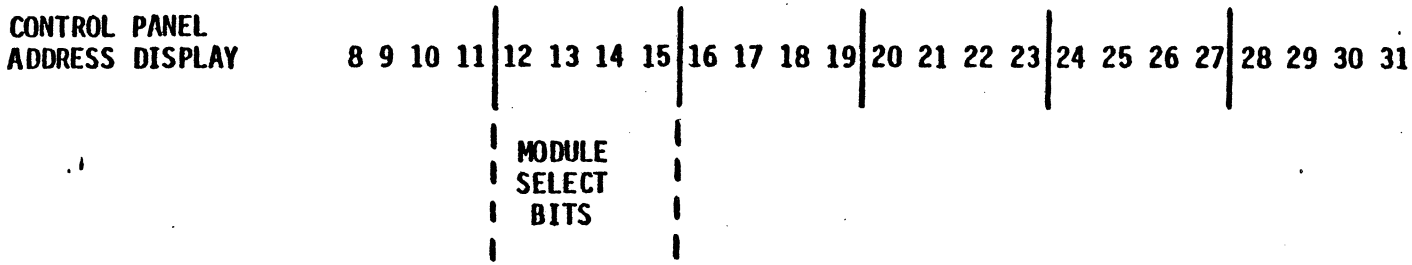
# 8 KW MEMORY ADDRESSING

CONTROL PANEL ADDRESS DISPLAY

| 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MODULE SELECT BITS | | | | | | | | | | | | | | | | | | | |

| MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE | MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE |
|---|---|---|---|---|---|
| 0 | 00000 - 07FFC | 8 KW | 8 | 40000 - 47FFC | 72 KW |
| 1 | 08000 - 0FFFC | 16 KW | 9 | 48000 - 4FFFC | 80 KW |
| 2 | 10000 - 17FFC | 24 KW | A | 50000 - 57FFC | 88 KW |
| 3 | 18000 - 1FFFC | 32 KW | B | 58000 - 5FFFC | 96 KW |
| 4 | 20000 - 27FFC | 40 KW | C | 60000 - 67FFC | 104 KW |
| 5 | 28000 - 2FFFC | 48 KW | D | 68000 - 6FFFC | 112 KW |
| 6 | 30000 - 37FFC | 56 KW | E | 70000 - 77FFC | 120 KW |
| 7 | 38000 - 3FFFC | 64 KW | F | 78000 - 7FFFC | 128 KW |

# 16 KW MEMORY ADDRESSING



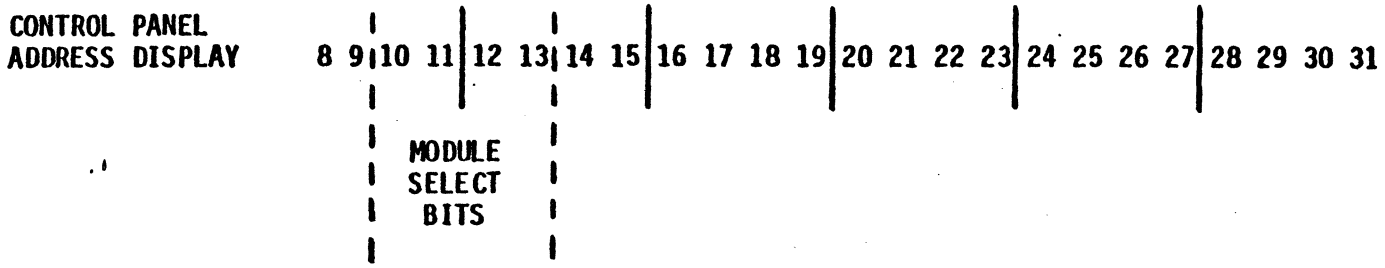

| MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE | MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE |
|---|---|---|---|---|---|
| 0 | 00000 - 0FFFC | 16 KW | 8 | 80000 - 8FFFC | 144 KW |
| 1 | 10000 - 1FFFC | 32 KW | 9 | 90000 - 9FFFC | 160 KW |
| 2 | 20000 - 2FFFC | 48 KW | A | A0000 - AFFFC | 176 KW |
| 3 | 30000 - 3FFFC | 64 KW | B | B0000 - BFFFC | 192 KW |
| 4 | 40000 - 4FFFC | 80 KW | C | C0000 - CFFFC | 208 KW |
| 5 | 50000 - 5FFFC | 96 KW | D | D0000 - DFFFC | 224 KW |
| 6 | 60000 - 6FFFC | 112 KW | E | E0000 - EFFFC | 240 KW |
| 7 | 70000 - 7FFFC | 128 KW | F | F0000 - FFFFC | 256 KW |

# 32 KW MEMORY ADDRESSING

| CONTROL PANEL ADDRESS DISPLAY | 8 9 10 | 11 | 12 13 14 | 15 | 16 17 18 19 | 20 21 22 23 | 24 25 26 27 | 28 29 30 31 |
|---|---|---|---|---|---|---|---|---|
| | | | MODULE SELECT BITS | | | | | |

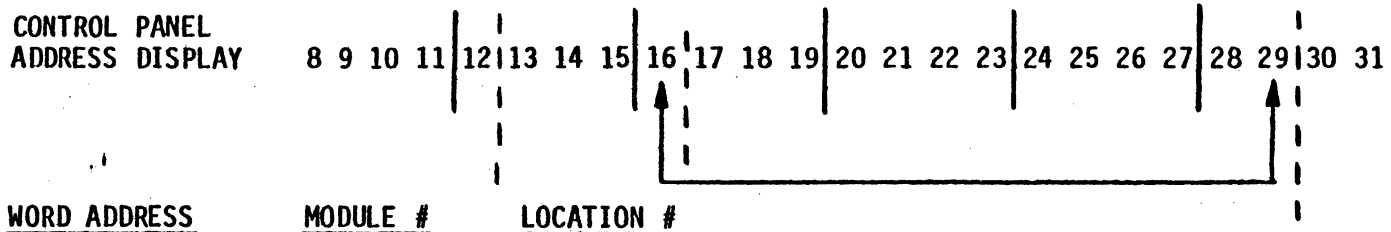
| MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE | MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE |
|---|---|---|---|---|---|
| 0 | 000000-01FFFC | 32 KW | 8 | 100000-11FFFC | 288 KW |
| 1 | 020000-03FFFC | 64 KW | 9 | 120000-13FFFC | 320 KW |
| 2 | 040000-05FFFC | 96 KW | A | 140000-15FFFC | 352 KW |
| 3 | 060000-07FFFC | 128 KW | B | 160000-17FFFC | 384 KW |
| 4 | 080000-09FFFC | 160 KW | C | 180000-19FFFC | 416 KW |
| 5 | 0A0000-0BFFFC | 192 KW | D | 1A0000-1BFFFC | 448 KW |
| 6 | 0C0000-0DFFFC | 224 KW | E | 1C0000-1DFFFC | 480 KW |
| 7 | 0E0000-0FFFFC | 256 KW | F | 1E0000-1FFFFC | 512 KW |

# 64 KW MEMORY ADDRESSING



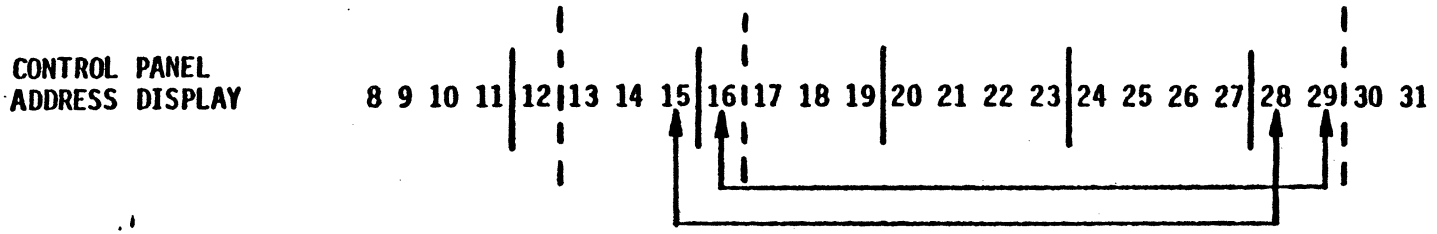

| MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE | MEMORY MODULE # | HEX ADDRESS RANGE | PHYSICAL MEMORY SIZE |
|---|---|---|---|---|---|
| 0 | 000000-03FFFC | 64 KW | 8 | 200000-23FFFC | 576 KW |
| 1 | 040000-07FFFC | 128 KW | 9 | 240000-27FFFC | 640 KW |
| 2 | 080000-0BFFFC | 192 KW | A | 280000-2BFFFC | 704 KW |
| 3 | 0C0000-0FFFFC | 256 KW | B | 2C0000-2FFFFC | 768 KW |
| 4 | 100000-13FFFC | 320 KW | C | 300000-33FFFC | 832 KW |
| 5 | 140000-17FFFC | 384 KW | D | 340000-37FFFC | 896 KW |
| 6 | 180000-1BFFFC | 448 KW | E | 380000-3BFFFC | 960 KW |
| 7 | 1C0000-1FFFFC | 512 KW | F | 3C0000-3FFFFC | 1024 KW |

## 2 WAY INTERLEAVING




| WORD ADDRESS | MODULE # | LOCATION # |
|---|---|---|
| 0000 | 0 | 0 |
| 0004 | 1 | 0 |
| 0008 | 0 | 8 |
| 000C | 1 | 8 |
| 0010 | 0 | 10 |
| 0014 | 1 | 10 |
| 0018 | 0 | 18 |
| 001C | 1 | 18 |
| ↓ | ↓ | ↓ |
| 7FF8 | 0 | 7FF8 |
| 7FFC | 1 | 7FF8 |
| 8000 | 0 | 4 |
| 8004 | 1 | 4 |
| 8008 | 0 | C |
| 800C | 1 | C |

## 4 WAY INTERLEAVING




| WORD ADDRESS | MODULE # | LOCATION # | WORD ADDRESS | MODULE # | LOCATION # |
|---|---|---|---|---|---|
| 0000 | 0 | 0 | 10000 | 0 | 8 |
| 0004 | 1 | 0 | 10004 | 1 | 8 |
| 0008 | 2 | 0 | 10008 | 2 | 8 |
| 000C | 3 | 0 | 1000C | 3 | 8 |
| 0010 | 0 | 10 | 10010 | 0 | 18 |
| 0014 | 1 | 10 | 10014 | 1 | 18 |
| 0018 | 2 | 10 | 10018 | 2 | 18 |
| 001C | 3 | 10 | 1001C | 3 | 18 |
| 8000 | 0 | 4 | 18000 | 0 | C |
| 8004 | 1 | 4 | 18004 | 1 | C |
| 8008 | 2 | 4 | 18008 | 2 | C |
| 800C | 3 | 4 | 1800C | 3 | C |
| 8010 | 0 | 14 | 18010 | 0 | 1C |
| 8014 | 1 | 14 | 18014 | 1 | 1C |
| 8018 | 2 | 14 | 18018 | 2 | 1C |
| 801C | 3 | 14 | 1801C | 3 | 1C |

SYSTEMS ENGINEERING LABORATORIES

# Technical Support Bulletin

| TITLE Multiported, MBC Jumpering Information | | TSB Number 0020 |
|---|---|---|
| Product SEL 32/35 & 55 | Model Number 2164 & 2162 | Date 1/27/78 |

The following pages contain all the jumpering information required for the installation of a model 2164 or 2162 Multiported Memory Bus Controller. Pictorial diagrams (Pages 7 & 8) are included to assist in cabling the Multiported MBC's.

Refer to TSB #0006 for information on configuring shared memory. TSB #0006 references the Model 2150 MBC; jumpers referred to in that TSB should be cross-referenced to the Multiported MBC jumpers called out on pages 2 through 6 of this TSB. The Model 2164 & 2162 MBC's do not require shared memory jumpers referenced in TSB #0006.

NOTE: An MBC cannot be configured with both dedicated memory module zero and 2 or 4 way interleaved memory; these features are mutually exclusive.

| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

| | Model 2164 | (Wire Wrap) | Model 2162 | (Copper) |
|---|---|---|---|---|
| 8k & 16k Memory | Loc. | Jumpers | Loc | Jumpers |
| Non-dedicated Memory Module Zero | D5<br>D13 | 1, 4 & 7<br>1 & 6 | D19<br>D18 | 1, 4 & 7<br>1 & 6 |

| 8k & 16k Memory | | | | |
|---|---|---|---|---|
| Dedicated Memory Module Zero | D5<br>D13 | 1, 4, 7 & 8<br>1 & 6 | D19<br>D18 | 1, 4, & 7<br>1, 6 & 8 |

| 8k Memory Modules | | | | |
|---|---|---|---|---|
| 2 Way Interleaved | D5<br>D13 | 2, 4 & 7<br>2 & 6 | D19<br>D18 | 2, 4 & 7<br>2 & 6 |

| 8k Memory Modules | | | | |
|---|---|---|---|---|
| 4 Way Interleaved | D5<br>D13 | 2, 5 & 7<br>2 & 4 | D19<br>D18 | 2, 5 & 7<br>2 & 4 |

| 16k Memory Modules | | | | |
|---|---|---|---|---|
| 2 Way Interleaved | D5<br>D13 | 1, 3 & 7<br>3 & 6 | D19<br>D18 | 1, 3 & 7<br>3 & 6 |

| 16k Memory Modules | | | | |
|---|---|---|---|---|
| 4 Way Interleaved | D5<br>D13 | 1, 3 & 6<br>3 & 5 | D19<br>D18 | 1, 3 & 6<br>3 & 5 |

| Configuration Jumpers for 8k or 16k Memory Modules | | | | |
|---|---|---|---|---|
| | Model 2164 | | Model 2162 | |
| | Loc. | Jumpers | Loc. | Jumpers |
| 8k Memory Modules | G5 | 1, 3, 5 & 7 | C19 | 1, 3, 5 & 7 |
| 16k Memory Modules | G5 | 2, 4, 6 & 8 | C19 | 2, 4, 6 & 8 |

| TITLE: Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

| | | Wire Wrap Model 2164 * | | Copper Model 2162 | |
|---|---|---|---|---|---|
| 8k & 16k Memory | Bus Priority | Loc | Jumpers | Loc | Jumpers |
| SEL BUS Transmit Priority | 1 | B17 | 1 | A18 | 1 |
| | 2 | B17 | 2 | A18 | 2 |
| | 3 | B17 | 3 | A18 | 3 |
| | 4 | B17 | 4 | A18 | 4 |
| | 5 | B17 | 5 | A18 | 5 |
| | 6 | B17 | 6 | A18 | 6 |
| | 7 | B17 | 7 | A18 | 7 |
| | 8 | B17 | 8 | A18 | 8 |

| 8k & 16k Memory | | Model 2164 * | | Model 2162 | |
|---|---|---|---|---|---|
| SEL BUS Receive Priority | 1 | B18 | None | B17 | None |
| | 2 | B18 | 1 | B17 | 1 |
| | 3 | B18 | 1 & 2 | B17 | 1 & 2 |
| | 4 | B18 | 1, 2 & 3 | B17 | 1, 2 & 3 |
| | 5 | B18 | 1, 2, 3 & 4 | B17 | 1, 2, 3 & 4 |
| | 6 | B18 | 1, 2, 3, 4, & 5 | B17 | 1, 2, 3, 4, & 5 |
| | 7 | B18 | 1, 2, 3, 4, 5 & 6 | B17 | 1, 2, 3, 4, 5 & 6 |
| | 8 | B18 | 1, 2, 3, 4, 5, 6, & 7 | B17 | 1, 2, 3, 4, 5, 6, & 7 |

| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

| 8k & 16k Memory | Inhibit Line | (Wire Wrap) Model 2164 ✳ Loc | Jumpers | (Copper) Model 2162 Loc | Jumpers |
|---|---|---|---|---|---|
| | 0 | L23 | None | L14 | None |
| | 1 | L23 | 1 | L14 | 1 |
| | 2 | L23 | 2 | L14 | 2 |
| | 3 | L23 | 1 & 2 | L14 | 1 & 2 |

| 8k & 16 Memory | Model 2164 = Loc N2 ✳ | Model 2162 = Loc H16 |
|---|---|---|
| | 1 = Jumper Installed | 1 = jumper |
| | 0 = Jumper Removed | |

| | | Jumper Pins | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MSB | | | | | | | | LSB |
| | Address | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Lower Address Boundary in 8k Word Increments | 000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 001 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| | 002 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| | 003 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| | 004 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| | 005 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| | 006 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| | 007 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| | 008 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| | 009 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| | 00A | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| | 00B | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 00C | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| | 00D | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| | 00E | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| | 00F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 1FE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 1FF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DT0-DT8

| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

| 8k & 16k Memory | Model 2164 = Loc L02 | Model 2162 = Loc H15 |
|---|---|---|

1 = Jumper Installed

0 = Jumper Removed

| | Jumper Pins | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | MSB | | | | | | | | LSB |
| Address | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 001 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 002 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 003 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 004 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 005 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 006 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 007 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 008 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 009 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 00A | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 00B | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 00C | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 00D | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 00E | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 00F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1FE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1FF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

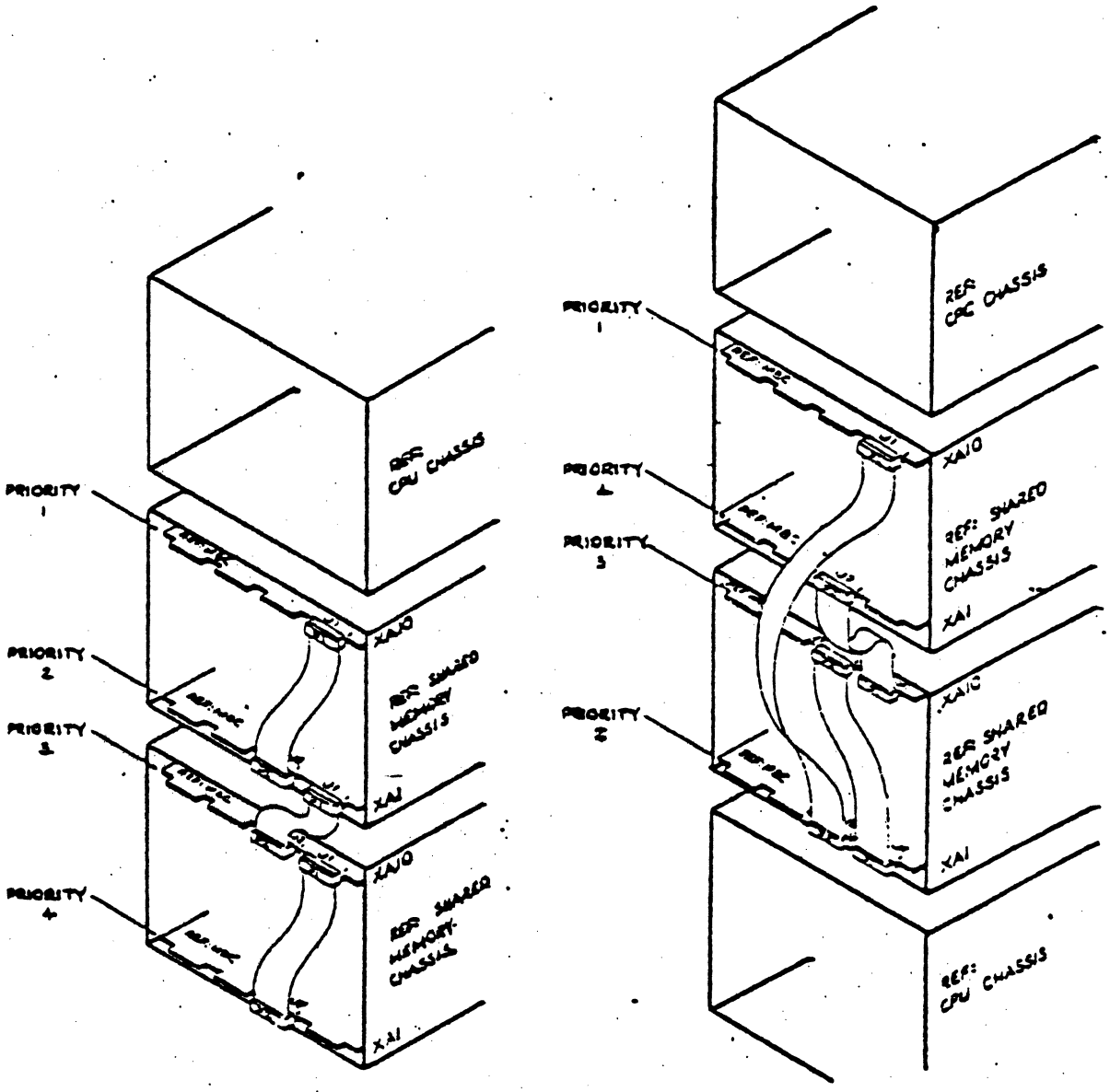
Upper Address Boundary in 8k Work Increments

| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

| | (Wire Wrap) Model 2164 | | (Copper) Model 2162 | |
|---|---|---|---|---|
| 8k & 16k Memory | Loc | Jumpers | Loc | Jumpers |
| Parity Selection option (odd) | N02 | 10 Removed | H16 | 10 Removed |
| Parity Selection option (even) | N02 | 10 Installed | H16 | 10 Installed |

NOTE: Parity is normally selected as odd.

| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|

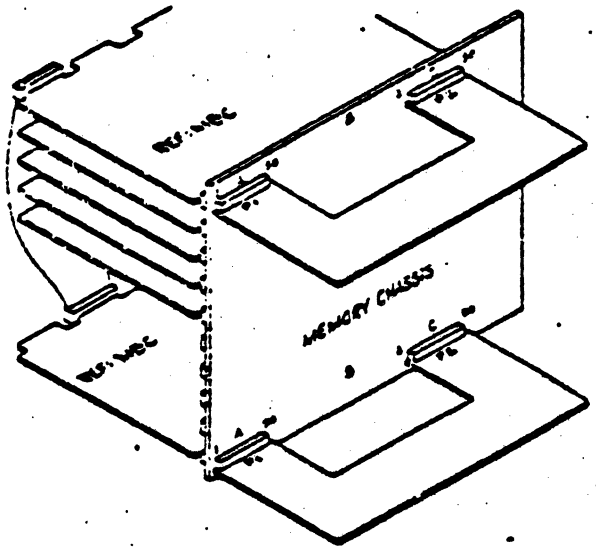


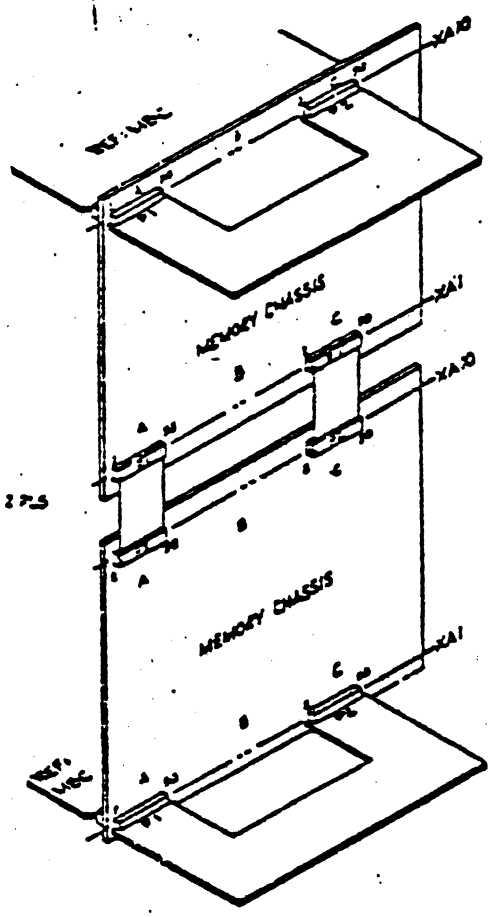

NOTES:

1. MBC PRIORITIES DETERMINED BY CABLING CONFIGURATIONS. CABLE J1 OF HIGHEST PRIORITY MBC TO J2 OF SUCCEEDING PRIORITY MBC. J2 NOT USED ON HIGHEST PRIORITY MBC. J1 NOT USED ON LOWEST PRIORITY MBC.

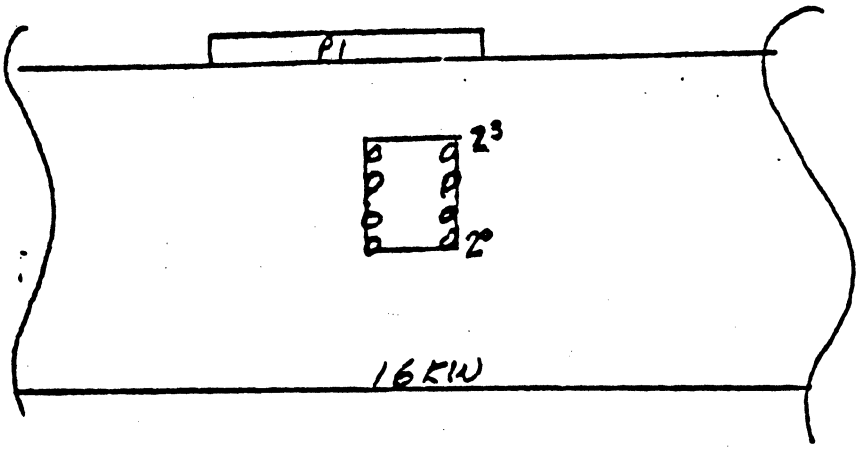
NOTE: 4 Ported Memory must use 2162 only

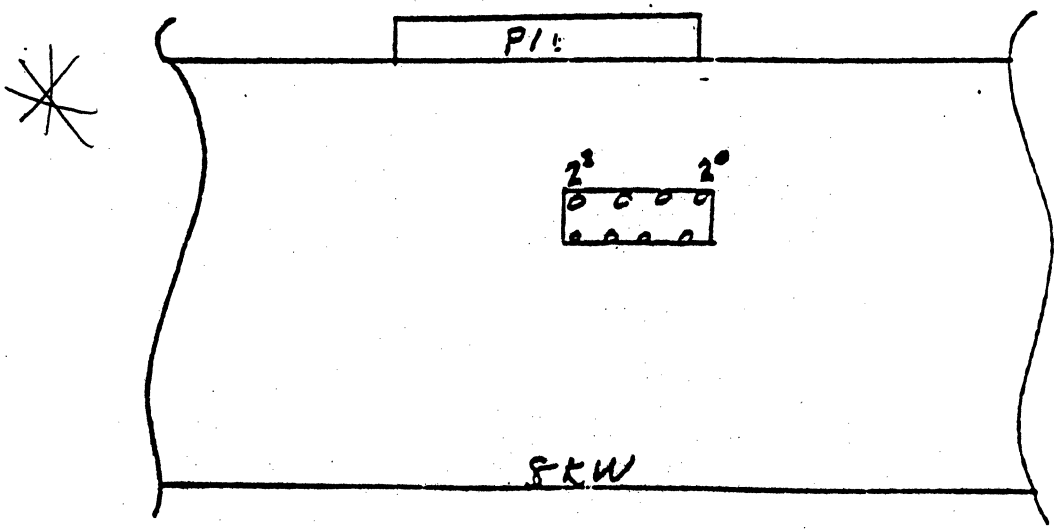
| TITLE Multiported, MBC Jumpering Information | TSB NO. 0020 |
|---|---|



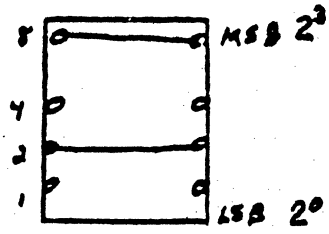




Refers to Model 2164 & 2162




Jumper in = ϕ




EXAMPLE: MODULE #5

CORE MEMORY MODULE SWITCH

Proven COMPUTER Performance

SYSTEMS

A Subsidiary of GOULD INC.

# TECHNICAL SUPPORT BULLETIN

| TITLE | TSB No. |
|---|---|
| MOS MEMORY BUS CONTROLLER | 0069B |

| Product | Model No. | Date |
|---|---|---|
| 32/77 | 2382 | 03/25/82 |

The following pages contain the necessary jumpering information to install the MOS MBC.

## MBC ADDRESS JUMPERS (Jumpers Installed)

D10 - D15 D0 - D8

| MODULE NUMBER 32K | MODULE NUMBER 64K | LOW ADDRESS LIMIT | | UPPER ADDRESS LIMIT | | LOWER LIMIT C10-7 | C10-10 | C10-9 | C10-8 | C13-7 | C13-10 | C13-9 | C13-8 | C11-6 | UPPER LIMIT C11-7 | C11-10 | C11-9 | C11-8 | C12-10 | C12-7 | C12-9 | C12-8 | C13-6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 00 | 0000 | 00 | FFFC | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | |
| 0 | 0 | 01 | 0000 | 01 | FFFC | X | X | X | X | X | X | X | | X | X | X | X | X | X | X | | | |
| 1 | 0 | 02 | 0000 | 02 | FFFC | X | X | X | X | X | X | | X | X | X | X | X | X | X | | X | | |
| 1 | 0 | 03 | 0000 | 03 | FFFC | X | X | X | X | X | X | | | X | X | X | X | X | X | | | | |
| 2 | 1 | 04 | 0000 | 04 | FFFC | X | X | X | X | X | | X | X | X | X | X | X | X | X | | X | X | |
| 2 | 1 | 05 | 0000 | 05 | FFFC | X | X | X | X | X | | X | | X | X | X | X | X | X | | X | | |
| 3 | 1 | 06 | 0000 | 06 | FFFC | X | X | X | X | X | | | X | X | X | X | X | X | X | | | X | |
| 3 | 1 | 07 | 0000 | 07 | FFFC | X | X | X | X | X | | | | X | X | X | X | X | X | | | | |
| 4 | 2 | 08 | 0000 | 08 | FFFC | X | X | X | X | | X | X | X | X | X | X | X | X | | X | X | X | |
| 4 | 2 | 09 | 0000 | 09 | FFFC | X | X | X | X | | X | X | | X | X | X | X | X | | X | X | | |
| 5 | 2 | 0A | 0000 | 0A | FFFC | X | X | X | X | | X | | X | X | X | X | X | X | | X | | X | |
| 5 | 2 | 0B | 0000 | 0B | FFFC | X | X | X | X | | X | | | X | X | X | X | X | | X | | | |
| 6 | 3 | 0C | 0000 | 0C | FFFC | X | X | X | X | | | X | X | X | X | X | X | X | | | X | X | |
| 6 | 3 | 0D | 0000 | 0D | FFFC | X | X | X | X | | | X | | X | X | X | X | X | | | X | | |
| 7 | 3 | 0E | 0000 | 0E | FFFC | X | X | X | X | | | | X | X | X | X | X | X | | | | X | |
| 7 | 3 | 0F | 0000 | 0F | FFFC | X | X | X | X | | | | | X | X | X | X | X | | | | | |
| | 4 | 10 | 0000 | 13 | FFFC | X | X | X | | X | X | X | X | X | X | X | X | | X | X | | | |
| | 5 | 14 | 0000 | 17 | FFFC | X | X | X | | X | | X | X | X | X | X | X | | X | | | | |
| | 6 | 1B | 0000 | 1B | FFFC | X | X | X | | | X | X | X | X | X | X | X | | | X | | | |
| | 7 | 1C | 0000 | 1F | FFFC | X | X | X | | | | X | X | X | X | X | X | | | | | | |
| | 8 | 20 | 0000 | 23 | FFFC | X | X | | X | X | X | X | X | X | X | X | | X | X | X | | | |
| | 9 | 24 | 0000 | 27 | FFFC | X | X | | X | X | | X | X | X | X | X | | X | X | | | | |
| | 10 | 2B | 0000 | 2B | FFFC | X | X | | X | | X | X | X | X | X | X | | X | | X | | | |
| | 11 | 2C | 0000 | 2F | FFFC | X | X | | X | | | X | X | X | X | X | | X | | | | | |
| | 12 | 30 | 0000 | 33 | FFFC | X | X | | | X | X | X | X | X | X | X | | | X | X | | | |
| | 13 | 34 | 0000 | 37 | FFFC | X | X | | | X | | X | X | X | X | X | | | X | | | | |
| | 14 | 38 | 0000 | 3B | FFFC | X | X | | | | X | X | X | X | X | X | | | | X | | | |
| | 15 | 3C | 0000 | 3F | FFFC | X | X | | | | | X | X | X | X | X | | | | | | | |

## SEL BUS PRIORITY TRANSMIT (Jumpers)

PRI 1 - A16-7
PRI 2 - A16-6
PRI 3 - A16-3
PRI 4 - A16-8
PRI 5 - A16-5
PRI 6 - A16-4
PRI 7 - A16-1
PRI 8 - A16-2

## SEL BUS PRIORITY RECEIVE (Jumpers)

PRI 1 - NONE
PRI 2 - A18-8
PRI 3 - A18-8,6
PRI 4 - A18-8,6,4
PRI 5 - A18-8,6,4,7
PRI 6 - A18-8,6,4,7,5
PRI 7 - A18-8,6,4,7,5,3
PRI 8 - A18-8,6,4,7,5,3,2

✱ Dedicated Option don't have

**32K Module**

C12-2,3,4,5
C10-4,2
C11-4,3

**64K Module**

C12-3,4,5
C10-4
C11-5,4,3,2

Jumpers In

## Configuration Jumpers (Jumpers Installed)

### Non Dedicated - No Interleave

| 32K Module | 64K Module |
|---|---|
| C10-2 | C11-4,3,2 |
| C11-3,4 | C12-5 |
| C12-5 | C13-4 |
| C13-4 | C14-4,9,6 |
| C14-4,9,2,7 | C15-5,8,1 |
| C15-5,9 | C16-3,7 |
| C16-3,7 | C17-6 |
| C17-6 | D15-2 |
| D15-1,2 | |

### 2 Way - Interleave - Non Dedicated

| 32K Module | 64K Module |
|---|---|
| C10-2 | C11-4,3,2 |
| C11-4,3 | C12-5 |
| C12-5 | C13-4 |
| C13-5 | C14-5,9,6 |
| C14-4,9,2,7 | C15-5,8,1 |
| C15-5,9 | C16-3,5 |
| C16-3,10 | C17-6 |
| C17-6 | D15-2 |
| D15-1,2 | D15-2 |

### 4 Way - Interleave - Non Dedicated

| 32K Module | 64K Module |
|---|---|
| C10-2 | C11-4,3,2 |
| C11-4,3 | C12-5 |
| C12-5 | C13-4 |
| C13-5 | C14-5,6 |
| C14-3,9,2,7 | C15-10,5,8,1 |
| C15-5,9 | C16-3,5 |
| C16-3,10 | C17-9 |
| C17-7 | D15-2 |
| D15-1,2 | |

## INHIBIT/ECHO DECODE (JUMPERS)

A modification has been made to the Model 2382 MBC, (160-103265-001 Rev. G or higher), to allow proper operation with the Concept 32/87 CPU. However, this does effect all other systems, (32/7X, 32/57, etc...), in which this board is used.

The following chart should be used for jumpering the INHIBIT/ECHO Bits.

| | {32/7X, 32/5X, 32/2X, 32/8X} | {32/7X, 32/5X, 32/2X} | {32/8X} |
| --- | --- | --- | --- |
| | SEL BUS PRIORITY | ECHO BITS | INHIBIT BITS | INHIBIT BITS |
| PRI 1 or 5 | NONE | E13-7 | E13-4,5,6,7 |
| PRI 2 or 6 | C17-1 | E13-6 | E13-4,5,6,7 |
| PRI 3 or 7 | C17-2 | E13-5 | E13-4,5,6,7 |
| PRI 4 | C17-1,2 | E13-4 | E13-4,5,6,7 |

## ERROR DETECTION OPTION (JUMPERS)

| | | |
| --- | --- | --- |
| | SINGLE BIT | C17-3 * Not Reported if J is out |
| NORMAL | DOUBLE BIT | NONE ** |
| | PROGRAM CONTROL | A18-1 |

*NOTE: Double Bit detection is the standard mode used when running under the operating system.

**NOTE: When performing Preventative Maintenance or troubleshooting a suspected problem in memory, the jumper for Single Bit detection should be inserted on the MBC. The Error Correction jumpers on the 32K and 64K MOS Memory Modules, (reference TSB 073A and TSB 0102), in location C10, should also be in the disabled positions

when troubleshooting/performing P.M. Jumpering for Single Bit detection will not report Double Bit errors.

Any module that fails the memory diagnostic should be replaced as soon as possible.

Before returning the system to the customer/user, the MBC should be jumpered for Double Bit detection, (remove C17-3), and the MOS Memory Modules should have the Error Correction enabled.

SYSTEMS
ENGINEERING LABORATORIES

# TECHNICAL SUPPORT BULLETI

| TITLE<br>MOS Memory Jumper Configurations | | TSB No.<br>0073 |
|---|---|---|
| Product<br>32 & 64K Word MOS Memory | Model No.<br>2376, 2378, 2379, 2301 | Date<br>3/20/79 |

32K Word Module Storage

Board Jumpers

| Socket Location | Pin Number | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| B13 | X | | X | | X | | X | | X | |
| B14 | | X | | X | | X | | X | | |

32K Word Module Interface Board Jumpers

| Memory Module Number | Socket A15 Pin Number | | | | | | | | | | Socket A16 Pin Number | | | | | | | | | | Socket A16B Pin Number | | | | Socket C10 Pin Number | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 0 | X | | X | | | | | | X | X | X | | X | | X | | X | | | | X | X | X | X | | | X | X |
| 1 | X | | X | | | | X | | X | X | X | | X | | X | | X | | | | | X | X | X | | | X | X |
| 2 | X | | X | | | | | X | X | X | X | | X | | X | | X | | | | X | | X | X | | | X | X |
| 3 | X | | X | | | | X | X | X | X | X | | X | | X | | X | | | | | | X | X | | | X | X |
| 4 | X | | X | | | | | | X | X | X | | X | | X | | X | | X | | X | X | X | | | | X | X |
| 5 | X | | X | | | | X | | X | X | X | | X | | X | | X | | X | | | X | X | | | | X | X |
| 6 | X | | X | | | | | X | X | X | X | | X | | X | | X | | X | | X | | X | | | | X | X |
| 7 | X | | X | | | | X | X | X | X | X | | X | | X | | X | | X | | | | X | | | | X | X |
| 8 | X | | X | | | | | | X | X | X | | X | | X | | X | | X | | X | X | | X | | | X | X |
| 9 | X | | X | | | | X | | X | X | X | | X | | X | | X | | X | | | X | | X | | | X | X |
| A | X | | X | | | | | X | X | X | X | | X | | X | | X | | X | | X | | | X | | | X | X |
| B | X | | X | | | | X | X | X | X | X | | X | | X | | X | | X | | | | | X | | | X | X |
| C | X | | X | | | | | | X | X | X | | X | | X | | X | | X | X | X | X | | | | | X | X |
| D | X | | X | | | | X | | X | X | X | | X | | X | | X | | X | X | | X | | | | | X | X |
| E | X | | X | | | | | X | X | X | X | | X | | X | | X | | X | X | X | | | | | | X | X |
| F | X | | X | | | | X | X | X | X | X | | X | | X | | X | | X | X | | | | | | | X | X |

| TITLE | TSB No. |
|---|---|
| MOS Memory Jumper Configurations | 0073 |

## 64K Word Module Storage

### Board Jumpers

| Socket Location | Pin Number | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| B13 | X | | X | | X | | X | | X | |
| B14 | | X | | X | | X | | X | X | |

## 64K Word Module Interface Board Jumpers

| Memory Module Number | Socket A15 Pin Number | | | | | | | | | | Socket A16 Pin Number | | | | | | | | | | Socket A18B Pin Number | | | | Socket C10 Pin Number | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 0 | X | X | | | X | | | X | X | | X | | X | | X | | | | | | X | X | X | X | | | X | X |
| 1 | X | X | | | X | | X | X | X | | X | | X | | X | | | | | | | X | X | X | | | X | X |
| 2 | X | X | | | X | | | X | X | | X | | X | | X | | | | | X | X | | X | X | | | X | X |
| 3 | X | X | | | X | | X | X | X | | X | | X | | X | | | | | X | | | X | X | | | X | X |
| 4 | X | X | | | X | | | X | X | | X | | X | | X | | X | | | | X | X | X | | | | X | X |
| 5 | X | X | | | X | | X | X | X | | X | | X | | X | | X | | | | | X | X | | | | X | X |
| 6 | X | X | | | X | | | X | X | | X | | X | | X | | X | | X | X | X | | X | | | | X | X |
| 7 | X | X | | | X | | X | X | X | | X | | X | | X | | X | | X | X | | | X | | | | X | X |
| 8 | X | X | | | X | X | | X | X | | X | | X | | X | | | | | | X | X | | X | | | X | X |
| 9 | X | X | | | X | X | X | X | X | | X | | X | | X | | | | | | | X | | X | | | X | X |
| A | X | X | | | X | X | | X | X | | X | | X | | X | | | | | X | X | | | X | | | X | X |
| B | X | X | | | X | X | X | X | X | | X | | X | | X | | | | | X | | | | X | | | X | X |
| C | X | X | | | X | X | | X | X | | X | | X | | X | | | | X | | X | X | | | | | X | X |
| D | X | X | | | X | X | X | X | X | | X | | X | | X | | | | X | | | X | | | | | X | X |
| E | X | X | | | X | X | | X | X | | X | | X | | X | | | | X | X | X | | | | | | X | X |
| F | X | X | | | X | X | X | X | X | | X | | X | | X | | | | X | X | | | | | | | X | X |

X = Jumper in

Not Error Correcting
" "

## 900NS Module Refresh

### Board Jumpers

| Socket Location | Pin Number | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B6 | | | X | | X | X | X | |
| C7 | | | | X | | | | |
| C11 | | | | X | | | | |

## 600NS Module Refresh

### Board Jumpers

| Socket Location | Pin Number | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| B6 | | | X | | X | X | X | |
| C7 | | X | | | | | | |
| C11 | | X | | | | | | |

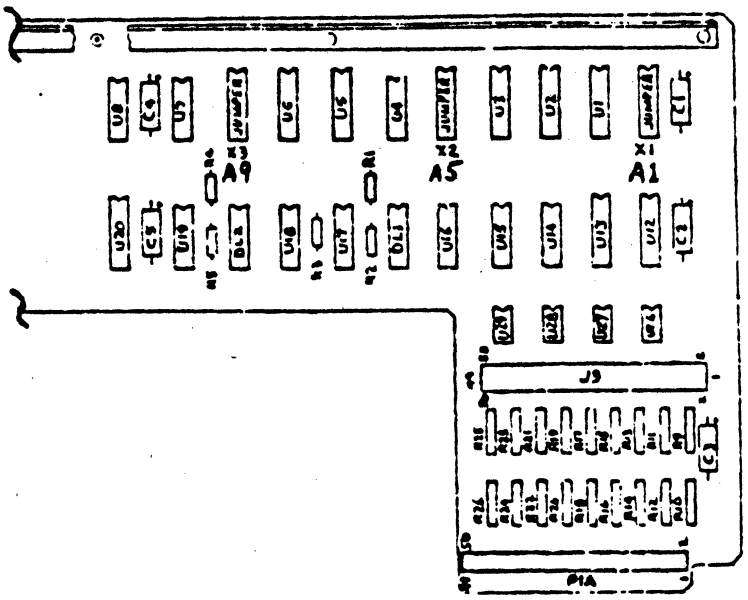
SYSTEMS
ENGINEERING LABORATORIES

# TECHNICAL SUPPORT BULLETIN

| TITLE | | TSB No. |
|---|---|---|
| MOS MBC REFRESH TERMINATOR JUMPER CONFIGURATION | | 0122 |
| **Product** | **Model No.** | **Date** |
| SEL 32 MOS MBC | | 10-20-80 |

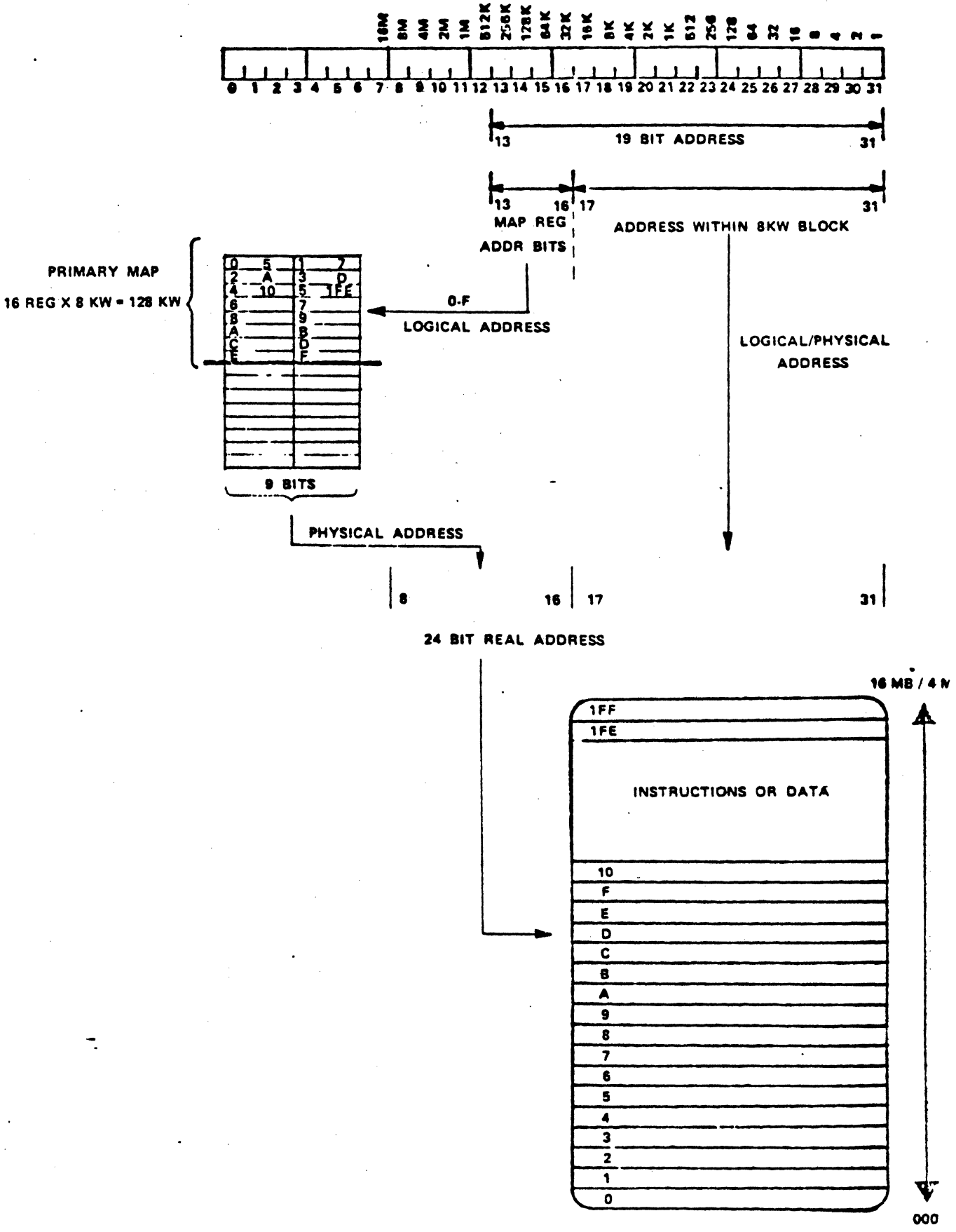
The following chart is an excerpt from drawing 130-103314-000. It will aid in configuring the MOS MBC refresh logic for a MOS memory system. The mechanical section of the bus terminator with jumper locations (160-103314-001) is shown for reference. Refer to TSB 0073A for MOS memory jumper configurations.

| MEMORY SYSTEM | SOCKET LOCATION A1 PIN NO. — | | | | | | | | SOCKET LOCATION A5 PIN NO. — | | | | | | | | SOCKET LOCATION A9 PIN NO. — | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 16K WORDS 600 NANOSECONDS | | X | | | | | | | | X | | | | | | | X | X | | X | X | | X |
| 32K WORDS 600 NANOSECONDS | | X | | | | | | | | X | | | | | | | X | X | | X | X | X | |
| 64K WORDS 600 NANOSECONDS | | X | | | | | | | | X | | | | | | | X | X | | X | X | X | |
| 32K WORDS 900 NANOSECONDS | | | | X | | | | | | | | X | | | | | X | X | | X | X | X | |
| 64K WORDS 900 NANOSECONDS | | | | X | | | | | | | | X | | | | | X | X | | X | X | X | |

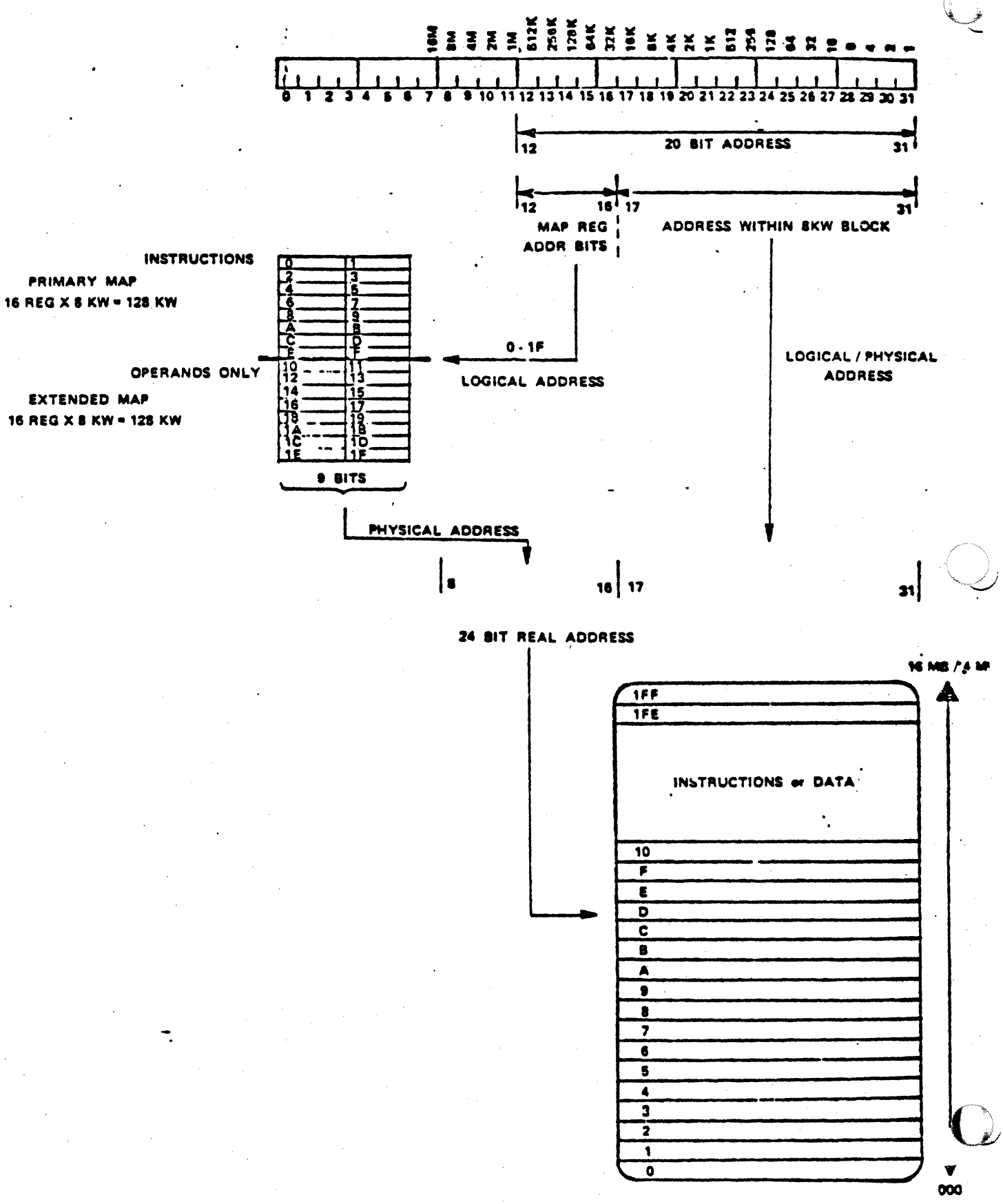


# MEMORY MAPPING

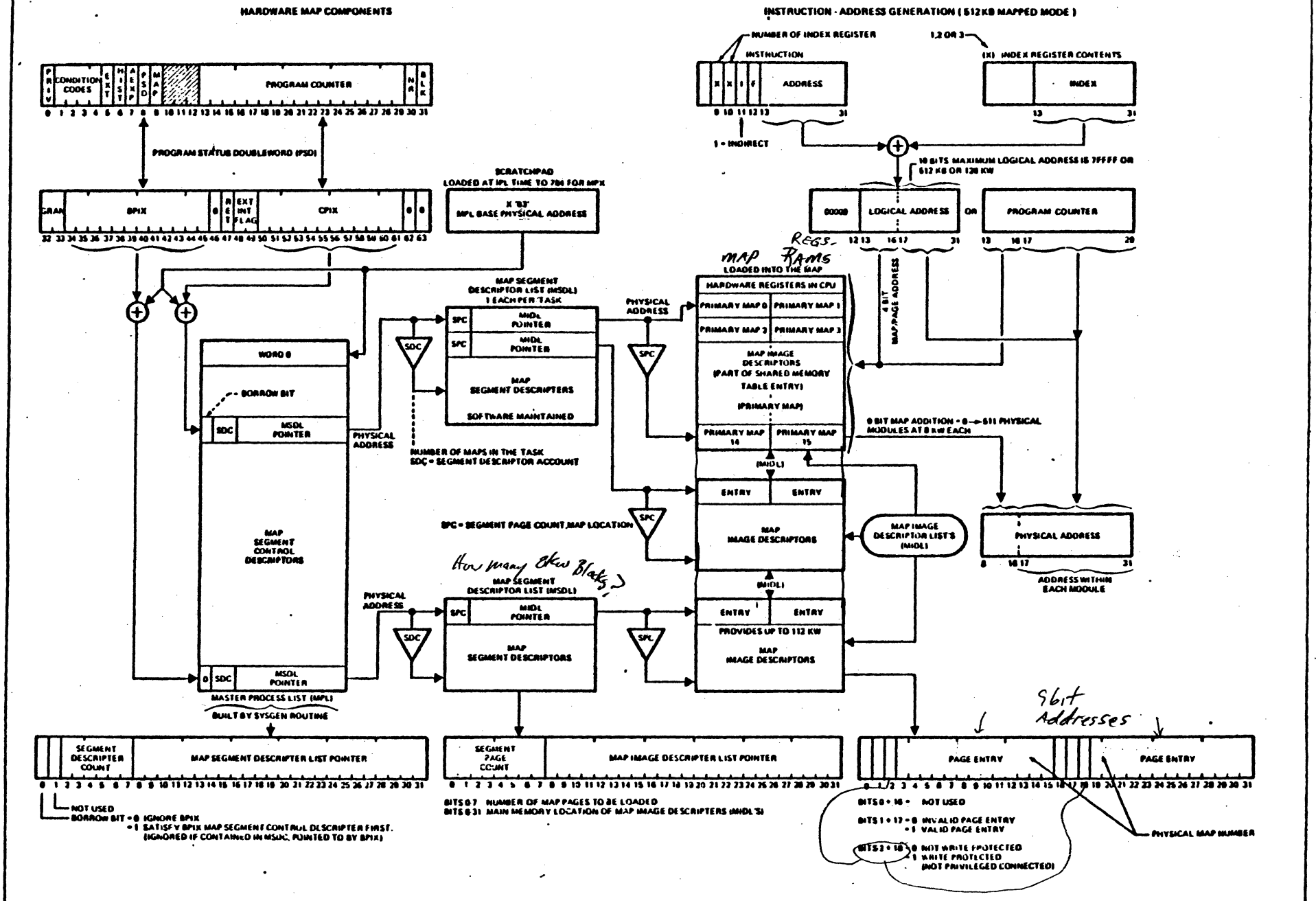
# 512 KB MAPPED MODE



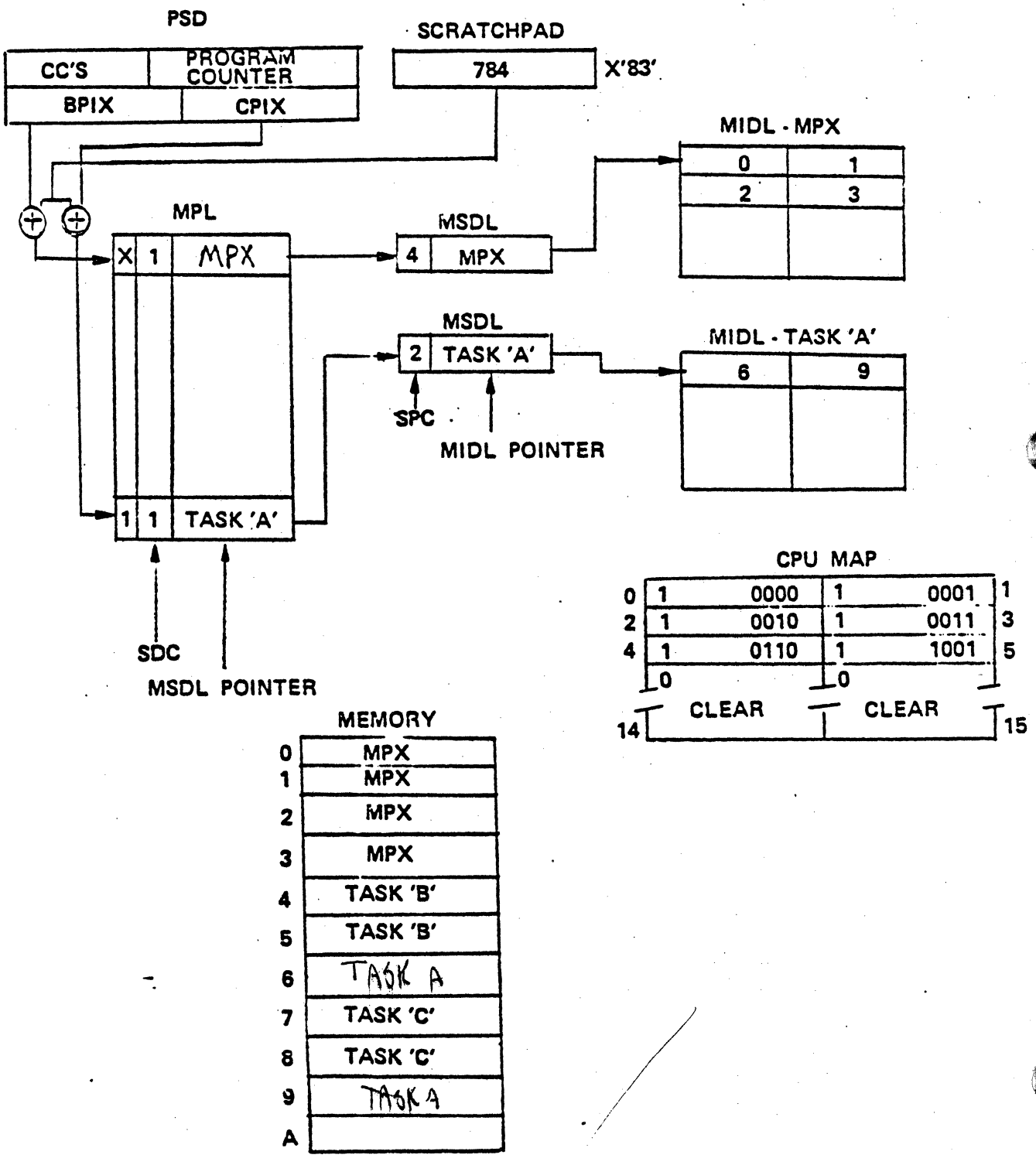
# MAPPED EXTENDED MODE



# MAPPING COMPONENTS



# MAPPING EXAMPLE



## EXPLANATION OF MAPPING EXAMPLE

ASSUME A NEED TO LOAD INTO MEMORY, FROM THE DISC, A TASK CALLED "A", REQUIRING 16K OF CORE. MPX DETERMINES THAT MODULES 6 AND 9 ARE AVAILABLE, SO IT BUILDS SOFTWARE TABLES, NAMELY, A MAP SEGMENT DESCRIPTOR LIST (MSDL) FOR TASK "A", SPECIFYING A TWO PAGE MAP IMAGE DESCRIPTOR LIST (MIDL); A MIDL, SPECIFYING MODULES 6 AND 9; AND A PROGRAM STATUS DOUBLEWORD (PSD) WHICH WILL POINT TO TASK "A" AND BE USED TO LOAD THE MAP.

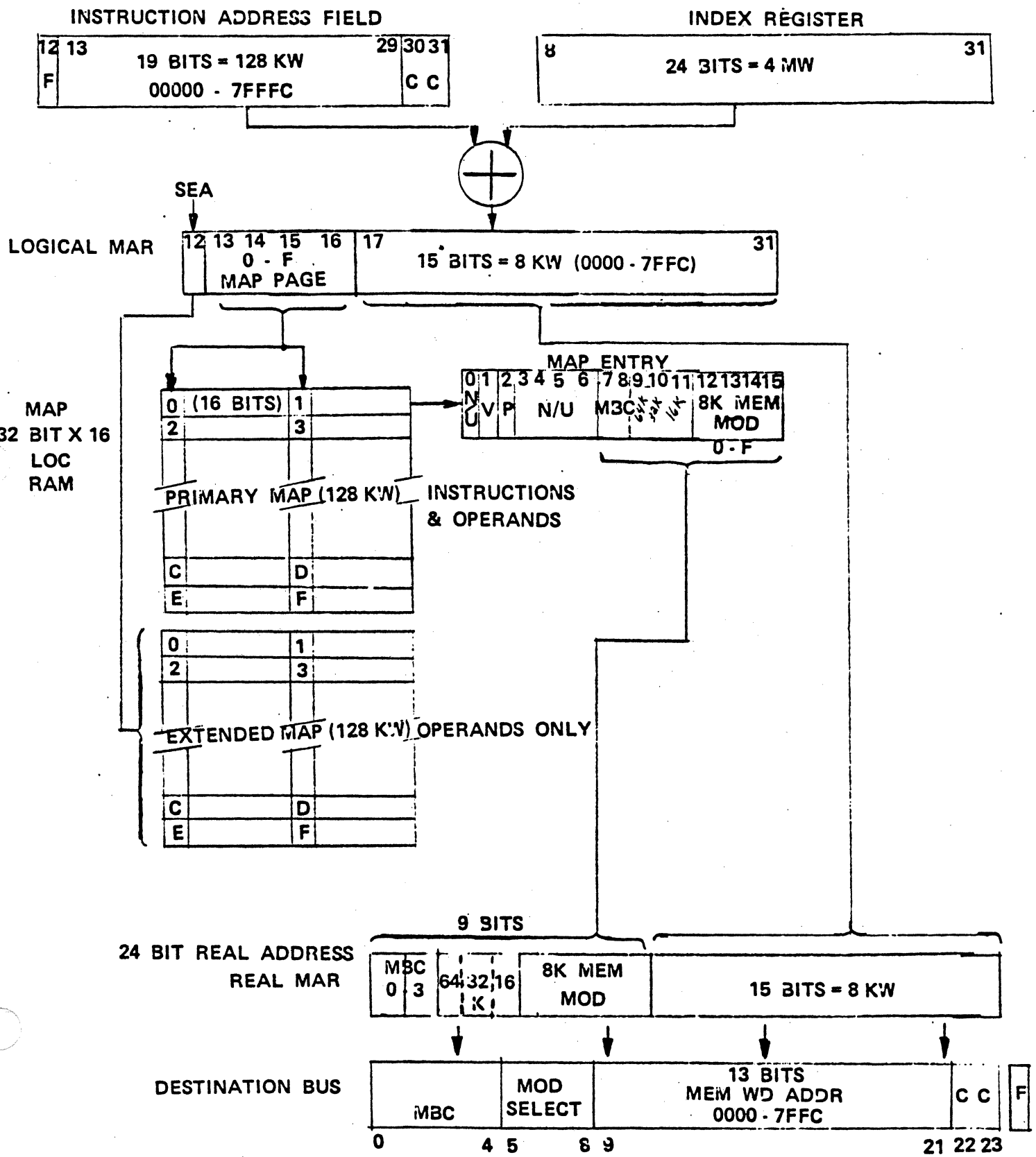
MPX NOW EXECUTES A "LOAD PROGRAM STATUS DOUBLEWORD AND CHANGE MAP" (LPSDCM) INSTRUCTION. FIRMWARE INDEXES INTO THE MASTER PROCESS LIST (MPL), ADDING THE CONTENTS OF SCRATCHPAD LOCATION 83 TO THE PSD's CPIX. IT FINDS THE "BORROW BIT" SET IN THE ENTRY FOR TASK "A" (ALL TASKS ON DISC ARE INCLUDED IN THE MPL), SO FIRMWARE RETURNS AND INDEXES AGAIN, THIS TIME USING THE PSD's BPIX, WHICH IS ALWAYS ZERO.

THE FIRST ENTRY IN THE MPL IS REFERENCED. THIS WILL BE A MAP SEGMENT CONTROL DESCRIPTOR (MSCD), POINTING TO ONE MAP SEGMENT DESCRIPTOR (MSD). THIS MSD SPECIFIES FOUR PAGES TO BE LOADED INTO THE CPU MAP FROM THE MPX MIDL IN MEMORY. THE MAP IS LOADED, PAGE BY PAGE, DECREMENTING THE PAGE COUNT UNTIL ALL FOUR PAGES ARE LOADED.

FIRMWARE NOW INDEXES AGAIN, USING THE CPIX. NOW TASK "A"s MSCD IS REFERENCED, WHICH LOADS PAGES 4 AND 5 OF THE MAP FROM TASK "A"s MIDL, WHICH WAS JUST CREATED.

THE REMAINING MAP PAGE ENTRIES HAVE THEIR VALID BITS RESET. TASK "A" NOW GOES INTO EXECUTION.

# MAPPING ADDRESS TRANSLATION



BUILDING CPU's RAM MIDL SEQUENCE
USING PAGES 4-10 and 4-11 (FIG.S
4-2 and 4-3) of the REFERENCE MANUAL

I. LOAD NEW PSD

II. CPIX AND BASE ADDRESS FROM SCRATCH PAD ADDRESS MAP SEGMENT CONTROL DESCRIPTORS (MSCD)

III. CHECK BORROW BIT

A. IF RESET:

(1) ADDRESS MAP SEGMENT DESCRIPTOR (MSD)

B. IF SET:

(1) USE BPIX AND BASE ADDRESS TO RE-ADDRESS MSCD

IV. ASSUME FOR EXPLANATION PURPOSES THAT THE BORROW BIT WAS SET.

V. MSCD POINTS TO MAP SEGMENT DESCRIPTOR (MSD)

A. SDC COUNTER IS LOADED WITH COUNT FROM MSCD

(1) ASSUME COUNT OF 2

VI. MSCD ADDRESS MSD

VII. MSD POINTS TO MAP IMAGE DESCRIPTOR IN MAIN MEMORY

A. SPC COUNTER LOADED WITH COUNT FROM MSD

(1) ASSUME A COUNT OF 2

VIII. FIRMWARE LOADS 16-BIT PAGE ENTRY INTO THE CPU'S MAP RAM.

IX. SPC DECRIMENTED BY 1

A. COUNT IN SPC IS NOW 1

X. FIRMWARE NOW LOADS SECOND 16-BIT PAGE ENTRY INTO ITS OWN RAM

A. COUNT IN SPC IS NOW 0

XI. SDC COUNTER IS DECREMENTED BY 1

A. COUNT IS NOW 1

B. NEW MSD IS USED TO ADDRESS MID

C. SPC COUNTER NOW LOADED WITH NEW COUNT

(1) ASSUME COUNT OF 1

XII. NEW MID IS LOADED INTO CPU's RAM

XIII. SPC COUNTER IS DECREMENTED BY 1

A. SPC COUNTER EQUALS 0

XIV. SDC COUNTER DECREMENTED BY 1

A. COUNTER NOW EQUAL TO 0

XV. SINCE BORROW BIT WAS SET, FIRMWARE NOW GOES BACK TO THE MSD.

XVI. CPIX AND BASE ADDRESS ARE NOW USED TO ADDRESS MSCD

A. BORROW BIT IS IGNORED AT THIS TIME

XVII. SEQUENCE OF ADDRESSING MSD, MID, AND LOADING SDC/SPC COUNTERS BEGINS AGAIN.

A. FIRMWARE CONTINUES TO BUILD CPU's RAM MIDL UNTIL BOTH SPC AND SDC ARE EQUAL TO 0

XVIII. FIRMWARE NOW SIGNALS CPU IT HAS BUILT MIDL PROGRAM INTO RAM

XIX. CPU INITIATES PROGRAM

NOTE: IF PROGRAM CALLS FOR CHANGING ANY MAP REGISTER, ALL REGISTER PRECEDING REGISTER TO BE CHANGED WILL ALSO BE CHANGED. THIS CALLS FOR RE-INITIALIZING FIRMWARE SEQUENCING JUST DISCUSSED.

EXP.: YOU WISH TO CHANGE PRIMARY MAP REGISTER 4.

FIRMWARE WILL CHANGE REGISTER 0,1,2,3,AND 4.

# System 32/75 Mapping Lab Exercise

Introduction:

The purpose of this worksession is to familiarize the student with the mapping instructions used by the 32/75.

The student will be given the information to enter into memory and registers in order to execute the instruction. The student will write down the results observed after execution.

Reference:

SYSTEMS 32/75 Reference Manual

Procedure:

1. Enter a set extended address (Sea) instruction in hexadecimal into memory as follows:

| Location | Data | Comments |
|---|---|---|
| 00000 | 000D0002 | Sea Instruction |
| Set P.C. to 00000 | | |
| Press Instruction Step | | |
| After Execution | | |
| PSD1 = | | |

2. Enter a "Load Map" (LM) instruction and execute as follows:

| Location | Data | Comments |
|---|---|---|
| 00004 | 80000000 | PSD2 Map Mode |
| 00784 | 01002000 | Map Segment Control Descriptor |
| 01000 | 2C070000 | Load Map Instruction |
| 02000 | 02003000 | Map Segment Descriptor |
| 03000 | FFFFFFFF | Map Image Descriptor |

Set P.C. to 01000

Press Instruction Step

Note: You have just set up the 1st two page entries in the map register. DO NOT press system reset. In order to prove it was loaded the map register entries must be read back into a GPR. Execution of the next step will read it back.

3. Enter a "Read Map" in order to verify the previous step.

| Location | Data | Comments |
|---|---|---|
| 01004 | 2CAA0002 | Transfer Map to Register Instruction |

Set P.C. to 01004

Press Instruction Step

GPR1 =

Why does GPR1 appear as it does?

4. System reset and re-execute the "Transfer Map to Register" instruction.

GPR1 =

Why does GPR1 appear as it does?

5. Enter a "Load Program Status Doubleword" (LPSD) instruction and execute as follows:

| Location | Data | Comments |
|---|---|---|
| 00008 | F9800010 | LPSD Instruction |
| 00010 | FFC7FFFF | Data to PSD1 |
| 00014 | 80000000 | Data to PSD2 |

Set P.C. to 00008

Press Instruction Step

PSD1 =

Why does PSD1 appear as it does?

6. Enter and execute a program that will load all CPU map registers with the following pattern.

| Location | Data | Location | Data |
|---|---|---|---|
| 08000 | 00000001 | 08020 | 00160017 |
| 08004 | 00020003 | 08024 | 00180019 |
| 08008 | 00040005 | 08028 | 00200021 |
| 0800C | 00060007 | 0802C | 00220023 |
| 08010 | 00080009 | 08030 | 00240025 |
| 08014 | 00100011 | 08034 | 00260027 |
| 08018 | 00000001 | 08038 | 00280029 |
| 0801C | 00140015 | 0803C | 00300031 |

Enter program as follows:

| Location | Data | Comments |
|---|---|---|
| 00784 | 01002000 | Map Segment Control Descriptor |
| 02000 | 20008000 | Map Segment Descriptor |
| 02004 | 2C070000 | Load Map Instruction |
| 02008 | 00000000 | PSD1 |
| 0200C | 80000000 | PSD2 |

Enter 00002008 into GPR 0.

Set P.C. to 02004

Press Instruction Step

Note: The program just loaded the data pattern into the CPU map registers. DO NOT press system reset. In order to look at the data pattern in the map registers unload it into memory with the next step.

7. The following program must be entered to unload the CPU map registers into memory starting at location 03000. DO NOT system reset.

| Location | Data | Comments |
|---|---|---|
| 02100 | C980FFC0 | Load a Neg. Index into R3 |
| 02104 | 2CAA0002 | Transfer map to GPR1 |
| 02108 | D4E03040 | Store GPR1 at Loc. 3000 |
| 0210C | 23230002 | Add +1 to bit 30 in GPR2 |
| 02110 | F5C02104 | Incr. GPR3 if ≠ branch |
| 02114 | 00000000 | Halt (finished) |

Set P.C. to 02100

Press Run

Enter below what locations 03000 to 0303C contain

03000 =

DAY 8

# MEMORY SYSTEM AND MEMORY MAPPING - WORKSESSION

REFERENCES:

32/70 SERIES
REFERENCE MANUAL
SECTION IV

TECHNICAL MANUAL
SECTION 1, III

WORKBOOK

Objectives: After completing the referenced reading sections of the Reference and Technical Manual, and completing the general information questions of this worksession; the student will be capable of:

1. Describing the functions of the MBC.

2. Identifying the various circuit units of the MBC and describe their purpose.

1. One Memory Bus Controller (MBC) can handle up to 16 memory modules.

2. What is contained in the Source Address Buffers on an MRT? ~~Contain addresses of memory locations referenced by the Memory Read Transfer~~ Physical Address of Requesting Device (CPU/IOP)

3. When writing or reading from memory (8K memory modules), what do address bits 13 thru 16 define? They define the module to be selected

4. Does an MBC have a SEL Bus priority? Interrupt priority?
SEL Bus Priority = 1
Does not have Interrupt Priority

5. The primary map has _____ page entry locations? The extended map has _____? ~~8K, 16K~~
16, 16

6. How do we gain access to the extended map? ~~Through the PSD mode of operation~~ SEA mapped mode

7. Where in scratchpad is the MPL base address stored?
at X'83'

8. The extended map is used for storage and retrieval of _____ only.
Operands (Data)

9. What instruction is used to put the system into the extended addressing mode? SEA Set Extended Addressing

10. What bit(s) of the PSD set the system into the mapped mode of operation? Bit a indicates Mapped, Unmapped
Bits 32,33

11. What defines the lower and upper address limits of an MBC?
Jumpers on MBC

12. When does the MBC activate its inhibit line? When is inhibit removed? When its buffer is 3/4 full. When buffer is 3/4 empty

13. When is the "LOAD MAP INSTRUCTION" normally used? Why?
For Diagnostics. It loads the Map Image Descriptor List from main memory into the CPU MAP Registers verify Registers are good

14. What instruction is used to perform context switching in the mapped mode? Load Program Status Double word and Change Map

15. If an MBC is improperly jumpered for high address, and attempts to access a non-present module, what error indications will be displayed.

Parity Error Error Light
Halt Light
Non-Present Memory Trap from the Display

Write Stores Parity Error

Read Displays Parity Error

# DAY 9

# SECTION 9

# IOM'S

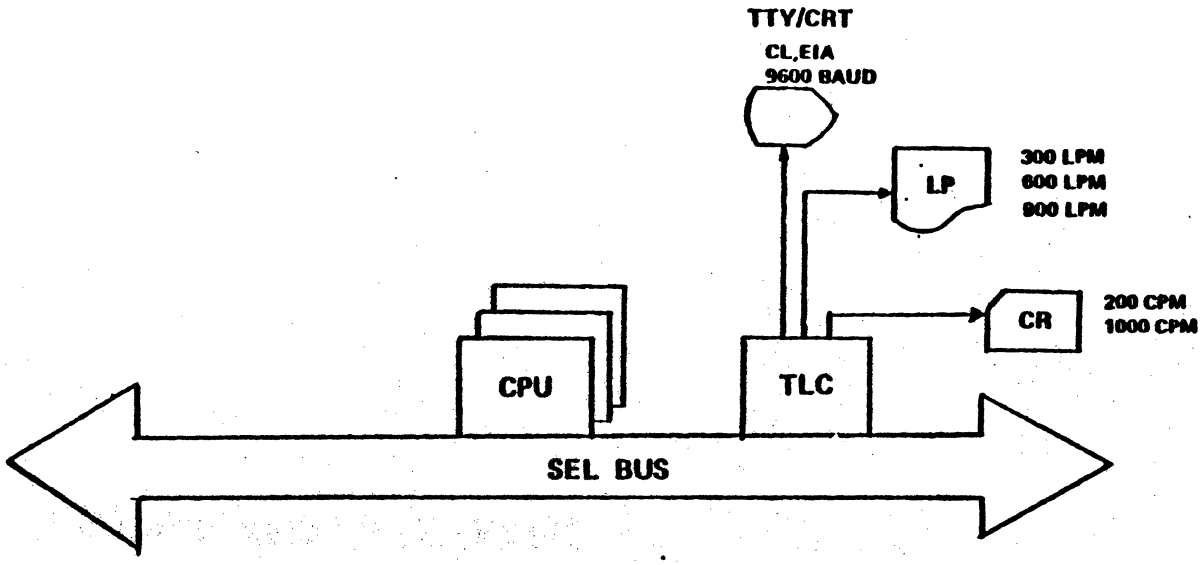
- INPUT/OUTPUT MICROPROGRAMMABLE PROCESSOR (IOM)

  PROVIDES DIRECT COMMUNICATION BETWEEN PERIPHERAL & SEL BUS
  FIRMWARE PROGRAMMABLE I/O CONTROL
  GENERATES I/O SERVICE INTERRUPTS (INTERNAL SI)

- TLC - TTY/LINE PRINTER/CARD READER CONTROLLER

  MULTIPLEXER CHANNEL (ALL 3 UNITS ACTIVE)
  DEDICATED I/O CONNECTOR SLOTS
  3 SEPARATE SERVICE INTERRUPTS

# CONSOLE DEVICES




**TLC CONTROLLER**

- **MODEL 9004 WIRE/WRAP**
- **MODEL 9005 COPPER**

# MAG TAPE SUBSYSTEM

MAG TAPE CONTROLLER - MODEL #9012 (W/W)
#9013 (PC)

* 4 UNITS MAX
* SELECTOR CHANNEL (ONLY 1 UNIT ACTIVE AT A TIME)
* CLASS E - 128 KW ADDRESSING
* 4096 HALFWORD = 8192 BYTE RECORD LENGTH

SUPPORTS PERTEC & KENNEDY:

*IMBEDDED FORMATTER
*45 IPS TENSION
*75 IPS VACUUM
*800 BPI (NRZI)
*1600 BPI (PE)

LOW SPEED TAPE PROCESSOR - MODEL #9020

* 4 UNITS (OPTIONAL 8)
* MULTIPLEXER CHANNEL (ALL 4/8 ACTIVE)
* CLASS 'F' - 16 MB ADDRESSING
* INFINITE RECORD SIZE

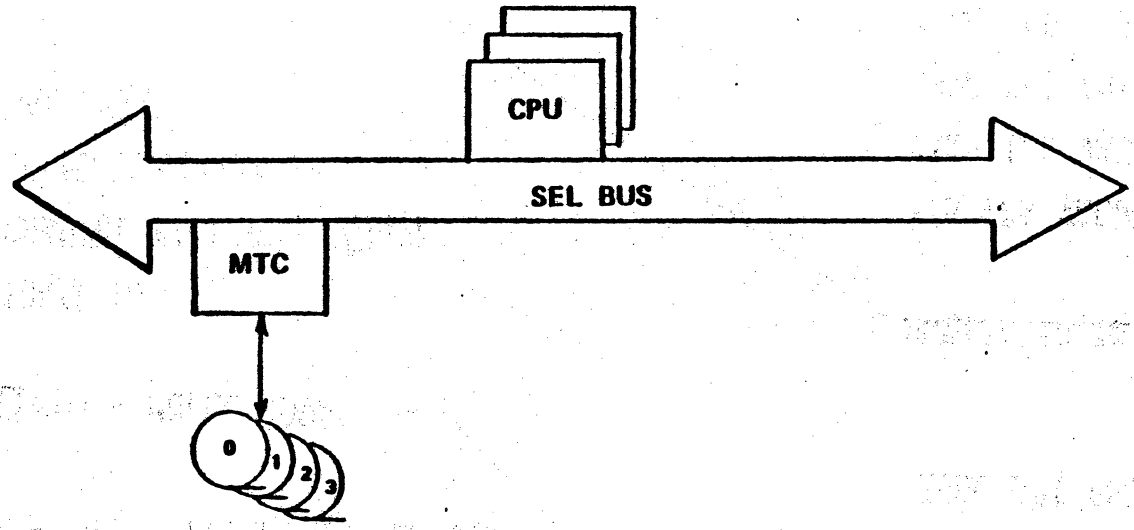
DEDICATED TO STC DRIVES:

*75 IPS VACUUM
*125 IPS VACUUM
*800 BPI (NRZI)
*1600 BPI (PE) 1 BIT CORRECT
*6250 BPI (GCR) 2 BIT CORRECT
* AUTO LOAD & UNLOAD
* AUTO & MANUAL DENSITY SELECT

HIGH SPEED TAPE PROCESSOR - MODEL #8050

* 4 UNITS (OPTIONAL 8)
* MULTIPLEXER CHANNEL
* CLASS 'F' - 16 MB ADDRESSING
* INFINITE RECORD SIZE

# MAG TAPE SUBSYSTEM




MAG TAPE CONTROLLER - MODEL # 9012 (W/W)
# 9013 (PC)

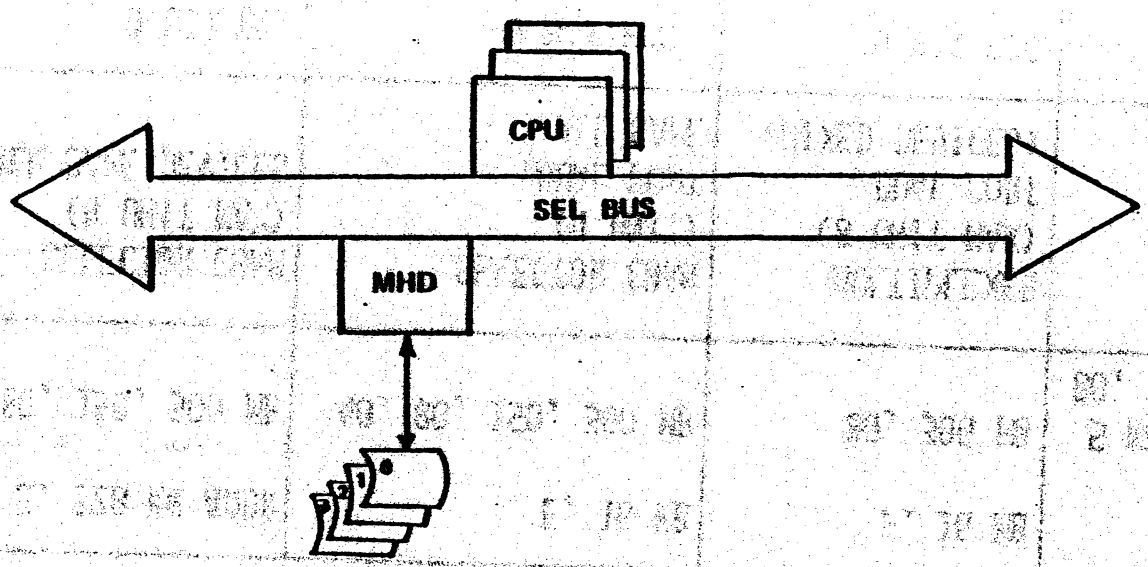
LOW SPEED TAPE PROCESSOR - MODEL # 9020

HIGH SPEED TAPE PROCESSOR - MODEL # 8050

# DISC SUBSYSTEM

| MODEL #: | 9010 (W/W) | 73-9032 (W/W) | 9024 DISC PROCESSOR | 8055 DP II |
|---|---|---|---|---|
| CLASS: | E, 128 KW ADDR | E, 16 MB | F, 16 MB | F, 16 MB |
| DEVICE TYPE: | 40, 80, 150, 300 MB | 40, 80, 150, 300 MB | 80, 300 MB | 5 MB FHD, 32 MB CMD<br>80, 300 MHD, 600 FMD |
| CHANNEL OPERATION: | SELECTOR CHAN<br>(4 UNIT MAX)<br>ALL SAME DEVICES | SELECTOR CHAN<br>(4 UNIT)<br>DUAL PORT<br>ALL SAME | MULTIPLEXER<br>(8 UNIT MAX)<br>DUAL PORT<br>(MIXED DEVICES) | MULTIPLEXER<br>(8 UNIT MAX)<br>DUAL PORT<br>(MIXED DEVICES) |
| ERROR CORRECTION: | 9 BIT ECC | 9 BIT ECC | 9 BIT ECC | 9 BIT ECC |

# DISC SUBSYSTEM




MHD CONTROLLER - MODEL # 9010 (W/W)
# 73-9032 (W/W)

DISC PROCESSOR - MODEL # 9024

DISC PROCESSOR II - MODEL # 8055

# ICL DEVICE ENTRY FORMAT

Format #1 *DEVXX=FCILCASA (,NN)

where:

*DEV defines that the record contains a controller definition entry.

XX is the hexadecimal address that will be used by macro level input/output instructions to address the controller.

= is a necessary delimiter. Each letter to the right of this delimiter represents one hexadecimal digit (four binary bits).

F flags used by the CPU for input/output emulation. Presently, this field is always zero.

C defines the class of controller being emulated. Presently, this field can contain one of the following values:

0 = LINE PRINTER
1 = CARD READER
2 = TELETYPE
3 = INTERVAL TIMER
4 = PANEL
5 to D = Unassigned
E = ALL OTHERS
F = EXTENDED I/O (32/75 ONLY)

IL is the hexadecimal interrupt level of the Service Interrupt (i.e., priority levels $14_{16}$ through $23_{16}$) for the defined controller.

CA is the hexadecimal controller address as defined by the hardware switches on the IOM.

SA is the lowest hexadecimal device subaddress used by the controller. This field is normally zero when more than one device is configured.

( ) denotes optional parameters.

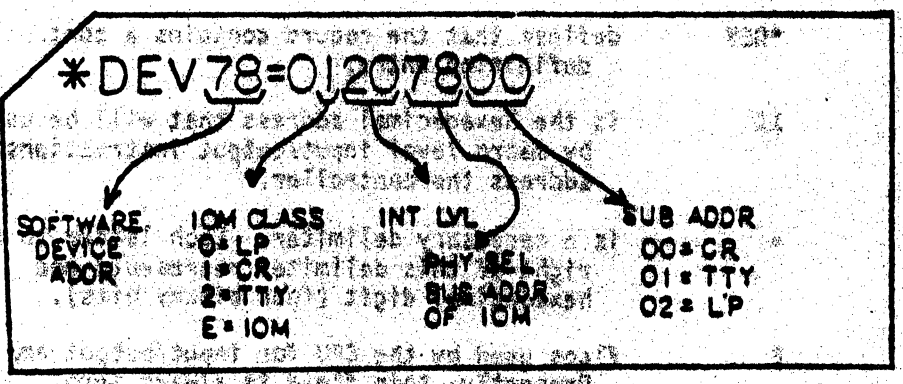
, is a delimiter that must be used when more than one device is configured.

NN is a 2-digit hexadecimal number that specifies the number of devices configured on the controller.

NOTE 1: The subaddress (SA) field must reflect the following for the Teletype, Line Printer, Card Reader (TLC) controller:

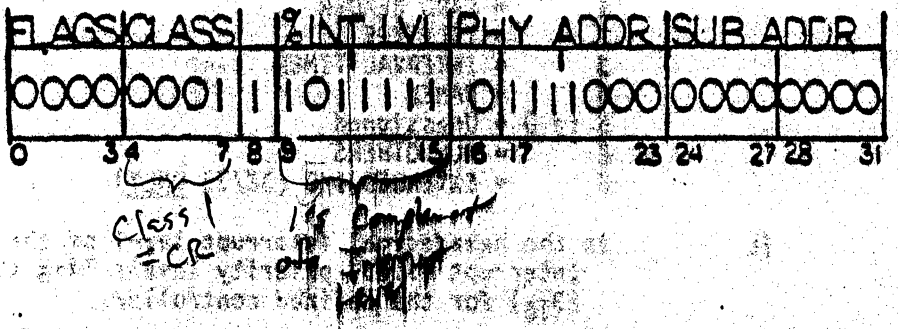
1. Card Reader is subaddress $0_{16}$.
2. Teletype is subaddress $1_{16}$.
3. Line Printer is subaddress $2_{16}$.

# DEVICE ENTRY CARD



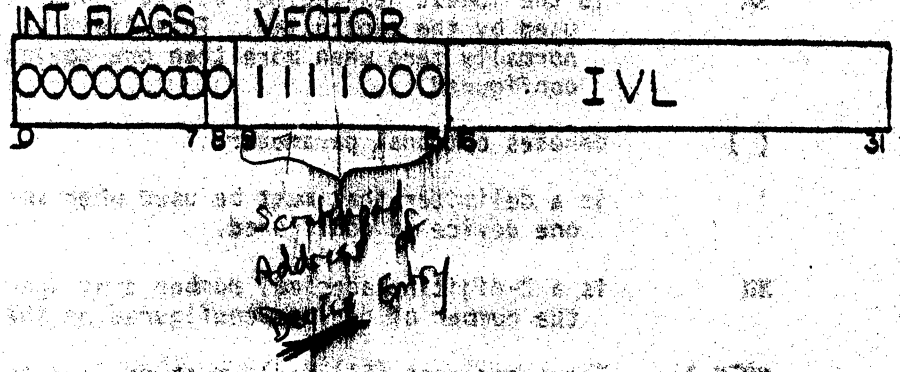

SCRATCHPAD DEVICE ENTRY

LOC 78 IN SCRATCHPAD

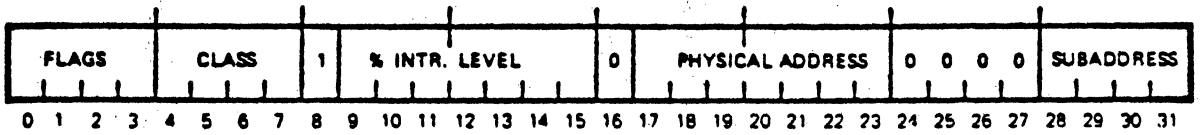



SCRATCHPAD INTERRUPT ENTRY

LOC AØ IN SCRATCHPAD



# SCRATCHPAD DEVICE ENTRY




| FLAGS | |
|---|---|
| BIT | DEFINITION |
| 00 | PROGRAM VIOLATION |
| 01 | NOT ASSIGNED |
| 02 | NOT ASSIGNED |
| 03 | NOT ASSIGNED |

| CLASS | | | | |
|---|---|---|---|---|
| BITS | | | | DEFINITION |
| 04 | 05 | 06 | 07 | |
| 0 | 0 | 0 | 0 | CLASS 0 - 'TLC' LINE PRINTER |
| 0 | 0 | 0 | 1 | CLASS 1 - 'TLC' CARD READER |
| 0 | 0 | 1 | 0 | CLASS 2 - 'TLC' TELETYPEWRITER |
| 0 | 0 | 1 | 1 | CLASS 3 - RTOM INTERVAL TIMER |
| 1 | 1 | 1 | 0 | CLASS 'E' STANDARD I/O CONTROLLERS |
| 1 | 1 | 1 | 1 | CLASS 'F' I/O CONTROLLER |

| INTERRUPT LEVEL |
|---|
| BIT 08 = 1<br>BITS 09-15 PROVIDE THE ONES COMPLEMENT OF THE I/O CONTROLLER INTERRUPT LEVEL |

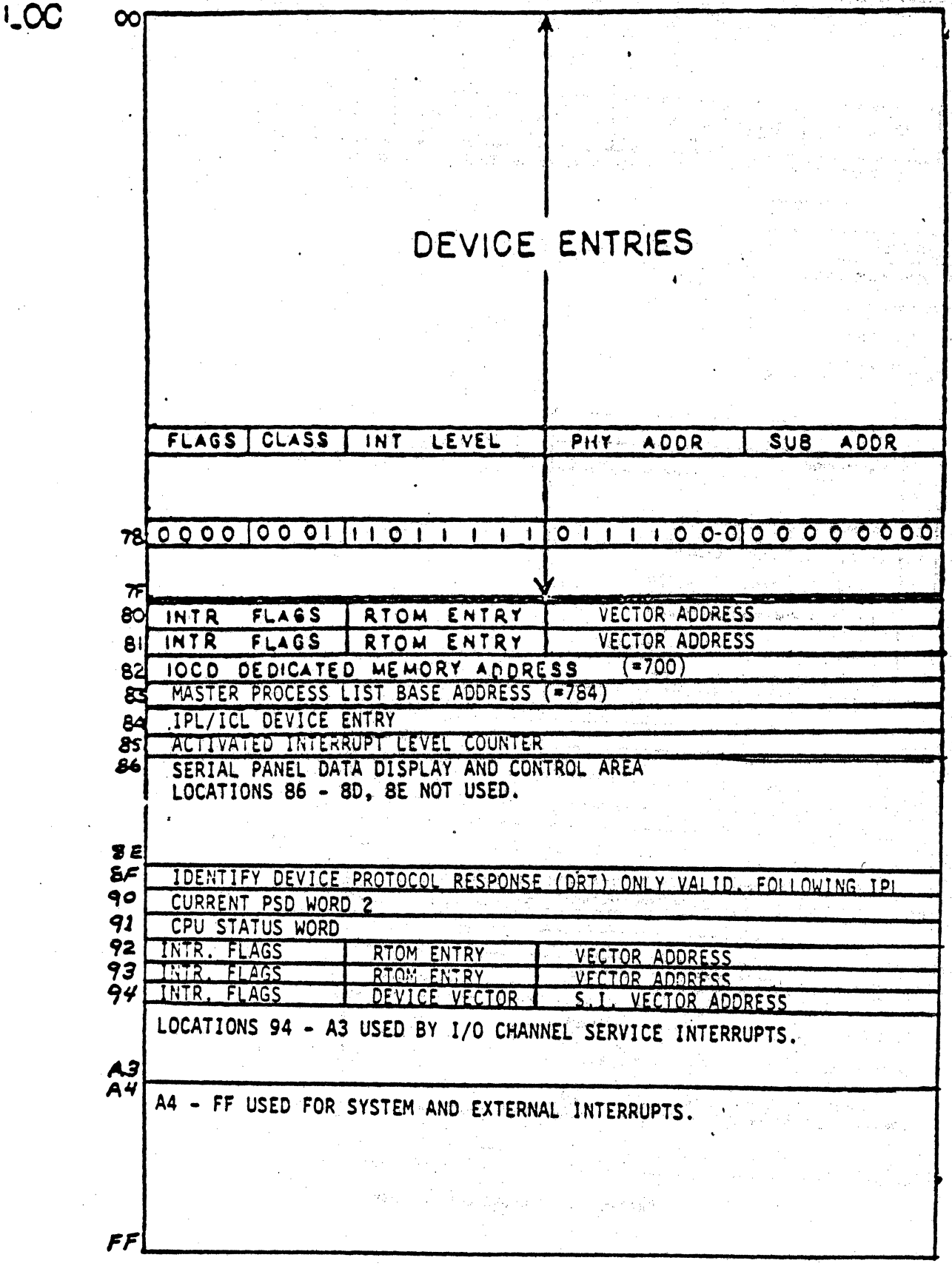
| PHYSICAL ADDRESS |
|---|
| BIT 16 = 0<br>BITS 17 - 23 PROVIDE THE I/O CONTROLLER (IOM) PHYSICAL ADDRESS, WHICH MUST MATCH THE IOM'S ADDRESS SWITCH CONFIGURATION |

| SUBADDRESS |
|---|
| BITS 28-31 PROVIDE THE SUBADDRESS OF SPECIFIC I/O DEVICE |

NOTES:
1. DEVICE ENTRIES ARE IN SCRATCHPAD ADDRESSES 00-7F.
2. CORRESPONDS TO THE SCRATCHPAD ADDRESS.
3. DEVICE ENTRY SCRATCHPAD ADDRESS CAN ALSO BE OBTAINED FROM THE INTERRUPT TABLE ENTRY, VECTOR FIELD (REFER TO FIGURE 5-50).

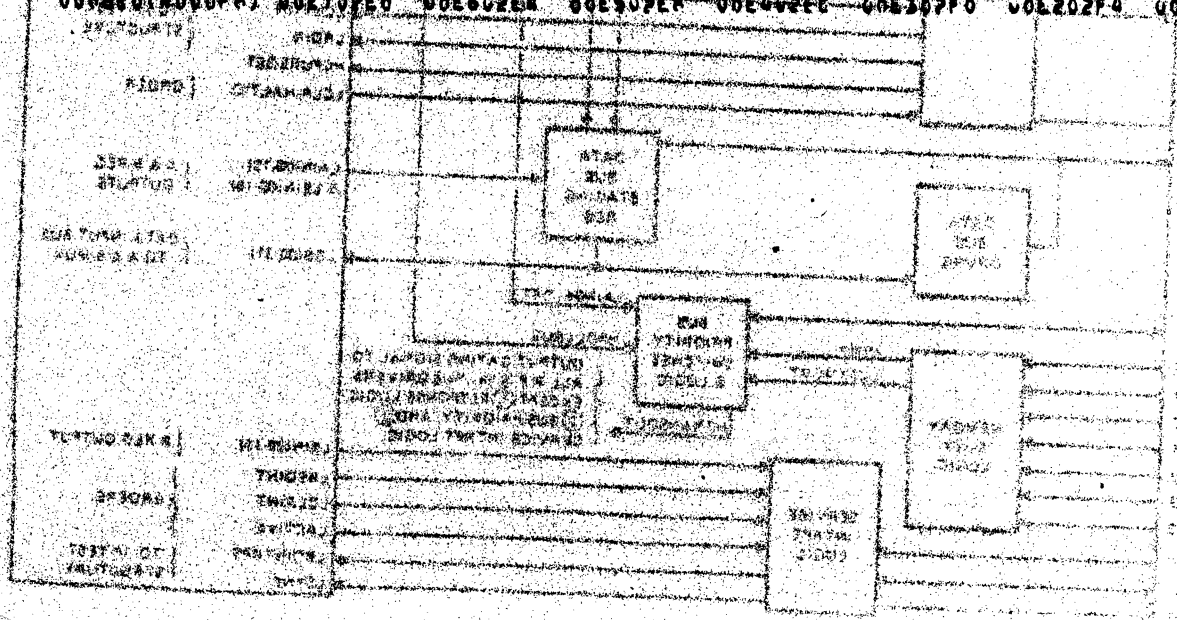
Format - Scratchpad Device Entry

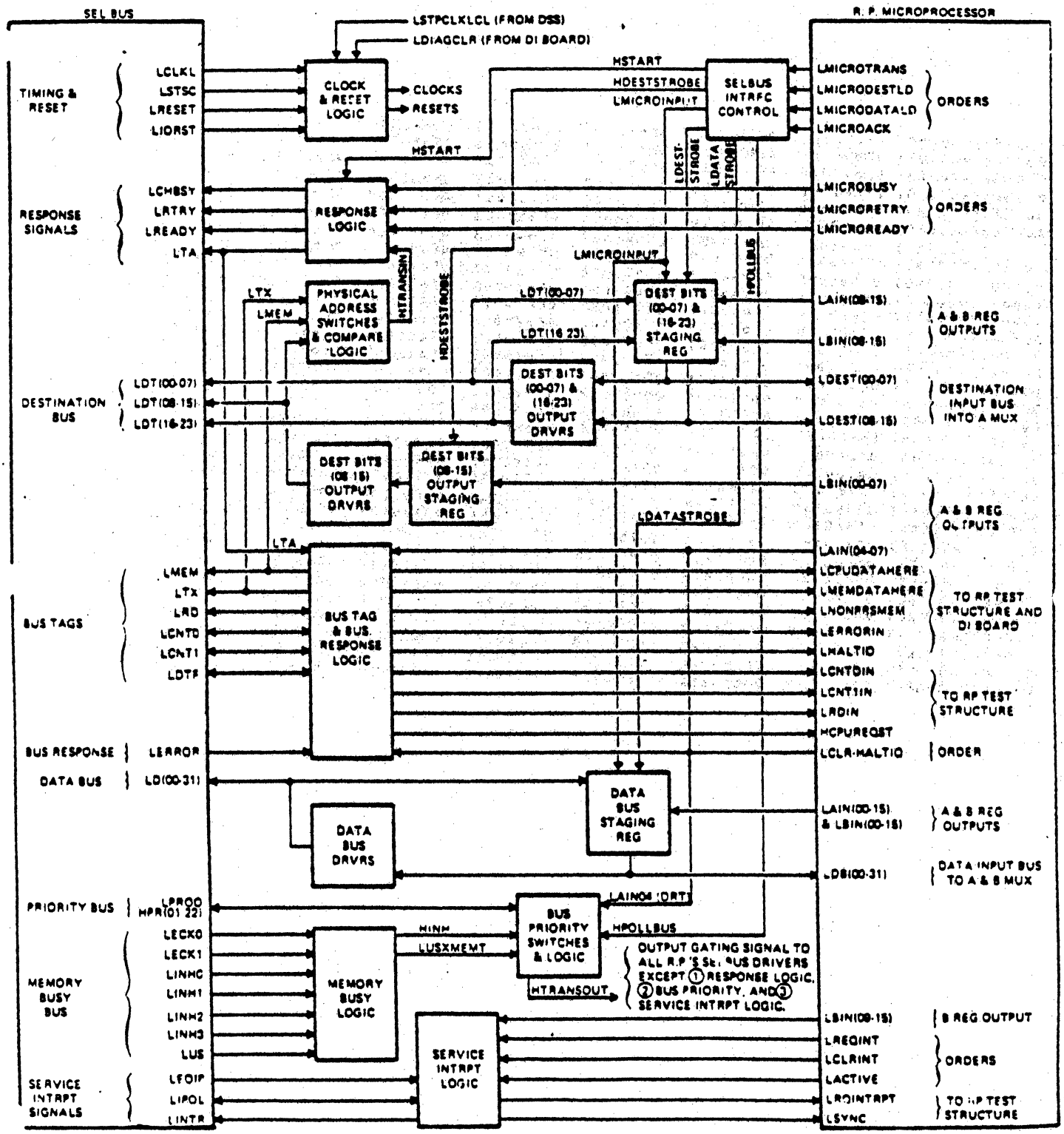
# SEL 32 SCRATCHPAD LAYOUT



# SCRATCHPAD DUMP

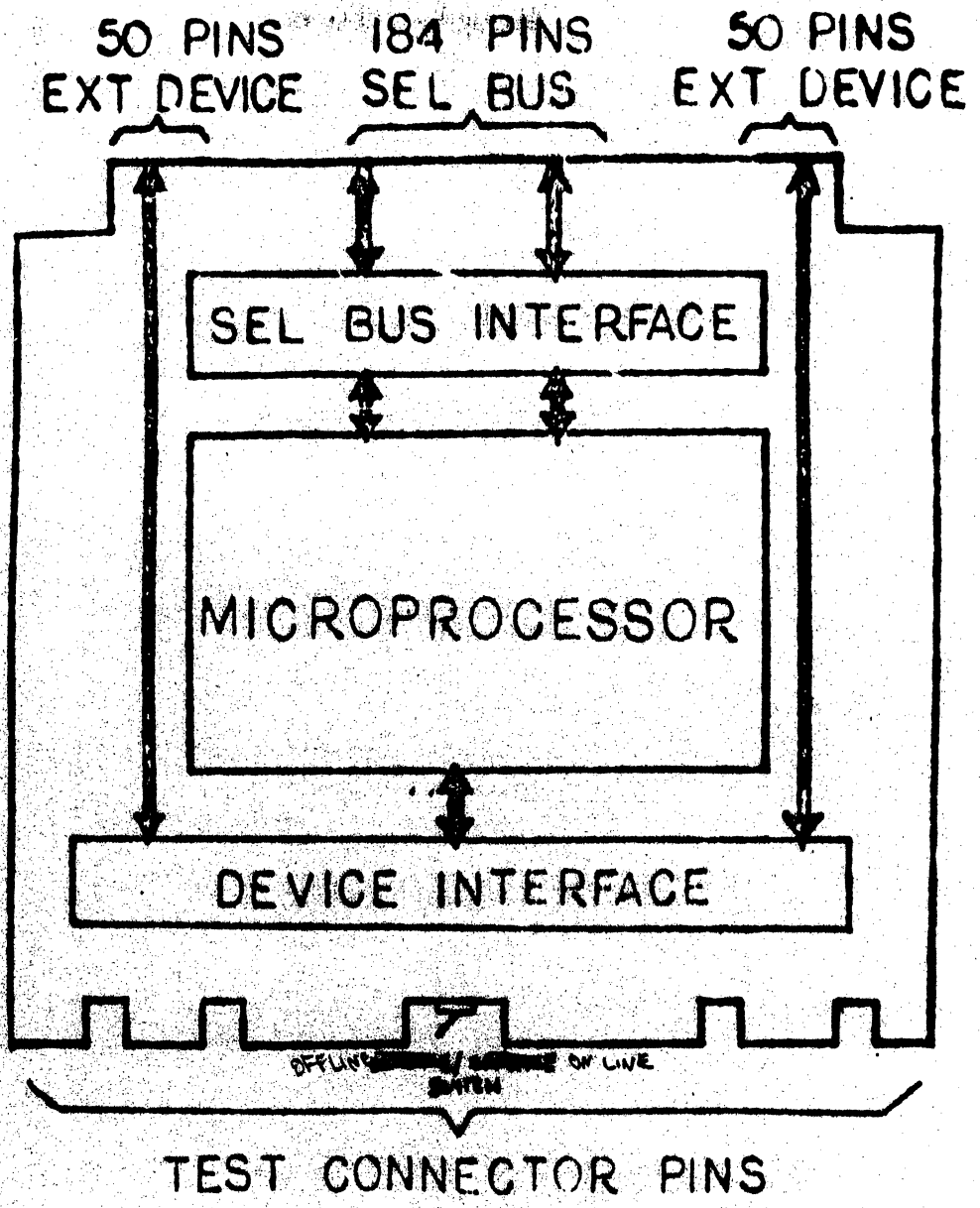
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 000300(H00000) | 0E6B01U0 | 0E6B0101 | 0E6B0102 | 0E6B0103 | 0E6A0400 | 0E6A0401 | 0E6A0402 | 0E6A0403 |
| 000320(H00008) | 0F6B0A00 | 0E6B0B01 | 0E6B0B02 | 0E6B0B03 | 0E6A0C00 | 0E6B0C01 | 0E6B0C02 | 0E680C03 |
| 000340(H00010) | 0E671000 | 0E671001 | 0E671002 | 0E671003 | 0E671004 | 0E671005 | 0E671006 | 0E671007 |
| 000360(H00018) | 0E661800 | 0E661801 | 0E661802 | 0E661803 | 0E661804 | 0E661805 | 0E661806 | 0E661807 |
| 000380(H00020) | 0E652000 | 0E652001 | 0E652002 | 0E652003 | 0E652004 | 0E652005 | 0E652006 | 0E652007 |
| 0003A0(H00028) | 0E65200A | 0E652009 | 0E65200A | 0E65200B | 0E65200C | 0E65200D | 0E65200E | 0E65200F |
| 0003C0(H00030) | 0E643000 | 0E643001 | 0E643002 | 0E643003 | 0F643004 | 0E643005 | 0E643006 | 0E643007 |
| 0003E0(H00038) | 0E643008 | 0E643009 | 0E64300A | 0E64300A | 0E64300C | 0E64300D | 0E64300E | 0E64300F |
| 000400(H00040) | 0E634000 | 0E634001 | 0E634002 | 0F634003 | 0E634004 | 0F634005 | 0E634006 | 0E634007 |
| 000420(H00048) | 0F6340UA | 0E634009 | 0E63400A | 0E63400A | 0E63400C | 0E63400D | 0F6340UE | 0E63400F |
| 000440(H00050) | 0E625000 | 0F625001 | 0E625002 | 0F625003 | 0F625004 | 0E625005 | 0E625006 | 0E625007 |
| 000460(H00058) | 0E62500A | 0E625009 | 0E62500A | 0F62500A | 0F62500C | 0E62500D | 0E62500E | 0E62500F |
| 000480(H00060) | 0E616000 | 0F616001 | 0F616002 | 0E616003 | 0F616004 | 0E616005 | 0E616006 | 0E616007 |
| 0004A0(H00068) | 0E61600A | 0E61600A | 0E61600A | 0E61600B | 0E61600C | 0E61600D | 0E61600E | 0E61600F |
| 0004C0(H00070) | 0E607000 | 0E607001 | 0F607002 | 0E607003 | 0F607004 | 0E607005 | 0E607006 | 0E607007 |
| 0004E0(H00078) | 015F7A00 | 00000000 | 005E7B0F | 00000000 | 005D7C02 | 005D7C03 | 025C7801 | 03547904 |
| 000500(H00080) | 0009F00E0 | 00F600F8 | 00000700 | 00000784 | 00U010U0 | 00000000 | 00000000 | BFFFFFFF |
| 000520(H00088) | BFFFFFFF | BFFFFFFF | BFFFFFFF | BFFFFFFF | 80FFFFFC | 00U00000 | 00000000 | 00000000 |
| 000540(H00090) | 00000000 | 00000001 | 009D00E8 | 00FC00EC | 00000140 | 00040144 | 000A0148 | 000C014C |
| 000560(H00098) | 00100150 | 00180154 | 00200158 | 00300150 | 00400160 | 00500164 | 00600168 | 0070016C |
| 000580(H000A0) | 007801T0 | C17A0170 | 007C0170 | C17E017C | 00980190 | 009A0194 | 00990198 | 0098019C |
| 0005A0(H000A8) | 00960140 | 00970140 | 00950140 | 00940140 | 00950180 | 00920184 | 00910188 | 0090018C |
| 0005C0(H000B0) | 00AF01C0 | 00AE01C4 | 00AD01C8 | 00AC01CC | 00AB01D0 | 00A401D4 | 00A901D8 | 00AA01DC |
| 0005E0(H000B8) | 00A701E0 | 00A601E4 | 00A501E8 | 00A401EC | 00A301F0 | 00A201F4 | 00A101F8 | 00A001FC |
| 000600(H000C0) | 00HE0200 | 00HF0204 | 00HD0208 | 00HC020C | 00HB0210 | 00HA0214 | 00H90218 | 00H8021C |
| 000620(H000C8) | 00070220 | 00080224 | 00050228 | 00040220 | 00030230 | 00020234 | 00010238 | 0000023C |
| 000640(H000D0) | 00LF0240 | 00CE0244 | 00CD0248 | 00CC024C | 00LB0250 | 00CA0254 | 00C90258 | 00C8025C |
| 000660(H000D8) | 00C70260 | 00C60264 | 00C50268 | 00C4026C | 00L30270 | 00C20274 | 00C10278 | 00C0027C |
| 000680(H000E0) | 00DF0280 | 00DE0284 | 00DD0288 | 00DC028C | 00DB0290 | 00DA0294 | 00D90298 | 00D8029C |
| 0006A0(H000E8) | 00D702A0 | 00D602A4 | 00D502A8 | 00D402AC | 00D302B0 | 00D202B4 | 00D102B8 | 00D002BC |
| 0006C0(H000F0) | 00CF02C0 | 00CE02C4 | 00CD02C8 | 00CC02CC | 00CB02D0 | 00CA02D4 | 00C902D8 | 00C802DC |
| 0006E0(H000F8) | A0A702E0 | 00L602E4 | 00E502E8 | 00E402EC | 00L302F0 | 00L202F4 | 00E102F8 | 00E002FC |






IOM Sel Bus Interface Block Diagram

# IOM


50 PINS
EXT DEVICE
184 PINS
SEL BUS
50 PINS
EXT DEVICE
SEL BUS INTERFACE
MICROPROCESSOR
DEVICE INTERFACE
TEST CONNECTOR PINS

# BOARD JUMPERING

SYSTEMS ENGINEERING LABORATORIES

# TECHNICAL SUPPORT BULLETIN

| TITLE | TSB No. |
|---|---|
| TLC Jumper Configuration | 0063 |
| **Product** | **Model No.** | **Date** |
| TLC | 9004 & 9005 | 11/22/78 |

This TSB contains configuration jumper information for the 9004 TLC and the new Copper 9005 TLC.

To configure transmit priority, one jumper should be installed for the desired priority.

| Transmit Priority | 9004 | 9005 |
|---|---|---|
| 1 | A18 1-16 | X1 1-16 |
| 2 | A18 2-15 | X1 2-15 |
| 3 | A18 3-14 | X1 3-14 |
| 4 | A18 4-13 | X1 4-13 |
| 5 | A18 5-12 | X1 5-12 |
| 6 | A18 6-11 | X1 6-11 |
| 7 | A18 7-10 | X1 7-10 |
| 8 | A18 8-9 | X1 8-9 |
| 9 | A19 1-16 | X2 1-16 |
| 10 | A19 2-15 | X2 2-15 |
| 11 | A19 3-14 | X3 2-14 |
| 12 | A19 4-13 | X2 4-13 |
| 13 | A19 5-12 | X2 5-12 |
| 14 | A19 6-11 | X2 6-11 |
| 15 | A19 7-10 | X2 7-10 |
| 16 | A19 8-9 | X2 8-9 |
| 17 | A20 1-16 | X3 1-16 |
| 18 | A20 2-15 | X3 2-15 |
| 19 | A20 3-14 | X3 3-14 |
| 20 | A20 4-13 | X3 4-13 |
| * 21 | A20 5-12 | X3 5-12 |
| 22 | A20 6-11 | X3 6-11 |

CUSTOMER SERVICES DEPARTMENT
6901 West [illegible] se Blvd., Fort Lauderdale, Fla. 33313

| TITLE | TSB No. |
|---|---|
| TLC Jumper Configuration | 0063 |

To configure receive priority a jumper should be installed for all priorities below the transmit priority.

| Receive Priority | 9004 | 9005 |
|---|---|---|
| 1 | C18 2-15 | X4 2-15 |
| 2 | C18 3-14 | X4 3-14 |
| 3 | C18 4-13 | X4 4-13 |
| 4 | C18 5-12 | X4 5-12 |
| 5 | C18 6-11 | X4 6-11 |
| 6 | C18 7-10 | X4 7-10 |
| 7 | C18 8-9 | X4 8-9 |
| 8 | C19 1-16 | X5 1-16 |
| 9 | C19 2-15 | X5 2-15 |
| 10 | C19 3-14 | X5 3-14 |
| 11 | C19 4-13 | X5 4-13 |
| 12 | C19 5-12 | X5 5-12 |
| 13 | C19 6-11 | X5 6-11 |
| 14 | C19 7-10 | X5 7-10 |
| 15 | C19 8-9 | X5 8-9 |
| 16 | C20 1-16 | X6 1-16 |
| 17 | C20 2-15 | X6 2-15 |
| 18 | C20 3-14 | X6 3-14 |
| 19 | C20 4-13 | X6 4-13 |
| 20 | C20 5-12 | X6 5-12 |
| 21 | C20 6-11 | X6 6-11 |

| TITLE | TSB No. |
|---|---|
| TLC Jumper Configuration | 0063 |

The physical address of the TLC is configured with 7 jumpers as follows:

| Physical Address | 9004 | 9005 |
|---|---|---|
| MSB | E11 1-16 | X7 1 -20 |
| | E11 2-15 | X7 2 -19 |
| | E11 3-14 | X7 3 -18 |
| | E11 4-13 | X7 4 -17 |
| | E11 5-12 | X7 5 -16 |
| | E11 6-11 | X7 6 -15 |
| LSB | E11 7-10 | X7 7 -14 |

The multi-controller controller (MCC) jumper is required for the 9004 TLC but is not used on the 9005.

| | 9004 | 9005 |
|---|---|---|
| MCC | E11 8-9 | NU |

A new function of the 9005 TLC is the ability to read cards in full ASC II mode only, or to TLC in full ASC II and read normal in half ASC II as does the 9004 TLC.

| | Delete | | Add |
|---|---|---|---|
| Enable Full ASC II only | X11 7-10 | O.T → | X11 8-9 |
| Full & Half ASC II | X11 8-9 | IN → | X11 7-10 |
| Like 9004 | | | |

The ability to configure for 1 or 2 stop bits after the parity bit has been incorporated in the 9005. The 9004 is always [illegible]

| | | |
|---|---|---|
| 2 STOP BITS | Generate | X7 9-13 |
| 2 STOP BITS | Delete | X7 9-12 |

The method of selecting EIA or current mode for the console is changed from the 9004 and is as follows:

| Mode | | Insert | Delete |
|---|---|---|---|
| Current | 9004 | N1 4-13 | N1 3-14 |
| Current | 9005 | X7 10-12 | |
| EIA | 9004 | N1 13-14 | N1 4-13 |
| EIA | 9005 | | X7 10-11 |

| TITLE | TSB No. |
|---|---|
| TLC Jumper Configuration | 0063 |

The baud rate for the console device is selected as follows:

9005

| BAUD RATE | HEX CODE X8 X9 X10 | X8 1234 | X8 5678 | X9 1234 | X9 5678 | X10 1234 | X10 5678 |
|---|---|---|---|---|---|---|---|
| 110 | 10 17 2C | 0001 | 000X | 0001 | 0111 | X01X | 11XX |
| 150 | 10 67 24 | 0001 | 000X | 0110 | 0111 | X01X | 01XX |
| 300 | 20 9E 48 | 0010 | 000X | 1001 | 1110 | X10X | 10XX |
| 600 | 40 79 20 | 0100 | 000X | 0111 | 1001 | X01X | 00XX |
| 900 | 80 D7 40 | 1000 | 000X | 1101 | 0111 | X10X | 00XX |
| 1200 | 80 E6 40 | 1000 | 000X | 1110 | 0110 | X10X | 00XX |
| 1800 | 02 DB 00 | 0000 | 001X | 1101 | 1011 | X00X | 00XX |
| 2400 | 02 5D 00 | 0000 | 001X | 0101 | 1101 | X00X | 00XX |
| 3600 | 04 3A 00 | 0000 | 010X | 0011 | 1010 | X00X | 00XX |
| 4800 | 04 F2 00 | 0000 | 010X | 1111 | 0010 | X00X | 00XX |
| 7200 | 08 6C 00 | 0000 | 100X | 0110 | 1100 | X00X | 00XX |
| 9600 | 08 CC 00 | 0000 | 100X | 1100 | 1100 | X00X | 00XX |

0 8 C C 0 0

X = don't care

9004

| BAUD RATE | HEX CODE N1 N2 N3 | N1 1234 | N1 5678 | N2 1234 | N2 5678 | N3 1234 | N3 5678 |
|---|---|---|---|---|---|---|---|
| 110 | X7 80 33 | 1 | 0111 | 1000 | 0000 | 0011 | 0011 |
| 150 | X5 80 8B | | 0101 | 1000 | 0000 | 0101 | 1011 |
| 300 | XA 40 B6 | | 1010 | 0100 | 0000 | 1011 | 0110 |
| 600 | X4 20 6D | | 0100 | 0010 | 0000 | 0110 | 1101 |
| 900 | X8 10 F3 | | 1000 | 0001 | 0000 | 1111 | 0011 |
| 1200 | X8 10 DA | | 1000 | 0001 | 0000 | 1101 | 1010 |
| 1800 | X0 08 E7 | | 0000 | 0000 | 1000 | 1110 | 0111 |
| 2400 | X0 08 75 | | 0000 | 0000 | 1000 | 0111 | 0101 |
| 3600 | X0 04 2E | | 0000 | 0000 | 0100 | 0010 | 1110 |
| 4800 | X0 04 EA | | 0000 | 0000 | 0100 | 1110 | 1010 |
| 7200 | X0 02 5C | | 0000 | 0000 | 0010 | 0101 | 1100 |
| 9600 | X0 02 D4 | 1 | 0000 | 0000 | 0010 | 1101 | 0100 |

EIA = 0010

| 1 | N1 1234 |
|---|---|
| X=2/EIA | --10 |
| X=1/CURRENT | --01 |

9004

## BAUD RATE SWITCH SETTINGS FOR THE TLC IOM (WIREWRAP)

| BAUD RATE | SWITCH N1 POSITIONS 12345678 | SWITCH N2 POSITIONS 12345678 | SWITCH N3 POSITIONS 12345678 |
|---|---|---|---|
| 110 | 00**0111 | 10000000 | 00110011 |
| 150 | 00**0101 | 10000000 | 01011011 |
| 300 | 00**1010 | 01000000 | 10110110 |
| 600 | 00**0100 | 00100000 | 01101101 |
| 900 | 00**1000 | 00010000 | 11119011 |
| 1200 | 00**1000 | 00010000 | 11011010 |
| 1800 | 00**0000 | 00001000 | 11100111 |
| 2400 | 00**0000 | 00001000 | 01110101 |
| 3600 | 00**0000 | 00000100 | 00101110 |
| 4800 | 00**0000 | 00000100 | 11101010 |
| 7200 | 00**0000 | 00000010 | 01011100 |
| 9600 | 00**0000 | 00000010 | 11010100 |

MODES

| * | EIA | CURRENT |
|---|---|---|
| POSITION 3 | 1 | 0 |
| POSITION 4 | 0 | 1 |

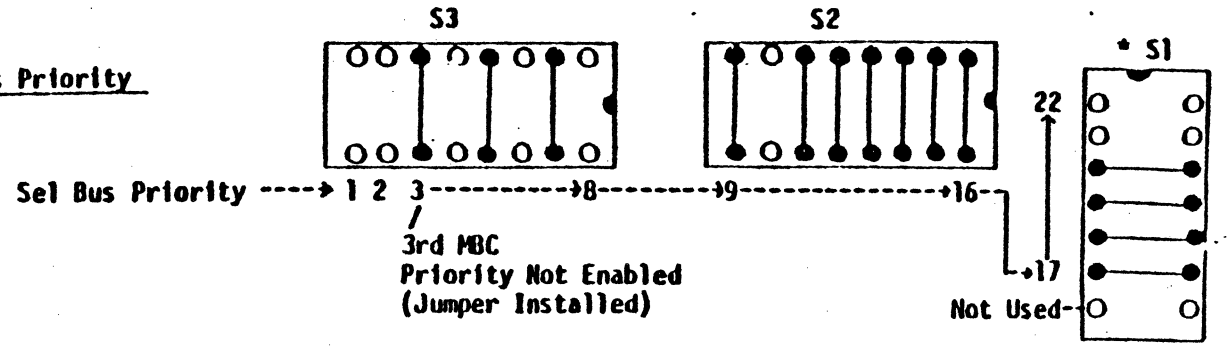
## SEL BUS PRIORITY ENABLE JUMPER CONNECTIONS

The sockets that are to be jumpered for SEL Bus priority are located on the SEL Bus terminator circuit card with coordinates: S1, S2, S3. When an IOM or MBC is installed, the appropriate jumper points are left OPEN (NOT JUMPERED).

As an example, the following assignments are made for a given system; and the jumpered sockets would result, as is illustrated for S1, S2, and S3.

Example of Assignment of SEL Bus Priority

a. First MBC----------> 1
b. Second MBC----------> 2
c. FHD IOM ----------> 4
d. CD IOM ----------> 6
e. MT IOM ----------> 8
f. ADS IOM ----------> 10
g. TLC IOM ----------> 21
h. SCP IOM ----------> 22




* S1 is a 14 pin DIP socket; S2 & S3 are 16 pin DIP sockets.

PICTORIAL ILLUSTRATION OF S1, S2, S3 JUMPER SOCKETS ON SEL BUS TERMINATOR CIRCUIT CARD

NOTE: EXAMPLE ONLY

# NON-STANDARD IOM CONFIGURATION

MAG TAPE IOM:

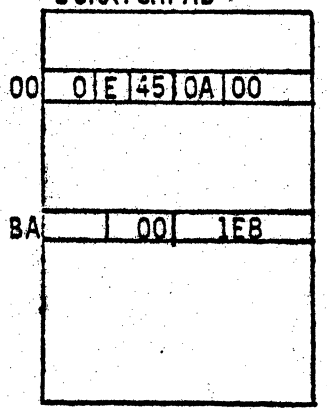
| | |
|---|---|
| DEVICE ADDRESS | 00 |
| PHYSICAL ADDRESS | 0A |
| SEL BUS PRIORITY | 03 |
| S.I. PRIORITY | 3A |
| TCW DED. LOC. | 1A8 |
| IVL DED. LOC. | 1E8 |
| IOCD DED. LOC. | 840 |

take J out of terminator

ICL DEVICE ENTRY:

*DEV00=0E3A0A00

SCRATCHPAD




PROBLEMS:

1) TCW ADDR = IVL ADDR FOR INT LVL 2A
2) IOCD ADDR dedicated for something else?

CALCULATE TCW & IVL ADDRESS FOR ANY INTERRUPT GREATER THAN 23:

```
EX:    INT LVL 3A (HEX)
     -  14      SUBTRACT 14 (HEX)
        26 (HEX)
      010 0110
                SHIFT ANSWER LEFT 2 PLACES
0000 1001 1000  (MULTIPLY BY 4)
+ 1    1    0   ADD 110 (HEX)
  1    A    8   (HEX) TCW DED. LOC.
     + 4    0   ADD 40 (HEX)
  1    E    8   (HEX) IVL DED. LOC.
```

CALCULATE THE IOCD ADDRESS FOR ANY INTERRUPT GREATER THAN 23:

```
EX:    INT LVL 3A (HEX)
     -  14      SUBTRACT 14 (HEX)
        26 (HEX)
      010 0110
                SHIFT ANSWER LEFT 3 PLACES
0001 0011 0000  (MULTIPLY BY 8)
+ 7    1    0   ADD 710 (HEX)
  8    4    0   (HEX) IOCD DED. LOC.
```

# NON-STANDARD IOM CONFIGURATION WORKSESSION

PROBLEM:

1) CONFIGURE THE M/T IOM USING ANY INTERRUPT LEVEL GREATER THAN 3F

2) USE A DEVICE ADDRESS OF 00

3) JUMPER THE PHYSICAL ADDRESS TO 0A

4) JUMPER SEL BUS PRIORITY FOR 03

5) GENERATE THE PROPER ICL ENTRY FOR CONFIGURATION OF 4 DEVICES ON THE CONTROLLER.

6) WRITE A THUMB-IN TO 'WRITE' TO THE TAPE DRIVE THAT IS PHYSICAL UNIT #2.

WORKSESSION

Input/Output Microprocessor (IOM)

Objectives:

This worksession will help to reinforce your knowledge of the IOM. It may also make you aware of some features of the IOM that are not obvious. This worksession may also stimulate questions that you may bring up in class on points that you do not understand.

Referenced Reading:

1. 32/70 System Architecture Course Student Workbook.
   A. Microprocessor, Functional Block Diagram
   B. SEL Bus Interface Block Diagram
2. SEL 32 Series Computer Tech. Manual, Pgs. 1-9 and 1-10, 3-20 thru 3-31.

1. How many input data lines are there from the device logic to the Microprocessor? 16

2. How many data input lines are there to the Microprocessor from the SEL Bus? 32

3. What locations in memory are reserved for Input/Output Command Doublewords (IOCD)? 700-778

4. What is the major difference between a Class 0,1,2 IOCD and a Class 'E' IOCD? Builds FNX Code TCW

5. When is an IOM's SEL Bus Priority zero?

DRT response to CPU

6. What normally causes a "Service Interrupt?" to be generated by an IOM?

Termination I/O

7. What is the difference between Interrupt Level and SEL Bus Priority? Interrupt is for CPU software service

Contention for SEL Bus (hardware)

8. Referencing the ICL Device Entry of *DEV08-0E160800,04

   a) What Device Address would be used to issue a CD 'Write' command to drive #2? Device 0A

   b) What is Drive #2's S.I. priority: 16

9. List the 3 functional areas of an IOM and give a brief description of the function of each.

Microprocessor Device Interface

SEL Bus

10. What does the IOM place on the Data Bus when performing a Memory Read? The physical Address of requesting the device

# DAY 10

# SECTION 10

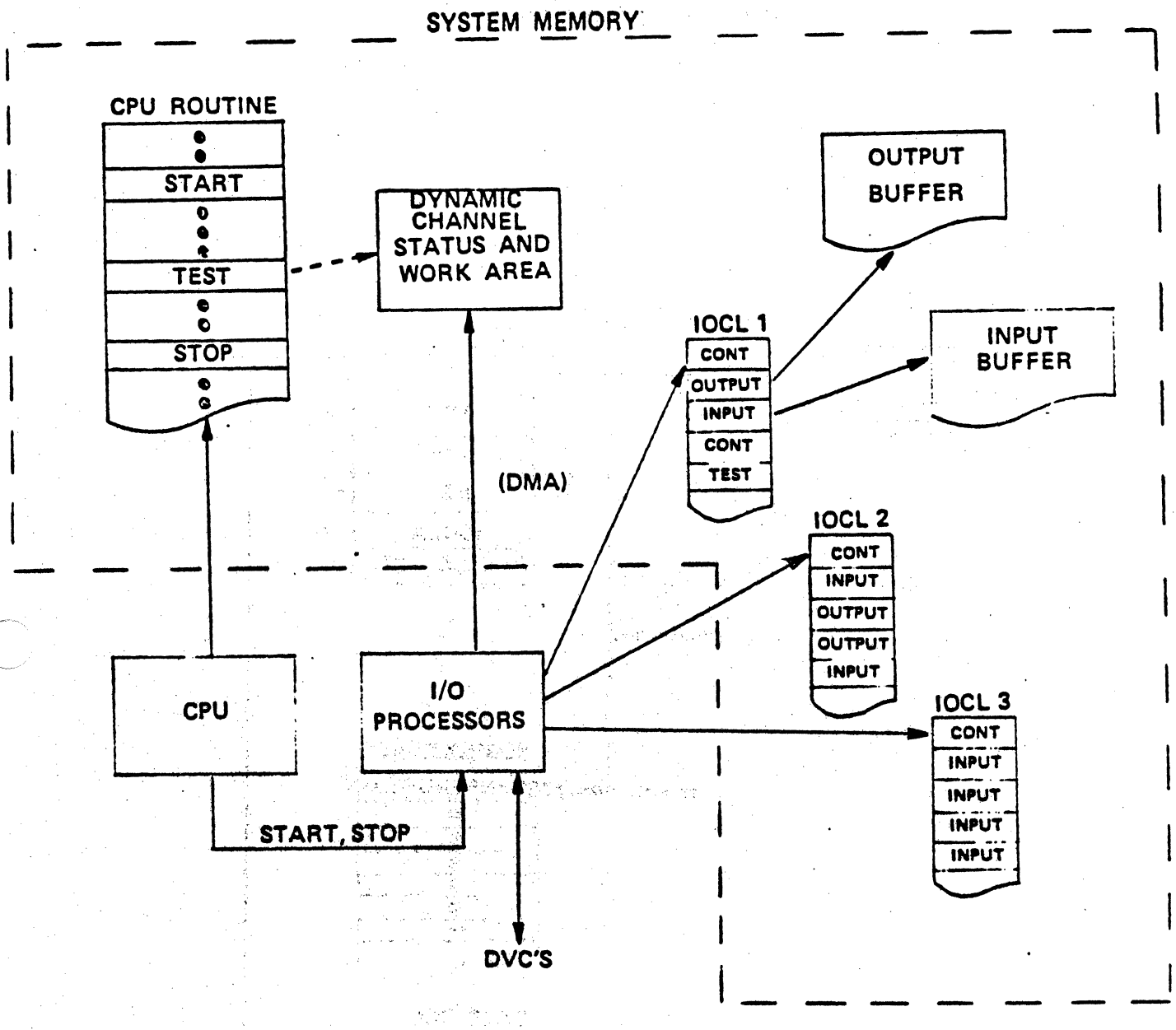
# CLASS 'F' PROGRAMMING

OF YAD

OF MORTGAGE

CLASS 'F' PROGRAMMING

# CLASS 'F' I/O

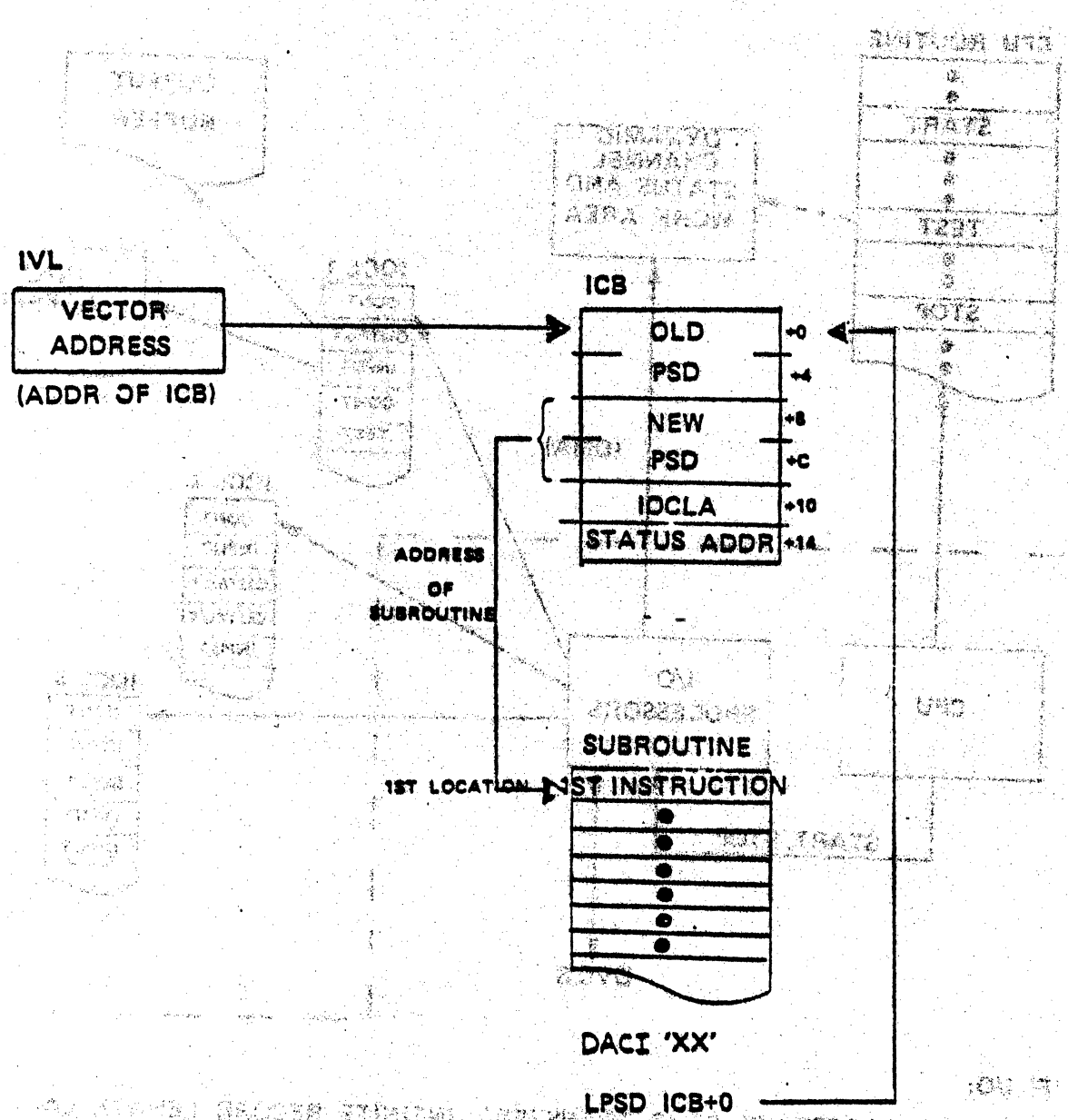



CLASS 'F' I/O:

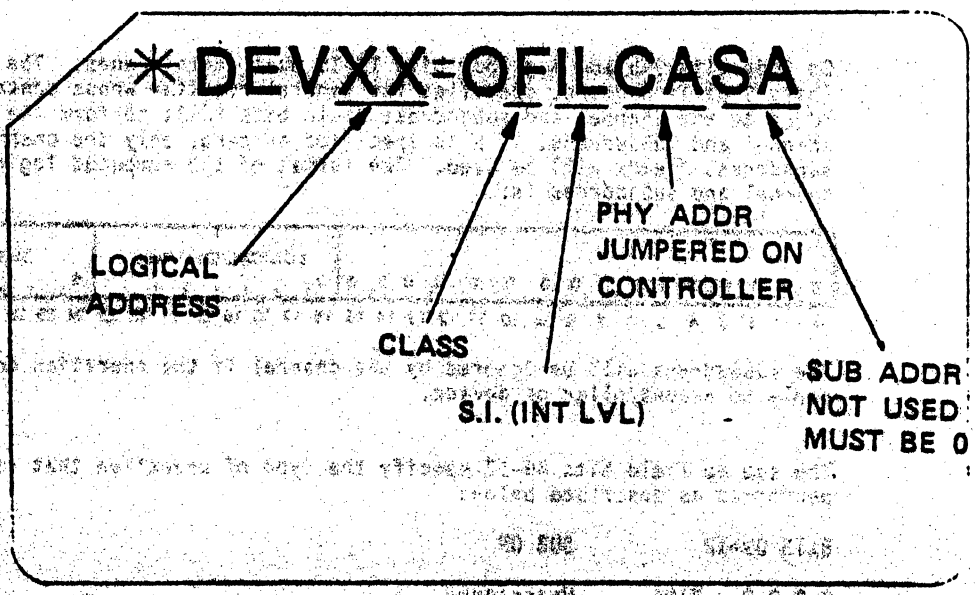
- DMA, 16 MB ADDRESSING', 65 KB TRANSFERS, INFINITE RECORD LENGTH CAPABILITY (DATA CHAINED IOCL CMDS).
- INTELLIGENCE TO PERFORM CONTROL OF MULTIPLE I/O ACTIONS FOR COMPLEX I/O PROCESS
- CLASS F MACRO INSTRUCTION SET PROVIDES FOR ADDRESSING 128 CHANNELS WITH EACH HAVING 255 SUBADDRESSES (DVC'S).
- DYNAMIC STATUS POSTING IN ASSIGNED AREAS OF MEMORY FOR DIRECT VIEWING.
- CLASS F MACRO INSTRUCTION SET HAS A STANDARD I/O PROTOCOL FOR ALL CLASS F PROCESSORS.
- CLASS F: TAPE PROCESSOR
  DISC PROCESSOR

## CLASS 'F'

# I/O INTERRUPT PROCESSING



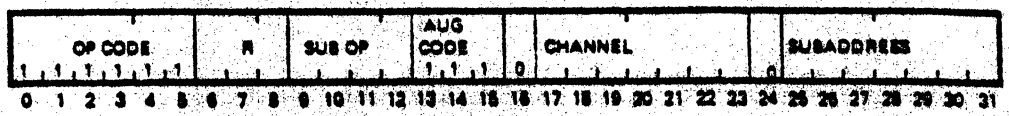
# CLASS 'F' DEVICE ENTRY FORMAT



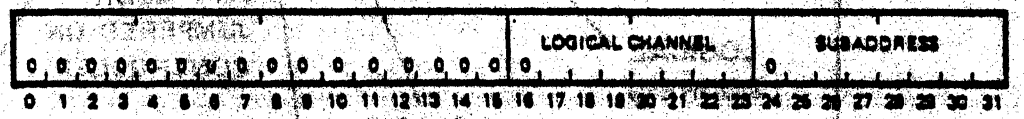
## CLASS F I/O INSTRUCTIONS

INSTRUCTION FORMAT

All Class F I/O instructions will be in the following format:




Op Code bits 0-5 and Aug Code bits 13-15 must contain ones. The R field (bits 6-8) if nonzero, specifies the general register whose contents will be added to the channel and subaddress field bits 16-31 to form the logical channel and subaddress. If R is specified as zero, only the channel and subaddress fields will be used. The format of the computed logical channel and subaddress is:




The subaddress will be ignored by the channel if the operation does not apply to a controller or device.

The sub op field bits 09-12 specify the type of operation that is to be performed as described below:

| BITS 09-12 | SUB OP |
|---|---|
| 0 0 0 0 - X'0' | Unassigned |
| 0 0 0 1 - X'1' | Unassigned |
| 0 0 1 0 - X'2' | START I/O (SIO) |
| 0 0 1 1 - X'3' | TEST I/O (TIO) |
| 0 1 0 0 - X'4' | STOP I/O (STPIO) |
| 0 1 0 1 - X'5' | RESET CHANNEL (RSCHNL) |
| 0 1 1 0 - X'6' | HALT I/O (HIO) |
| 0 1 1 1 - X'7' | GRAB CONTROLLER (GRIO) |
| 1 0 0 0 - X'8' | RESET CONTROLLER (RSCTL) |
| 1 0 0 1 - X'9' | ENABLE WRITE CHANNEL WCS (ECWCS) |
| 1 0 1 0 - X'A' | Unassigned |
| 1 0 1 1 - X'B' | WRITE CHANNEL WCS (WCWCS) |
| 1 1 0 0 - X'C' | ENABLE CHANNEL INTERRUPT (ECI) |
| 1 1 0 1 - X'D' | DISABLE CHANNEL INTERRUPT (DCI) |
| 1 1 1 0 - X'E' | ACTIVATE CHANNEL INTERRUPT (ACI) |
| 1 1 1 1 - X'F' | DEACTIVATE CHANNEL INTERRUPT (DACI) |

NOTES

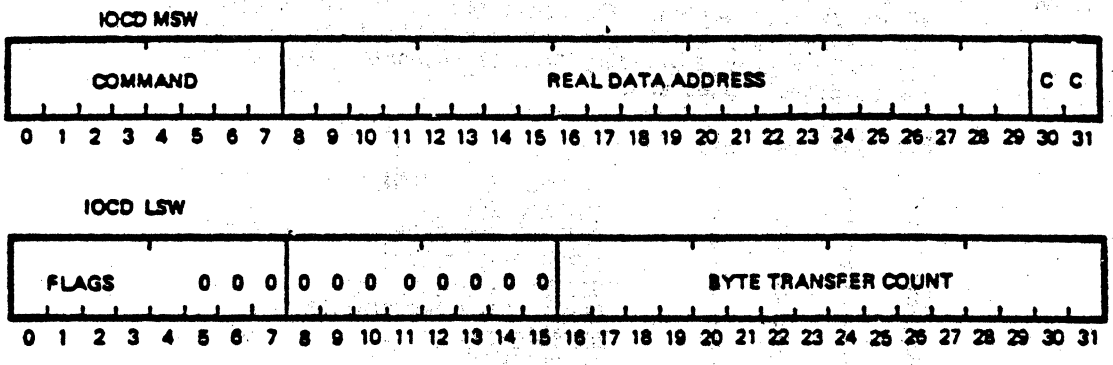
1. Channel must be ICL'd as Class F.
2. EXR, EXRR, and EXM may not be used.
3. Must be in PSD mode.
4. CCs must be tested after each instruction.
5. CD, TD, EI, DI, AI, DAI, and RI cannot be executed to Class F channel.

## CLASS F CONDITION CODES

The condition codes will be set for the execution of all Class F I/O instructions and indicate the successful or unsuccessful initiation of an I/O instruction. The condition codes can be set by the CPU, for channel busy and inoperable or undefined channel, or by the information passed directly from the channel. The assignments for the condition codes are:

| CC1 | CC2 | CC3 | CC4 | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | REQUEST ACTIVATED, WILL ECHO STATUS |
| 0 | 0 | 0 | 1 | CHANNEL BUSY |
| 0 | 0 | 1 | 0 | CHANNEL INOPERABLE OR UNDEFINED |
| 0 | 0 | 1 | 1 | SUBCHANNEL BUSY |
| 0 | 1 | 0 | 0 | STATUS STORED |
| 0 | 1 | 0 | 1 | UNSUPPORTED TRANSACTION |
| 0 | 1 | 1 | 0 | UNASSIGNED |
| 0 | 1 | 1 | 1 | UNASSIGNED |
| 1 | 0 | 0 | 0 | REQUEST ACCEPTED AND QUEUED, NO ECHO STATUS |
| 1 | 0 | 0 | 1 | UNASSIGNED |
| 1 | 0 | 1 | 0 | UNASSIGNED |
| 1 | 0 | 1 | 1 | UNASSIGNED |
| 1 | 1 | 0 | 0 | UNASSIGNED |
| 1 | 1 | 0 | 0 | UNASSIGNED |
| 1 | 1 | 1 | 0 | UNASSIGNED |
| 1 | 1 | 1 | 1 | UNASSIGNED |

Although 16 encoded condtions are possible, only the assigned patterns will occur.




BIT ASSIGNMENTS IN THE COMMAND ARE:

| | |
|---|---|
| X X X X 0 0 0 0 | CHANNEL CONTROL |
| M M M M 0 1 0 0 | SENSE |
| X X X X 1 0 0 0 | TRANSFER IN CHANNEL |
| M M M M 1 1 0 0 | READ BACKWARD |
| M M M M M M 0 1 | WRITE |
| M M M M M M 1 0 | READ |
| M M M M M M 1 1 | CONTROL |

FLAG BIT ASSIGNMENTS ARE:

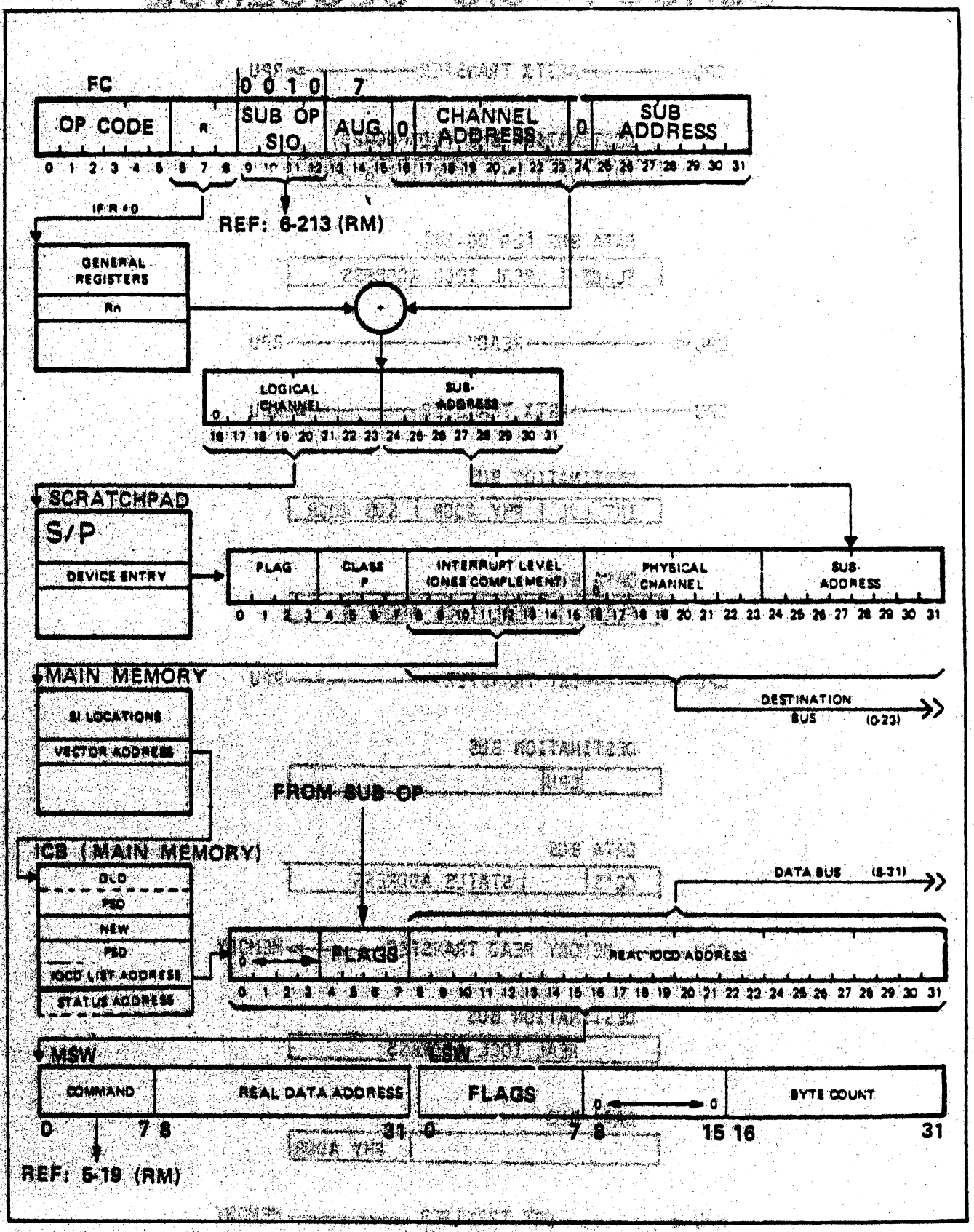
| | |
|---|---|
| 1 0 0 0 0 0 | DATA CHAIN |
| 0 1 0 0 0 0 | CMD CHAIN |
| 0 0 1 0 0 0 | SUPPRESS INCORRECT LENGTH |
| 0 0 0 1 0 0 | SKIP |
| 0 0 0 0 1 0 | POST PROGRAM CONTROLLED INTERRUPT |

C - BIT ASSIGNMENTS ARE:

| BIT 30 | BIT 31 | |
|---|---|---|
| 0 | 0 | BYTE 0 OR FULLWORD |
| 0 | 1 | BYTE 1 OR FIRST HALFWORD |
| 1 | 0 | BYTE 2 OR DOUBLEWORD* |
| 1 | 1 | BYTE 3 OR SECOND HALFWORD |

*IF DOUBLEWORD IS INDICATED TO A CHANNEL, AMBIGUOUS RESULTS MAY OCCUR.

# CLASS F 'SIO' FLOW




I/O Control Words (Class F)

# CLASS F 'SIO' SEQUENCE

CPU ——— ARSTX TRANSFER ———→ RPU

DESTINATION BUS (LDT 00-23)

| INT LVL | PHY ADDR | SUB ADDR |
|---|---|---|

DATA BUS (LD 00-31)

| FLAGS | REAL IOCL ADDRESS |
|---|---|

CPU ←——— READY ——— RPU

CPU ——— RSTX TRANSFER ———→ RPU

DESTINATION BUS

| INT LVL | PHY ADDR | SUB ADDR |
|---|---|---|

DATA BUS

| FLAGS | REAL IOCL ADDRESS |
|---|---|

CPU ←——— DRT TRANSFER ——— RPU

DESTINATION BUS

| | CPU | |
|---|---|---|

DATA BUS

| CC'S | | STATUS ADDRESS |
|---|---|---|

RPU ——— MEMORY READ TRANSFER ———→ MEMORY

DESTINATION BUS

| REAL IOCL ADDRESS |
|---|

DATA BUS

| | PHY ADDR |
|---|---|

RPU ←——— DRT TRANSFER ——— MEMORY

DATA BUS

| CMA | STATUS ADDRESS |
|---|---|

1ST IOCD OF LIST

EACH I/O HANDLER HAS A SCRATCHPAD AREA IN MAIN MEMORY WHERE THE HANDLER KEEPS TRACK OF THE DEVICE CHARACTERISTICS SUCH AS MODEL, TYPE OF RECORDING (NRZI, PE, etc.), AND ALSO EACH DEVICE'S STATUS ADDRESS.

HANDLER SCRATCHPAD AREA IN MAIN MEMORY

| |
|---|
| MODEL |
| RECORDING, NRZI, PE, etc |
| STATUS ADDRESS DEV. 0 |
| STATUS ADDRESS DEV. 1 |
| STATUS ADDRESS DEV. 2 |
| STATUS ADDRESS DEV. 3 |

THE FIRST TIME THE HANDLER (SOFTWARE) EXECUTES AN I/O INSTRUCTION TO A DEVICE, THE 1ST IOCD IN THE I/O COMMAND LIST CONTAINS THE INITIALIZATION BUFFER ADDRESS (SCRATCH-PAD AREA).

EXAMPLE:

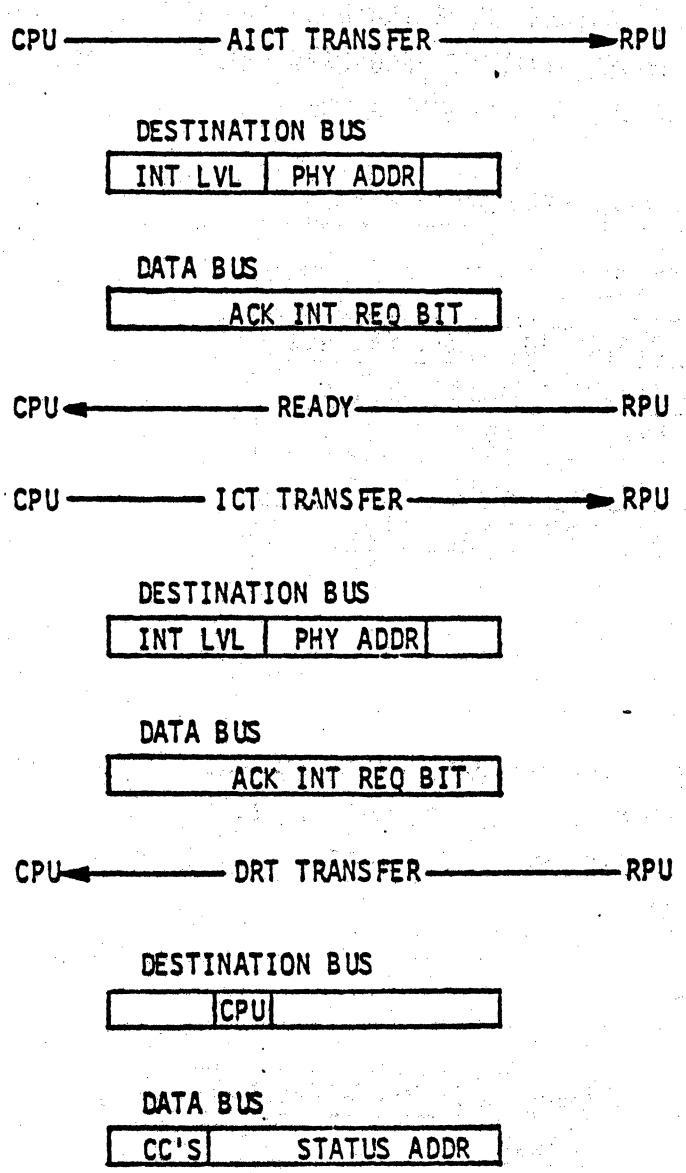
1ST IOCD

| | |
|---|---|
| CMA | INIT BUF ADDRESS |
| FLAGS | |

2ND IOCD

| | | |
|---|---|---|
| COMMAND | REAL DATA ADDRESS | |
| FLAGS | | BYTE XFR COUNT |

WHEN AN INTERRUPT REQUEST IS SENT TO THE CPU FROM THE I/O CHANNEL, THE CPU DOES NOT KNOW WHICH DEVICE IS REQUESTING. THE CHANNEL'S INTERRUPT LEVEL IS SHARED BY ALL DEVICES ON THE CHANNEL. DURING THE SEL BUS AICT, ICT SEQUENCE THE CHANNEL SENDS THE REQUESTING DEVICE STATUS ADDRESS ON THE DATA BUS WITH A DRT.



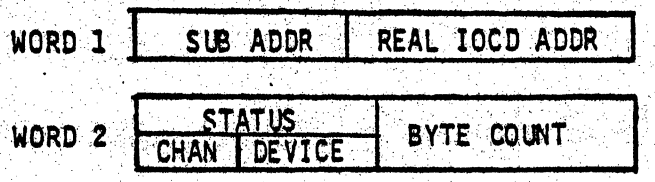

THE CPU STORES THE DEVICE STATUS ADDRESS IN THE ICB (WORD 6)

ICB

| | |
|---|---|
| WORD 1 | OLD PSD1 |
| WORD 2 | OLD PSD2 |
| WORD 3 | NEW PSD1 |
| WORD 4 | NEW PSD2 |
| WORD 5 | IOCL ADDRESS |
| WORD 6 | I/O STATUS ADDRESS |

THE I/O HANDLER CAN NOW ACCESS THE DEVICE STATUS USING THE STATUS ADDRESS IN THE ICB.

DEVICE STATUS IS AS FOLLOWS:



[redacted]

## 340 COURSE REVIEW

1-53 WB

1. What extended functions are available on the Serial Panel?

Extend / 4 → Read Memory Location (20mSec)
FNX → Lamp Test

2. How can PSD2 be displayed on the Serial Panel? Parallel Panel?

Read E

Xfer Scratchpad to Register

3. What are the 4 modes of memory addressing used in the 32/75?

mapped non-extended - Limited 128kw per task (instructions/operands anywhere)
mapped extended - 1MB/task (instructions first map range)
unmapped extended - 1/2 MB (instructions below 128kw)
unmapped non-extended - Limited 128kw total (instructions below 128kw)

4. How many bits are in the 'Program Counter' portion of the PSD?

13-29 = 17bits

5. What is the purpose of the SEA instruction?

To set extended addressing mode in CPU

6. What bit in PSD1 indicates Mapped Mode?

bit 9

7. On execution of a TD4000, where would IOM status be found?

The Condition Codes of PSW, PSD

8. What field in a CD instruction tells the CPU where to locate the Device Entry in Scratchpad?

Device Address Field Logical Address
bits 6-12

9. When executing of TD2000, where is device status stored in Memory? At location specified by TCW dedicated location for that device

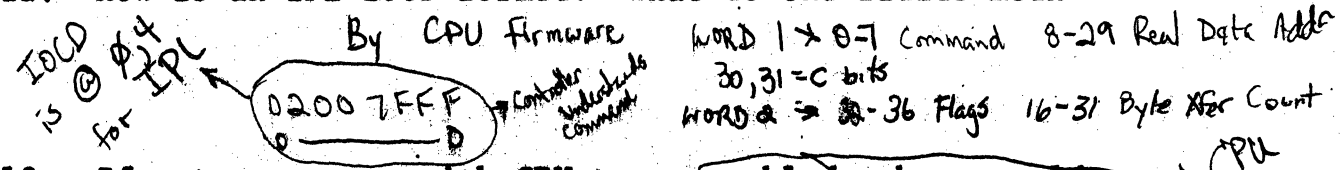
10. Where in Scratchpad is the 'IPL' Device Entry stored?

84

11. How do we determine where a Device Interrupt entry is located in Scratchpad?

IL + 80 = Scratchpad Location

12. How is an IPL IOCD formed? What do the fields mean?




13. If a trap occurs with CPU traps enabled, where would the CPU Status Word be located? If traps are not enabled, where would the CPU Status Word be located?

get CPU

Enabled traps the CPU status would be located 5th word of TCB. Traps Disabled the CPU status word would be located 538.

BEE - Bot External Interrupts Enable
PSW Enable Interrupts 0,1 Hardwired Enabled

14. What is the BLOCK MODE TIMEOUT TRAP? When will it occur?

A watchdog timer. A wait instruction is performed with interrupts blocked or if more than 128 instructions are executed with interrupts blocked

15. Where in memory is the dedicated IVL for Memory Parity Trap? Machine Check Trap?

0E8 Memory Parity Trap IVL

184 Machine Check Trap

16. What type of trap occurs when an unpriveleged program tries to write into a protected area of memory?
Privelege Violation Trap 198

17. What causes a System Check Trap? Software failure that attempted to force the CPU into an illogical sequence.

18. What causes a Machine Check Trap? (Initial Power-Up) Hardware/Firmware failure that occurred during an interrupt or context switch.

19. What happens if the PSD Traps are not enabled and a trap condition occurs? The CPU will halt
530 - 540 will contain info.

20. In the PSD Mode, if the CPU Trap Halts, can you find out why? What information is available? yes.
The type of trap is displayed and by checking the OLD PSD the location (or previous location) of instruction that caused trap and CPU status word provide more specific info.

21. Write a 'SET CPU' instruction that will enable the PSD Mode.
Set CPU 0010|11 000000|1001 NoOp 0000 0000 0000 0010
$R_0$ = 0000 0000 0010 0000 0001 0000 0000 0000
→ 2C090002
$R_0$ = 00001000

22. Write a 'READ CPU STATUS' instruction into Register Zero.
0009 0002 RDSTS; NoOp
CPU status a Reg Ø

23. What is the IVL for the REAL TIME CLOCK? Power Fail?

1AØ = RTC IVL

Trap ØF4 = PF IVL
OR
ØFØ Interrupt

24. What function does the 'SVC' instruction perform? Supervisor Call

Causes a pseudo trap, ... vector is grabbed from a table, ... specific trap subroutine

Can use privileged instructions in non-privileged mode

25. Which Traps do not Halt the CPU if they occur and are not enabled?

Supervisor Call — Arithmetic Exception

Call Monitor

Power Fail — Console Interrupt

26. How can the 32/75 CPU be put into the PSW Mode?

By using the Set CPU mode or changing a jumper on the CPU C Board.

Set Bit 16

27. You have a requirement to install an external interrupt, the 4th RTOM in your system has the LDVINTRØ1 line available. What would be the physical pin on the backplane to connect to and what ICL interrupt entry would you generate?

Pin 5 (PIC) Int ...-51 *Int 51 = CE

79 1st → 78 3rd → 72 4th RTOM (F) → Page 6-14

28. What jumpers must be in L24 CPU 'C' board for 75 Mode and Parallel Control Panel?

Parallel panel 1-10

3-11 ← Serial Panel only

6-11

29. What component of the system has the highest SEL Bus priority? The lowest? CPU ; CPU

Ø ↓ Memory 23

30. What are the field assignments on the Destination Bus for a DRT?

0-21 Memory Address

22,23 C Bits

L-S REQ

L-DTF Line F-bit

31. What is the Tag Line configuration for an MRT?

LTX = L LCNTI = H

LMEM = L LRD = L

LCNTO = H LERROR = H

32. When in the Mapped Mode, does the 'Program Count' in PSD1 give you the real memory address of an instruction? Why?
No. The address is modified by PSD2. Gives Logical Address
PC is only 17 bits

33. In the Mapped Mode, do programs have to be in contiguous memory? Why? No. The MAP Registers specify correct Sequence of 8KW Blocks

34. What is the minimum amount of memory a program is allocated in the Mapped Mode? 8KW

35. What is the minimum amount of memory that can be WRITE protected at any given time?
8KW Mapped
512KW Non Mapped

36. When is the "BP1X" used in PSD2? Gets used in mapping mode determined by Borrow Bit
Used only once when loading MPX-(0:5)

37. How is the CPU put into the Mapped Mode?
By Load Program Status Doubleword Change Map 32,33 of PSD

38. How can you examine the contents of the CPU Map Registers?
Xfer Map - Reg

39. How many bits are used in one Map Register entry (8KW MAP IMAGE)? What are they? 11 bits
Validity Bit - 1
Protect Bit - 1
9 MSB Address - 9

40. Class 'F' I/O operations using the DATA CHAIN flag in several IOCD's involves how many records?

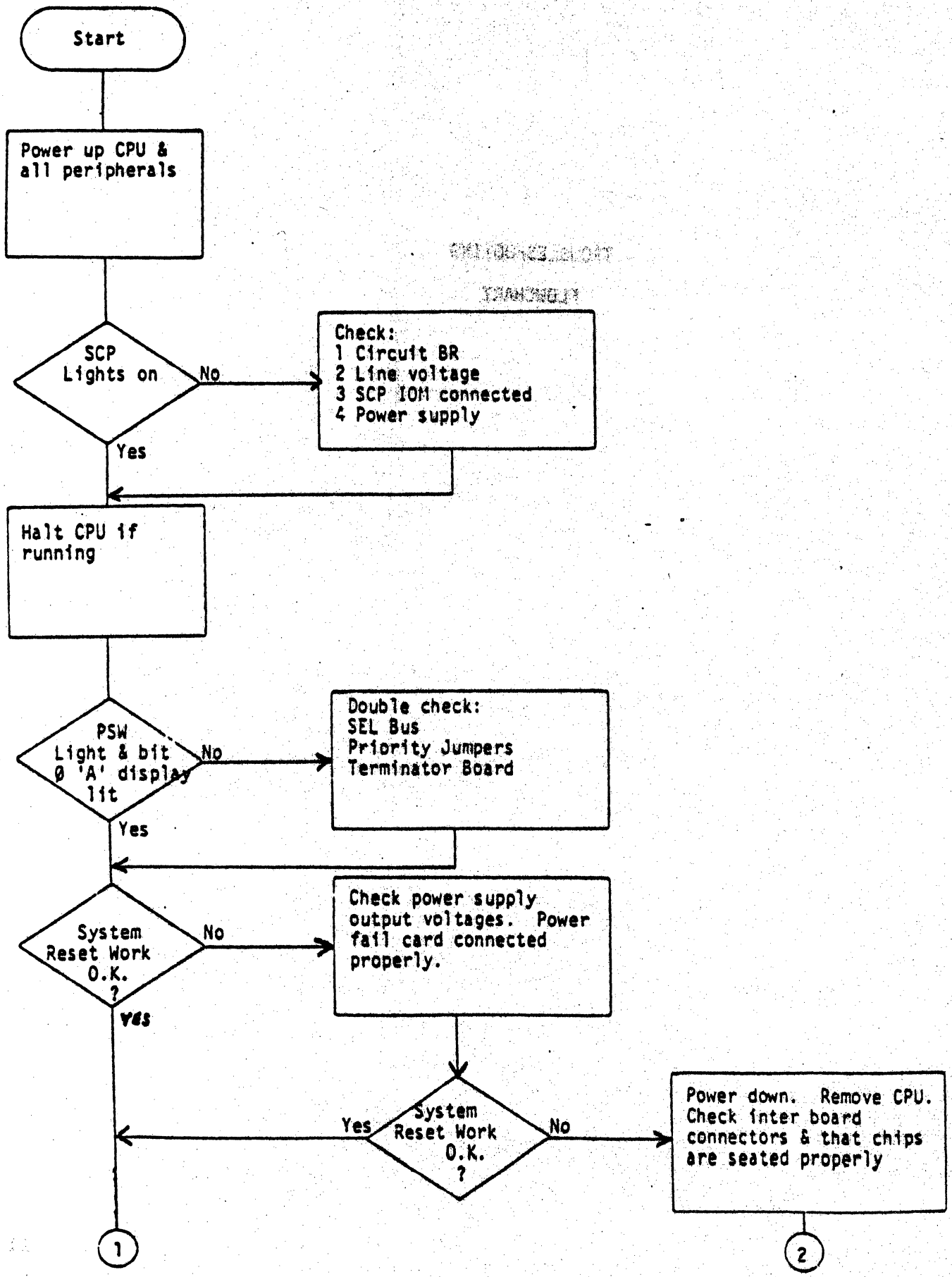
1 Record ∞ length

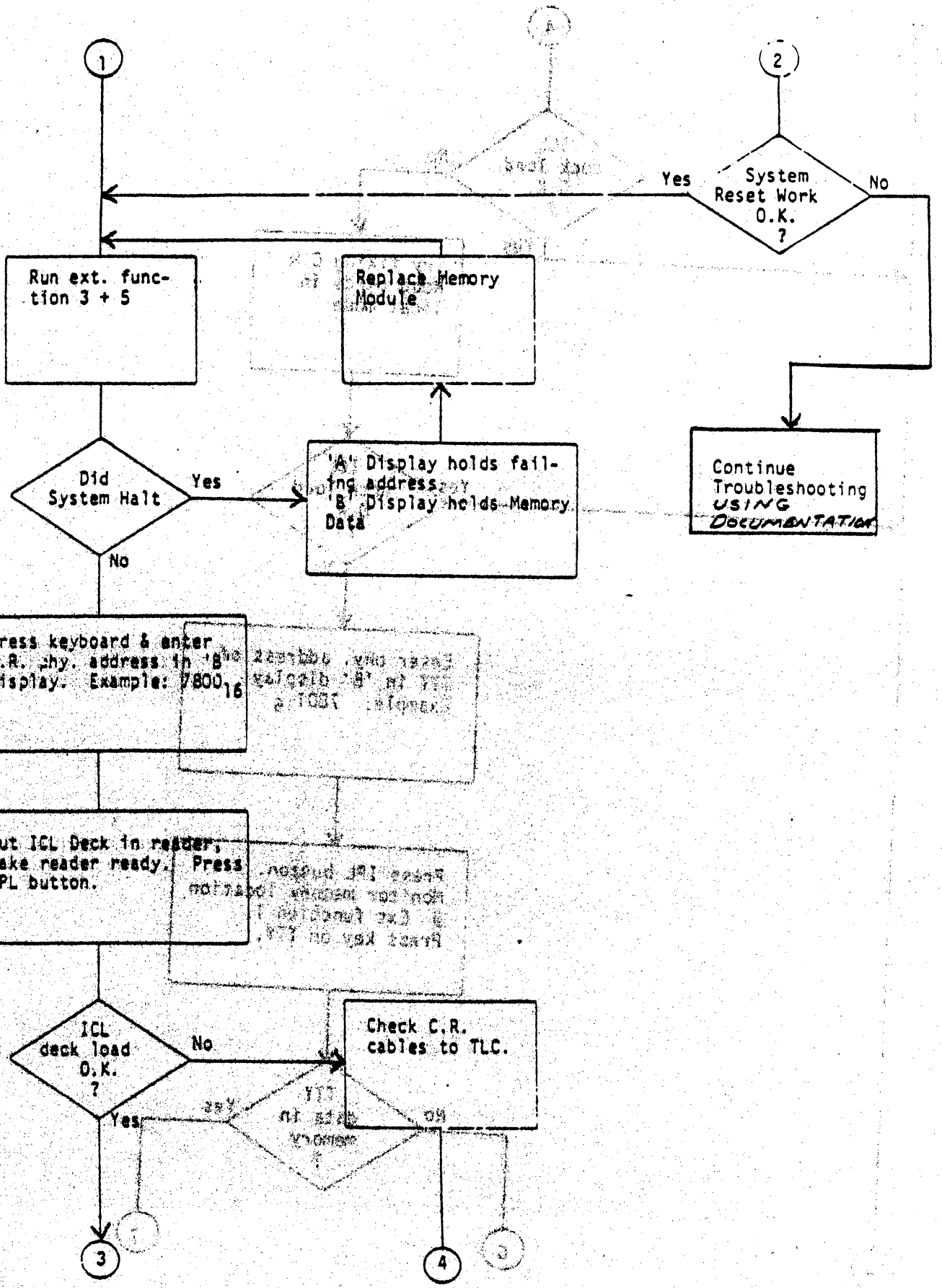
# SECTION 11

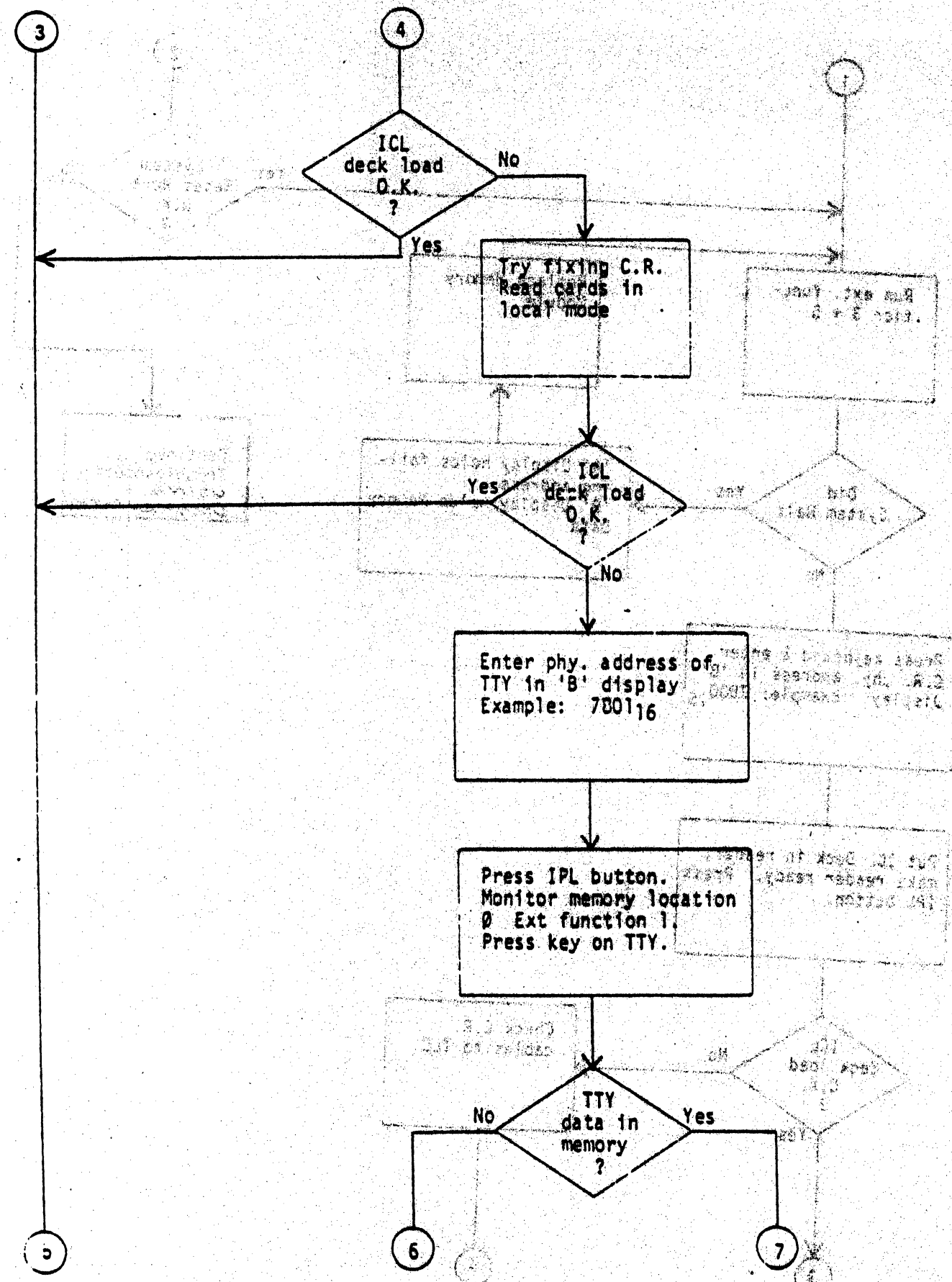
# MISCELLANEOUS INFORMATION

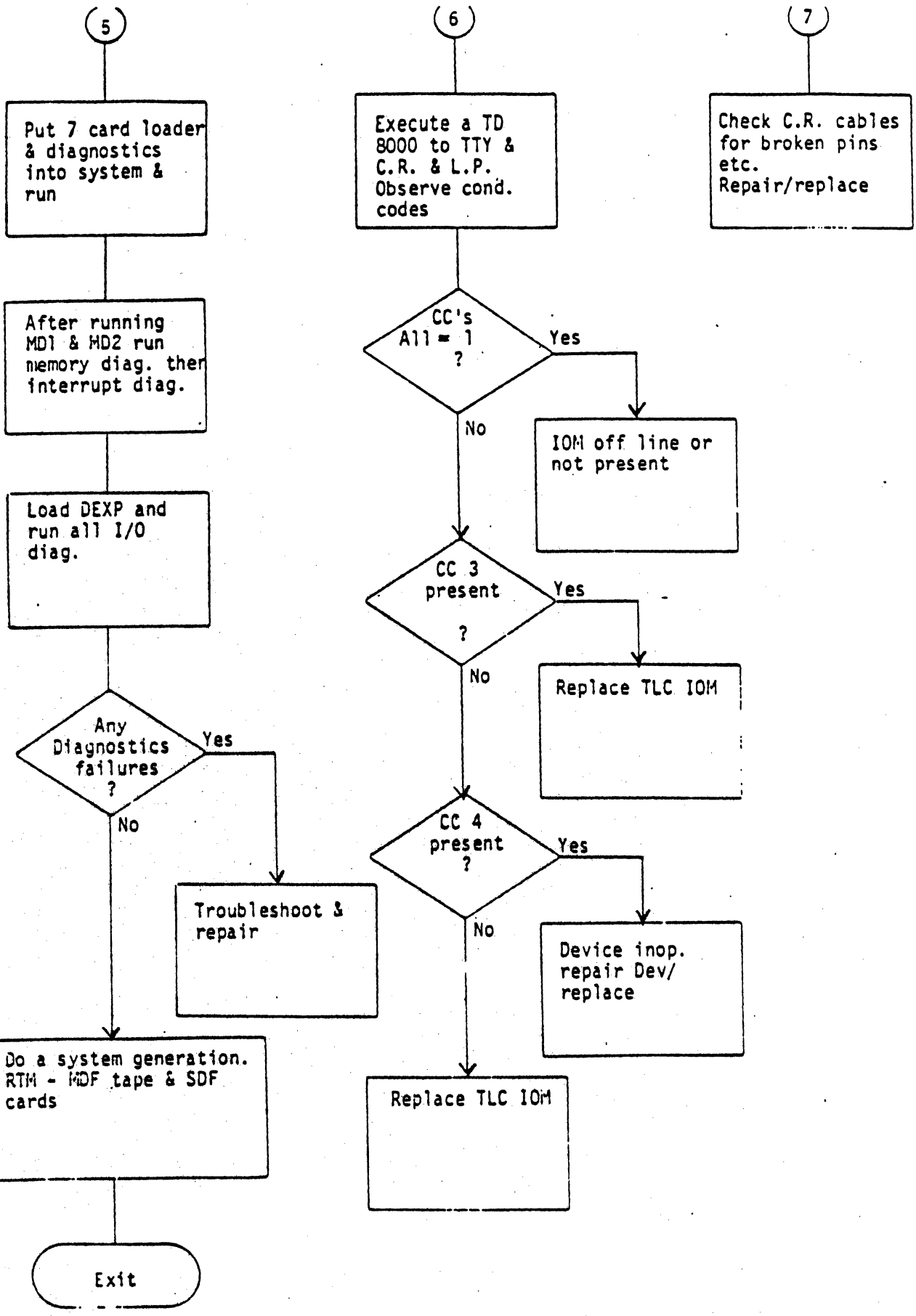
# TROUBLESHOOTING

# FLOWCHART


Start
Power up CPU & all peripherals
SCP Lights on
No
Check:
1 Circuit BR
2 Line voltage
3 SCP IOM connected
4 Power supply
Yes
Halt CPU if running
PSW Light & bit Ø 'A' display lit
No
Double check:
SEL Bus
Priority Jumpers
Terminator Board
Yes
System Reset Work O.K. ?
No
Check power supply output voltages. Power fail card connected properly.
Yes
System Reset Work O.K. ?
Yes
No
Power down. Remove CPU. Check inter board connectors & that chips are seated properly
1
2


1
2
System Reset Work O.K. ?
Yes
No
Run ext. function 3 + 5
Replace Memory Module
Did System Halt
Yes
No
'A' Display holds failing address
'B' Display holds Memory Data
Continue Troubleshooting USING DOCUMENTATION
Press keyboard & enter C.R. phy. address in 'B' Display. Example: 7800 16
Put ICL Deck in reader, make reader ready. Press IPL button.
ICL deck load O.K. ?
No
Yes
Check C.R. cables to TLC.
3
4


3
4
ICL deck load O.K. ?
No
Yes
Try fixing C.R. Read cards in local mode
ICL deck load O.K. ?
Yes
No
Enter phy. address of TTY in 'B' display Example: 7001 16
Press IPL button. Monitor memory location Ø Ext function 1. Press key on TTY.
TTY data in memory ?
No
Yes
6
7
5


5
6
7
Put 7 card loader & diagnostics into system & run
After running MD1 & MD2 run memory diag. then interrupt diag.
Load DEXP and run all I/O diag.
Any Diagnostics failures ?
Yes
No
Troubleshoot & repair
Do a system generation. RTM - MDF tape & SDF cards
Exit
Execute a TD 8000 to TTY & C.R. & L.P. Observe cond. codes
CC's All = 1 ?
Yes
No
IOM off line or not present
CC 3 present ?
Yes
No
Replace TLC IOM
CC 4 present ?
Yes
No
Device inop. repair Dev/ replace
Replace TLC IOM
Check C.R. cables for broken pins etc. Repair/replace

# S.E.L. 32 TROUBLESHOOTING FORM

NAME_____

WEEK_____ DAY_____ TIME EXPENDED_____

TROUBLE WAS: BOARD_____ CHIP_____

SYMPTOMS:

TEST & MAINTENANCE ROUTINES USED TO FIND PROBLEM:

STUDENTS COMMENTS:

# S.E.L. 32 TROUBLESHOOTING FORM

NAME_____

WEEK_____ DAY_____ TIME EXPENDED_____

TROUBLE WAS: BOARD_____ CHIP_____

SYMPTOMS:

TEST & MAINTENANCE ROUTINES USED TO FIND PROBLEM:

STUDENTS COMMENTS:

## S.E.L. 32 TROUBLESHOOTING FORM

NAME_____

WEEK_____ DAY_____ TIME EXPENDED_____

TROUBLE WAS: BOARD_____ CHIP_____

SYMPTOMS:

TEST & MAINTENANCE ROUTINES USED TO FIND PROBLEM: -

STUDENTS COMMENTS:

# S.E.L. 32 TROUBLESHOOTING FORM

NAME_____

WEEK_____ DAY_____ TIME EXPENDED_____

TROUBLE WAS: BOARD_____ CHIP_____

SYMPTOMS:

TEST & MAINTENANCE ROUTINES USED TO FIND PROBLEM:

STUDENTS COMMENTS:

# ABBREVIATIONS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ADS | ASYNCHRONOUS DATA SET, USED TO INTERFACE COMMUNICATIONS TERMINALS AND MODEMS TO THE SYSTEM. |
| AICT | ADVANCE INTERRUPT CONTROL TRANSFER. SELBUS. |
| ALIM | ASYNCHRONOUS LINE INTERFACE MODULE, IS A GPDC, USED TO INTERFACE COMMUNICATIONS DEVICES TO THE SYSTEM. |
| ADI | ANALOG DIGITAL INTERFACE, INTERFACES REAL TIME PERIPHERALS (RTP'S) SUCH AS ANALOG TO DIGITAL, DIGITAL TO ANALOG, CONVERTERS, ETC. TO THE SYSTEM. |
| ALU | ARITHMETIC LOGIC UNIT. |
| ANSI | AMERICAN NATIONAL STANDARDS INSTITUTE. |
| ARSTX | ADVANCE READ STATUS TRANSFER, IS A SELBUS TRANSFER. |
| BLIM | BINARY SYNCHRONOUS LINE INTERFACE MODULE, USED TO INTERFACE COMMUNICATIONS DEVICES TO THE SYSTEM. |
| BOT | BEGINNING OF TAPE. |
| BPI | BITS PER INCH, NUMBER OF BITS PER INCH ON MAG TAPE. |
| BPIX | BASE PROCESS INDEX, 32/75 PSD2. |
| CALM | CALL MONITOR, USED IN SOFTWARE TO COMMUNICATE WITH THE REAL TIME MONITOR (RTM). |
| CC | CONDITION CODE. FOUR BITS IN THE PSW. |
| DC | CARTRIDGE DISC OR COMMAND DEVICE (INSTRUCTION). |
| CDC | CARTRIDGE DISC CONTROLLER, IS AN IOM. |
| CE | CUSTOMER ENGINEER. |
| CR | CARD READER, READ PUNCHED CARDS. |
| CMOS | CAPACITOR METAL OXIDE SEMI-CONDUCTOR. |
| CPIX | CURRENT PROCESS INDEX, 32/75 PSD2. |
| CROM | CONTROL READ ONLY MEMORY, USUALLY THE FIRMWARE FOR A MICROPROCESSOR. |
| CP | CARD PUNCH, PUNCHES HOLES IN CARDS. |
| CPC | CARD PUNCH CONTROLLER, GPDC. |

| | |
|---|---|
| CPM | CARDS PER MINUTE. SPEED OF CARD READER. |
| CPU | CENTRAL PROCESSING UNIT, USED TO CONTROL THE ENTIRE SYSTEM. |
| CRC | CYCLICAL REDUNDENCY CHECK, PARITY CHECK CHARACTER ON NINE LEVEL (9 TRACK) MAGNETIC TAPE. |
| CRT | CATHODE RAY TUBE, USED AS OPERATOR CONSOLE OR COMMUNICATIONS TERMINALS. |
| CYL | CYLINDER, TERM USED WITH DISC DEVICES. |
| DCC | DEVICE CONTROLLER CHANNEL. USED TO IDENTIFY DEVICES DURING SYSGEN. |
| DRT | DATA RETURN TRANSFER. SELBUS. |
| DSS | DEVELOPMENT SUPPORT SYSTEM. |
| ECC | ERROR CORRECTION CODE. MOVING HEAD DISC. |
| ECL | EMITTER COUPLED LOGIC. |
| ECO | ENGINEERING CHANGE ORDER. |
| EOT | END OF TAPE. |
| FCB | FILE CONTROL BLOCK. SOFTWARE. |
| FHD | FIXED HEAD DISC, HEADS ARE ELECTRONICALLY SELECTED TO DESIRED TRACK. THERE IS A FIXED READ/WRITE HEAD FOR EACH TRACK. MAGNETIC STORAGE DEVICE. |
| FM | FREQUENCY MODULATION, ALSO FILE MARK ON MAG TAPE. |
| FPU | FLOATING POINT UNIT, USED TO PERFORM HIGH SPEED ARITHMETIC COMPUTATIONS. |
| FXD | FULL DUPLEX. COMMUNICATIONS TERM. |
| GPDC | GENERAL PURPOSE DEVICE CONTROLLER, USED WITH THE GPMC. |
| GPIO | GENERAL PURPOSE INPUT OUTPUT MICROPROCESSOR. |
| GPMC | GENERAL PURPOSE MULTIPLEXING CONTROLLER, USED TO CONTROL UP TO SIXTEEN GPDC'S. |
| HSD | HIGH SPEED DATA INTERFACE, USED TO INTERFACE HIGH SPEED DEVICES TO THE SYSTEM. SUCH AS ANOTHER COMPUTER. |
| HXD | HALF DUPLEX. COMMUNICATIONS TERM. |
| IBL | INTER-BUSS LINK. USED TO CONNECT TWO SELBUSSES. |
| ICB | INTERRUPT CONTEXT BLOCK, OR INITIAL CLEAR BUS. |
| ICL | INITIAL CONFIGURATION LIST. USED TO DEFINE THE PHYSICAL SYSTEM ENVIRONMENT TO THE CPU. STORED IN CPU SCRATCHPAD. |

| | |
|---|---|
| ICT | INTERRUPT CONTROL TRANSFER. SELBUS. |
| IOCB | INPUT OUTPUT COMMAND BLOCK. |
| IOCD | INPUT OUTPUT COMMAND DOUBLEWORD. |
| IOCL | INPUT OUTPUT COMMAND LIST. |
| IOM | INPUT OUTPUT MICROPROCESSOR. USED TO CONTROL I/O DEVICES. |
| IPL | INITIAL PROGRAM LOAD. USED TO LOAD PROGRAMS INTO THE SYSTEM FROM AN INPUT DEVICE. |
| IPS | INCHES PER SECOND ( TAPE SPEED ). |
| IRG | INTER RECORD GAP. MAG TAPE. |
| LED | LIGHT EMITTING DIODE. |
| LP | LINE PRINTER. |
| LPM | LINES PER MINUTE. |
| LRC | LONGITUDINAL REDUNDINCY CHECK. TRACK PARITY-ON TAPE. |
| LSB | LEAST SIGNIFICANT BIT. |
| LSI | LARGE SCALE INTEGRATION. |
| MBA | MEMORY BUS ADAPTER. USED TO CONNECT A SELBUS TO REMOTE MEMORY. |
| MBC | MEMORY BUS CONTROLLER. CONTROLS MEMORY MODULES. |
| MDF | MONITOR DISTRIBUTION FILE. SOFTWARE. |
| MHD | MOVING HEAD DISC. DIRECT ACCESS DEVICE. |
| MIA | MEMORY INTERFACE ADAPTER. SAME AS MBA. |
| MID | MAP IMAGE DESCRIPTOR, 32/75 MEMORY MANAGEMENT COMPONENT. |
| MIOP | MULTIPLEXING INPUT OUTPUT PROCESSOR. USED TO CONTROL VARIOUS TYPES OF I/O CONTROLLERS ON A TIME SHARING BASIS. |
| MIU | MEMORY INTERFACE UNIT. |
| MOS | METAL OXIDE SEMI-CONDUCTOR. |
| MPE | MEMORY PARITY ERROR. |
| MPL | MASTER PROCESS LIST. LOCATIONS 784 THRU 7FC. (32/75) |
| MPX | MAPPED PROGRAM EXECUTIVE. 32/75 OPERATING SYSTEM SOFTWARE. |

| | |
|---|---|
| MRLT | MEMORY READ AND LOCK TRANSFER. SELBUS |
| MRT | MEMORY READ TRANSFER. |
| MSD | MOST SIGNIFICANT DIGIT. |
| MTF | MAGNETIC TAPE FORMATTER |
| MUX | MULTIPLEXOR. USUALLY A CHIP THAT SELECTS ONE OF SEVERAL INPUTS AND SENDS IT TO THE CHIP'S OUTPUT. |
| MSDL | MAP SEGMENT DESCRIPTOR LIST. 32/75 MEMORY MANAGEMENT. |
| MTU | MAGNETIC TAPE UNIT. |
| MWT | MEMORY WRITE TRANSFER. SELBUS. |
| PCB | PRINTED CIRCUIT BOARD. |
| PDX | PROGRAM DEVELOPMENT EXECUTIVE. |
| PE | PHASE ENCODED. A TYPE OF MAGNETIC RECORDING USED ON TAPE. |
| PFS | POWER FAIL SAFE. |
| PLO | PHASE LOCK OSCILLATOR. USED IN DISC DEVICES. |
| PROM | PROGRAMMABLE READ ONLY MEMORY. ROM CHIP. |
| PSD | PROGRAM STATUS DOUBLEWORD. 32/75 |
| PSW | PROGRAM STATUS WORD. 32/55 |
| PTR | PAPER TAPE READER. |
| PTRP | PAPER TAPE READER AND PUNCH. |
| QC | QUALITY CONTROL. |
| RAM | RANDOM ACCESS MEMORY. SOLID STATE DEVICE (CHIP). |
| ROM | READ ONLY MEMORY. SOLID STATE DEVICE. |
| RPU | REGIONAL PROCESSING UNIT. |
| RSTX | READ STATUS TRANSFER. SELBUS. |
| RTOM | REAL TIME OPTION MODULE. USED TO HANDLE INTERRUPTS. |
| RTM | REAL TIME MONITOR, 32/55 SOFTWARE OPERATING SYSTEM. |
| SCPI | SYSTEM CONTROL PANEL INTERFACE. AN IOM. |
| SDC | SEGMENT DESCRIPTOR COUNT. 32/75 MEMORY MANAGEMENT COMPONENT. |

| | |
|---|---|
| SDF | SYSTEM DIRECTORY FILE. USED DURING SYSGEN. |
| SDI | SERIAL DATA INTERFACE. |
| SDS | SYNCHRONOUS DATA SET. |
| SEC | SECTOR ON A DISC. |
| SI | SERVICE INTERRUPT. OCCURS WHENEVER A DEVICE TERMINATES. |
| SLIM | SYNCHRONOUS LINE INTERFACE MODULE, COMMUNICATIONS TERM. IS A GPDC. |
| SYSGEN | SYSTEM GENERATION. SOFTWARE SYSTEM BUILT ON DISC. |
| TAW | TRANSFER ADDRESS WORD. |
| TCW | TRANSFER CONTROL WORD. HAS TRANSFER COUNT AND DATA ADDRESS. |
| TLC | TELETYPE, LINE PRINTER, CARD READER, CONTROLLER. |
| TSA | TASK SERVICE AREA. SOFTWARE TERM. |
| TSB | TECHNICAL SUPPORT BULLETIN. CUSTOMER SERVICE. |
| TSM | TERMINAL SERVICES MANAGER. SOFTWARE, USED WITH MPX FOR TERMINAL SUPPORT. |
| TSS | TERMINAL SUPPORT SYSTEM. SOFTWARE, USED WITH RTM FOR TERMINAL SUPPORT. |
| TTY | TELETYPE TERMINAL. |
| UART | UNIVERSAL ASYNCHRONOUS RECEIVE AND TRANSMIT. (CHIP) |
| VFO | VARIABLE FREQUENCY OSCILLATOR. |
| VLSI | VERY LARGE SCALE INTEGRATION. |
| VRC | VERTICAL REDUNDANCY CHECK. BYTE.PARITY ON MAG TAPE. |
| WCS | WRITABLE CONTROL STORAGE. CPU OR I/O MICROPROCESSOR. |
| WDOT | WORD DATA OR ORDER TRANSFER. SELBUS. |

# APPENDIX

## GLOSSARY OF BUZZ WORDS

A

ACCESS TIME

The time interval between the request for information and the instant this information is available.

ACCOUNTING MACHINE

1. A keyboard actuated machine that prepares accounting records.
2. A machine that reads data from external storage media, such as cards or tapes, and automatically produces accounting records or tabulations, usually on continuous forms.

ACCUMULATOR

A device which stores a number and which, on receipt of another number, adds the two and stores the sum.

ACCURACY

The degree of freedom from error, that is, the degree of conformity to truth or to a rule. Accuracy is contrasted with precision. For example, four-place numerals are less precise than six-place numerals, nevertheless a properly computed four-place numeral might be more accurate than an improperly computed six-place numeral.

ACK

Acknowledge message sent upon a communication link to indicate the reception of correct data. Used with error detectors in block and tree codes.

A (Cont'd)

ADAPTER

A device used to effect operative capability between different parts of one or more systems or subsystems.

ADDER

Switching circuit that combines binary bits to generate the Sum and Carry of these bits.

ADDRESS

An address is a coded instruction designating the location of data or program segments in storage. The address may refer to storage in registers or memories or both. The address code itself may be stored so that a location may contain the address of data rather than the data itself. This form of addressing is common in microprocessors. Addressing modes vary considerably because of efforts to reduce program execution time.

ADDRESS FORMAT

1. The arrangement of the address parts of an instruction. The expression "plus-one" is frequently used to indicate that one of the addresses specifies the location of the next instruction to be executed, such as one-plus-one, two-plus-one, three-plus-one, four-plus-one.

2. The arrangement of the parts of a single address, such as those required for identifying channel, module, track, etc., in a disc system.

APPENDIX

GLOSSARY OF BUZZ WORDS

A (Cont'd)

ADDRESS REGISTER

A register in which an address is stored.

ALGORITHM

A term used by mathmaticians to describe a set of procedures by which a given result is obtained.

ALPHANUMERIC CODE

A code whose code set consists of letters, digits, and associated special characters.

ALU (ARITHMETIC AND LOGIC UNIT):

The ALU is one of the three essential components of a microprocessor... the other two being the registers and the control block. The ALU performs various forms of addition and subtraction; the logic mode performs such logic operations as ANDing the contents of two registers, of masking the contents of a register.

ANALOG REPRESENTATION

A representation that does not have discrete values but is continuously variable.

ARCHITECTURE

Any design or orderly arrangement perceived by man; the architecture of the microprocessor. Since their existance microprocessors vary considerably in design, their architecture has become a bone of contention among specialists.

A (Cont'd)

ARITHMETIC SHIFT

1. A shift that does not affect the sign position.

2. A snift that is equivalent to the multiplication of a number by a positive or negative integral power of the radix.

ARRAY LOGIC

A logic network whose configuration is a rectangular array of intersections of its input-output leads, with elements connected at some of these intersections. The network usually functions as an encoder or decoder.

ASCII

American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for informatior interchange among data processing systems, communication systems, and associated equipment. The ASCII set consists of control characters ard graphic characters. Synonymous with USASCII.

ASSEMBLE

To prepare a machine language program from a symbolic language program by substituting absolute operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

ASSEMBLER

A computer program that assembles.

# APPENDIX

## GLOSSARY OF BUZZ WORDS

A (Cont'd)

ASSEMBLER PROGRAM

The Assembler Program translates man readable source statements (mnemonics) into machine understandable object code.

ASSEMBLY LANGUAGE

A machine oriented language. Normally the program is written as a series of source statements using mnemonic symbols that suggest the definition of the instruction and is then translated into machine language.

ASYNCHRONOUS

Operation of a switching network by a free-running signal which signals successive instructions, the completion of one instruction triggering the next. There is no fixed time per cycle.

ASYNCHRONOUS DEVICE

A device in which the speed of operation is not related to any frequency in the system to which it is connected.

B

BAUD

A unit of signaling speed equal to the number of discrete conditions or signal events per second. For example, one baud equals one-half dot cycle per second in Morse code, one bit per second in a train of binary signals, and one 3-bit value per second in a train of signals each of which can assume one of eight different states.

B (Cont'd)

BAUD RATE

A measure of data flow. The number of signal elements per second based on the duration of the shortest element. When each element carries one bit, the Baud rate is numerically equal to bits per second (bps). The Baud rates on UART data sheets are interchangeable with bps.

BAUDOT CODE

Information code used in data transmission.

BCD (BINARY CODED DECIMAL)

Each decimal digit is binary coded into 4-bit words. The decimal number 11 would become 0001 0001 in BCD. Also known as the 8421 code.

BENCHMARK

Originally a surveyor's mark used as a reference point in surveys. In connection with microprocessors, the benchmark is a frequently used routine or program selected for the purpose of comparing different makes of microprocessors. A flow chart in assembly language is written out for each microprocessor and the execution of the benchmark by each unit is evaluated on paper. It is not necessary to use hardware to measure capability by benchmark.

BENCHMARK PROBLEM

A problem used to evaluate the performance of hardware or software or both.

B (Cont'd)

BIDIRECTIONAL

A term applied to a port or bus line that can be used to transfer data in either direction.

BINARY

A system of numbers using 2 as a base in contrast to the decimal system which uses 10 as a base. The binary system requires only two symbols, 0 and 1. Two is expressed in binary by the number 10 (read one, zero). Each digit after the initial 1 is multiplied by the base 2. Hence the following table expresses the first ten numbers in decimal and binary:

| Decimal | Binary | Decimal | Binary |
|---|---|---|---|
| 0 | 0 | 5 | 101 |
| 1 | 1 | 6 | 110 |
| 2 | 10 | 7 | 111 |
| 3 | 11 | 8 | 1000 |
| 4 | 100 | 9 | 1001 |

BINARY CODED DECIMAL (BCD)

A binary numbering system for coding decimal numbers in groups of 4 bits. The binary value of these 4-bit groups ranges from 0000 to 1001, and codes the decimal digits "0" through "9". To count to 9 takes 4 bits; to count to 99 takes two groups of 4 bits; to count to 999 takes three groups of 4 bits, etc.

B (Cont'd)

BLOCK

1. A set of things, such as words, characters, or digits handled as a unit.

2. A collection of contiguous records recorded as a unit. Blocks are separated by block gaps and each block may contain one or more records.

3. A group of bits, or n-ary digits, transmitted as a unit. An encoding procedure is generally applied to the group of bits or n-ary digits for error-control purposes.

4. A group of contiguous characters recorded as a unit.

BLOCK DIAGRAM

A diagram of a system, instrument, or computer in which the principal parts are represented by suitable associated geometrical figures to show both the basic functions and the functional relationships among the parts.

BOOTSTRAP

A technique or device designed to bring itself into a desired state by means of its own action, e.g., a machine routine whose first few instructions are sufficient to bring the rest of itself into the computer from an input device.

## APPENDIX

## GLOSSARY OF BUZZ WORDS

B (Cont'd)

BORROW

An arithmetically negative carry.

BRANCH

Refers to the capability of a microprocessor to modify the function or program sequence. Such modification depends on the actual content of the data being processed at any given instant.

BRANCHING

A method of selecting, on the basis of results, the next operation to execute while the program is in progress.

BUS DRIVER

An integrated circuit which is added to the data bus system to facilitate proper drive to the CPU when several memories are tied to the data bus line. These are necessary because of capacitive loading which slows down the data rate and prevents proper time sequencing of microprocessor operation.

BUS SYSTEM

A network of paths inside the microcomputer which facilitate data flow. The important busses in a microprocessor are identified as Data Bus, Address Bus, and Control Bus.

B (Cont'd)

BUFFER

An isolating circuit used to avoid reaction of a driven circuit on the corresponding driver circuit. Also, a storage device used to compensate for a difference in the rate of flow of information or the time of occurrence of events when transmitting information from one device to another.

BUS

One or more conductors used for transmitting signals or power.

BYTE

A sequence of 8 adjacent binary digits operated upon as a unit.

C

CALL

To transfer control to a specified closed subroutine.

CARRY

1. One or more digits, produced in connection with an arithmetic operation on one digit place of two or more numerals in positional notation, that are forwarded to another digit place for processing there.

2. The number represented by the digit or digits in definition 1 above.

C (Cont'd)

COMBINATORIAL LOGIC SYSTEM

Digital system not utilizing memory elements. A circuit arrangement in which the output state is determined by the present state of the input. Also called Cominbatorial logic. (See also SEQUENTIAL LOGIC.)

COMMUNICATION CONTROL CHARACTER

A control character intended to control or facilitate transmission of data over communication networks.

COMMUNICATION LINK

The physical means of connecting one location to another for the purpose of transmitting and receiving data.

COMPILE

To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one machine instruction for each symbolic statement, or both, as well as performing the function of an assembler.

COMPILER

A program that compiles.

CONDITION CODE

Refers to a limited group of program conditions such as carry, borrow, overflow, etc., which are pertinent to the execution of instructions. The codes are contained in a Condition Codes Register. (See also STATUS WORD REGISTER.)

C (Cont'd)

CONDITIONAL JUMP

A jump that occurs if specified criteria are met.

CONTROL BLOCK

This is the circuitry which performs the control functions of the CPU. It is responsible for decoding microprogrammed instructions, and then generating the internal control signals that perform the operations requested.

CONTROL BUS

Conveys a mixture of signals which regulate system operation. These "traffic" signals are commands which may also originate in peripherals for transfer to the CPU or the reverse.

CONTROL CHARACTER

A character whose occurrence in a particular context initiates, modifies, or stops a control operation, e.g., a character that controls carriage return, a character that controls transmission of data over communication networks. A control character may be recorded for use in a subsequent action. It may in some circumstances have a graphic representation.

CONTROLLER

Digital subsystem responsible for implementing "how" a system is to function. Not to be confused with "timing" as timing tells the system "when" to perform its function.

C (Cont'd)

3. Most commonly, a digit as defined in definition 1 above that arises when the sum or product of two or more digits equals or exceeds the radix of the number representation system.

4. Less commonly, a borrow.

5. To forward a carry.

6. The command directing that a carry be forwarded.

CARRY LOOK-AHEAD

A type of adder in which the inputs to several stages are examined and the proper carries are produced simultaneously.

CENTRAL PROCESSOR UNIT (CPU)

Part of a computer system which contains the main storage, arithmetic unit, and special register groups. It performs arithmetic operations, controls instruction processing, and provides timing signals and other housekeeping operations.

CHANNEL

1. A path along which signals can be sent, e.g., data channel, output channel.

2. The portion of a storage medium that is accessible to a given reading or writing station, e.g., track, band.

C (Cont'd)

CHARACTER

A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. A character is often in the form of a spatial arrangement of adjacent or connected strokes.

CHECK BIT

A binary check digit, e.g., a parity bit.

CLOCK

A generator of pulses which controls the timing of switching circuits in a microprocessor. Clock frequency is not the only criterion of data manipulation speed. Hardware architecture and programming skill are more important. Clocks are a requisite for most microprocessors and multiple phased clocks are common in MOS processors.

CODE

1. A set of unambiguous rules specifying the way in which data may be represented, e.g., the set of correspondences in the standard code for information interchange. Synonymous with coding scheme.

2. In telecommunications, a system of rules and conventions according to which the signals representing data can be formed, transmitted, received, and processed.

3. In data processing, to represent data or a computer program in a symbolic form that can be accepted by a data processor.

C (Cont'd)

CONTROL PROGRAM

The Control Program is a sequence of instructions that will guide the CPU through the various operations it must perform. This program is stored permanently in ROM memory where it can be accessed by the CPU during operations.

COUNTER

A circuit which counts input pulses and will give an output pulse after receiving a predetermined number of input pulses.

CPU (CENTRAL PROCESSING UNIT)

The heart of any computer system. Basically the CPU is made up of storage elements called registers, computational circuits in the ALU, the Control Block, and I/O. As soon as LSI technology was able to build a CPU on an IC chip, the microprocessor became a reality. The one-chip microprocessors have limited storage space, so memory implementation is added in modular fashion. Most current microprocessors consist of a set of chips, one or two of which form the CPU.

CRC

The Cyclic Redundancy Check character.

CROM (CONTROL READ ONLY MEMORY)

This is a major component in the control block of some microprocessors. It is a ROM which has been microprogrammed to decode control logic.

C (Cont'd)

CROSS-ASSEMBLER

When the program is assembled by the same computer that it will run on, the program that performs the assembly is referred to as the resident or native or self-assembler. If the program is assembled by some other computer, the process is referred to as cross-assembly.

CYCLE

1. An interval of space or time in which one set of events or phenomena is completed.

2. Any set of operations that is repeated regularly in the same sequence. The operations may be subject to variations on each repetition.

D

DAISY CHAIN

A bus line which is interconnected with units in such a way that the signal passes from one unit to the next in serial fashion. The architecture of the Fairchild F-8 provides an example of daisy-chained memory chips. Each chip connects to its neighbors to accomplish daisy-chaining of interrupt priorities beginning with the chip closest to the CPU.

DATA

Information in numerical code which is assigned an address in memory that the CPU uses when storing or fetching the information.

D (Cont'd)

DATA BUS

The microprocessor communicates internally and externally by means of the data bus. It is bidirectional and can transfer data to and from the CPU, memory storage, and peripheral devices.

DATA PROCESSING

The execution of a systematic sequence of operations performed upon data. Synonymous with information processing.

DATA PROCESSOR

A device capable of performing data processing, including desk calculators, punched card machines, and computers. Synonymous with processor.

DEBUG

To detect, locate, and remove mistakes from a routine or malfunctions from a computer. Synonymous with troubleshoot.

D-BUS (See DATA BUS)

DECIMAL

1. Pertaining to a characteristic or property involving a selection, choice, or condition in which there are ten possibilities.

2. Pertaining to the number representation system with a radix of ten.

DECIMAL DIGIT

In decimal notation, one of the characters 0 through 9.

D (Cont'd)

DECODER

A conversion circuit that accepts digital input information - in the memory case, binary address information - that appears as a small number of lines and selects and activates one line of a large number of output lines.

DECREMENT

A programming instruction which decreases the contents of a storage location. (See also INCREMENT and DECREMENT.)

DEDICATED

To set apart for some special use. A dedicated microprocessor is one that has been specifically programmed for a single application such as weight measurement by scale, traffic light control, etc. ROMs by their very nature (Read-Only) are "dedicated" memories.

DIAGNOSTIC

Pertaining to the detection and isolation of a malfunction or mistake.

DIGIT

A symbol that represents one of the non-negative integers smaller than the radix. For example, in decimal notation, a digit is one of the characters from 0 to 9. Synonymous with numeric character.

DIGITIZE

To use numeric characters to express or represent data, e.g., to obtain from an analog representation of a physical quantity, a digital representation of the quantity.

D (Cont'd)

DIRECT ACCESS

1. Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage.

2. Pertaining to a storage device in which the access time is effectively independent of the location of the data.

3. Synonymous with random access.

DIRECT ADDRESSING

Method of programming that has the address pointing to the location of data or the instruction that is to be used.

DIRECT MEMORY ACCESS CHANNEL (DMA)

A method of input-output for a system that uses a small processor whose sole task is that of controlling input-output. With DMA, data are moved into or out of the system without program intervention.

DOT MATRIX

A matrix of dots that is used to identify alphanumeric characters.

DOUBLE PRECISION

Pertaining to the use of two computer words to represent a number.

D (Cont'd)

DUMP

1. To copy the contents of all or part of a storage, usually from an internal storage into an external storage.

2. A process as in definition 1 above.

3. The data resulting from the process as in definition 1 above.

DUPLEX

The method of operation of a communication circuit in which each end can simultaneously transmit and receive.

E

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-bit, 256-character code used in transmission of binary data.

ECL CIRCUITS

Bipolar emitter-coupled logic circuits, also called current-mode logic circuits.

EDGE TRIGGERING

Activation of a circuit at the edge of the pulse as it begins its change. Circuits then trigger at the edge of the input pulse rather than sensing a level change.

E (Cont'd)

EDIT

To modify the form or format of data, e.g., to insert or delete characters such as page numbers or decimal points.

ELECTROSTATIC STORAGE

A storage device that stores data as electrostatically charged areas on a dielectric surface.

EMULATE

To imitate one system with another such that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated system.

ENCODE

To apply a set of unambiguous rules specifying the way in which data may be represented such that a subsequent decoding is possible. Synonymous with code.

END-AROUND CARRY

A carry generated in the most significant digit place and sent directly to the least significant place.

ENTRY POINT

In a routine, any place to which control can be passed.

ERASE

To obliterate information from a storage medium, e.g., to clear, to overwrite.

F

FAN-OUT.

The number of loads connected to the output of a logic stage. (A load normally consists of the input impedance of a logic circuit.)

FEEDBACK LOOP

The components and processes involved in correcting or controlling a system by using part of the output as input.

FEEDBACK SYSTEM

See INFORMATION FEEDBACK SYSTEM.

FETCH

To go after and return with things or objects. In a microprocessor, the "objects" fetched are instructions and data which are entered in the Instruction and Data Registers. The next, or a later step in the program will cause the machine to execute what it was programmed to do with the fetched instructions and data.

FIELDS

A source statement is made up of a number of code fields, usually four, which are acceptable by the assembler. The four fields may connote Label, Operator, Operand, and Comment. Fields are also applicable to data storage. The eight bits stored in a memory location might contain two 4-bit fields, or eight 1-bit fields, etc.

FIRMWARE

(See SOLID STATE SOFTWARE.)

F (Cont'd)

FIXED-POINT BINARY NUMBER

A binary number represented by a sign bit and one or more number bits, with a binary point fixed somewhere between two neighboring bits.

FLAG

1. Any of various types of indicators used for identification, e.g., a wordmark.

2. A character that signals the occurrence of some condition, such as the end of a word.

3. Synonymous with mark, sentinel, tag.

FLAG BIT

An information bit which indicates some form of demarcation has been reached such as overflow or carry. Also an indicator of special conditions such as interrupts.

FLIP-FLOP (STORAGE ELEMENT)

A circuit having two stable states and the capability of changing from one state to another with the application of a control signal and remaining in that state after removal of signals.

FLOATING-POINT BINARY NUMBER

A binary number expressed in exponential notation. That is, a part of the binary word represents the mantissa and a part the exponent.

F (Cont'd)

FLOW CHART OR FLOW DIAGRAM

A sequence of operations charted with the aid of symbols, diagrams, or other representations to indicate an executive program. Flowcharts enable the designer to visualize the procedure necessary for each item on the program. A complete flowchart leads directly to the final code.

FORMAT

The arrangement of data in a usable form.

FORTRAN

(FORmula TRANslating system) A language primarily used to express computer programs by arithmetic formulas.

FULL-ADDER

A logic circuit like the half-adder, but with a provision for a carry-in from a preceding addition.

FUNCTION

1. A specific purpose of an entity, or its characteristic action.

2. In communications, a machine action such as a carriage return or line feed.

G

GENERAL-PURPOSE COMPUTER

A computer that is designed to handle a wide variety of problems.

APPENDIX

GLOSSARY OF BUZZ WORDS

H

HALF-ADDER

A logic circuit capable of adding two binary numbers with no provision for a carry-in from a preceding addition.

HANDSHAKING

A colloquial term which describes the method used by a modem to establish contact with another Modem at the other end of a telephone line. Often used interchangeably with buffering and interfacing, but with a fine line of difference in which handshaking implies a direct package to package connection regardless of functional circuitry.

HARDWARE

The individual components of a circuit, both passive and active, have long been characterized as hardware in the jargon of the engineer. Today, any piece of data processing equipment is informally called hardware. Physical equipment, as opposed to the computer program or method of use, e.g., mechanical, magnetic, electrical, or electronic devices.

HARD-WIRED LOGIC

Random Logic design solutions require interconnection of numerous integrated circuits representing the logic elements. An example of hard-wired logic is the use of a hand-wired diode matrix instead of a ROM. These interconnections, whether done with soldering iron or by printed circuit board, are referred to as hard-wired logic in contrast to the software solutions achieved by a programmed ROM or Microprocessor.

H (Cont'd)

HIGH-LEVEL LANGUAGE

This is a problem-oriented programming language as distinguished from a machine-oriented programming language. The former's instruction approach is closer to the needs of the problems to be handled than the language of the machine on which they are to be implemented.

HEXADECIMAL

Whole numbers in positional notation using 16 as a base. (See Octal and compare.) Since there are 16 hexadecimal digits (0 through 15) and there are only ten numerical digits (0 through 9) an additional six digits representing 10 through 15 must be introduced. Recourse is had to the alphabet to provide the extra digits. Hence, the least significant hexadecimal digits read: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. The decimal number 16 becomes the hexadecimal number 10. The decimal number 26 becomes the hexadecimal number 1A.

I

IMMEDIATE ADDRESS

Pertaining to an instruction in which an address part contains the value of an operand rather than its address. Synonymous with zero-level address.

INCREMENT (AND DECREMENT)

These two words are software operations most often associated with the stack and stack pointer. Bytes of information are stored in the stack register at the addresses contained in the stack pointer. The stack

I (Cont'd)

pointer is decremented after each byte of information is entered into the stack; it is incremented after each byte is removed from the stack. The terms can also refer to any addressable register.

INDEXED ADDRESS

An address that is modified by the content of an index register prior to or during the execution of a computer instruction.

INDEXING

In computers, a method of address modification that is implemented by means of index registers.

INDEX REGISTER

A register whose content may be added to the operand address prior to or during the execution of a computer instruction.

INDIRECT ADDRESSING

Programming method that has the initial address being the storage location of a word that contains another address. This indirect address is then used to obtain the data to be operated upon.

INITIALIZE

The reset control unit is used to start execution of program for initial startup. At the beginning of each new program, initialization requires re-setting all hardware controls to starting values.

I (Cont'd)

INPUT/OUTPUT DEVICES (I/O)

Computer hardware by which data is entered into a digital system or by which data are recorded for immediate or future use.

INSTRUCTION

One of a number of computer operations stored in memory that may, when called upon, command the CPU to perform arithmetic and logic functions, control peripheral devices, or indicate succeeding instructions.

INSTRUCTION COUNTER

A counter that indicates the location of the next computer instruction to be interpreted.

INSTRUCTION REGISTER

A register that stores an instruction for execution.

INSTRUCTION SET

Constitutes the total list of instructions which can be executed by a given microprocessor and is supplied to the user to provide the basic information necessary to assemble a program.

INTERFACE

A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

I (Cont'd)

INTERLEAVE (OR INTERLACE)

To assign successive storage location numbers to physically separated memory storage locations. This serves to reduce access time.

INTERPRETER

A method or program that translates high level language into machine assembleable information. (See also Machine and High Level Language.)

INTERRUPT

An interrupt involves the suspension of the normal programming routine of a microprocessor in order to handle a sudden request for service. The importance of the interrupt capability of a microprocessor depends on the kind of applications to which it will be exposed. When a number of peripheral devices interface the microprocessor, one or several simultaneous interrupts may occur on a frequent basis. Multiple interrupt requests require the processor to be able to accomplish the following: to delay or prevent further interrupts; to break into an interrupt in order to handle a more urgent interrupt; to establish a method of interrupt priorities; and, after completion of interrupt service, to resume the interrupted program from the point where it was interrupted. (See also VECTOR INTERRUPT.)

INTERRUPT MASK BIT

The Interrupt Mask Bit prevents the CPU from responding to further interrupt requests until cleared by execution of programmed instructions. It may also be manipulated by specific mask bit instructions.

I (Cont'd)

I/O (INPUT/OUTPUT)

Package pins which are tied directly to the internal bus network to enable I/O to interface the microprocessor with the outside world.

J

JUMP

The Jump operation, like the Branch operation is used to control the transfer of operations from one point to a more distant point in the control program. Jumps differ from Branching in not using the Relative Addressing mode.

JUMP CONDITIONS

Conditions defined in a transition table that determine the changes of flip-flops from one state to another state.

L

LABEL

A label may correspond to a numerical value or a memory location in the programmable system. The specific absolute address is not necessary since the intent of the label is a general destination. Labels are a requisite for jump and branch instructions.

LANGUAGE

A set of representations, conventions, and rules used to convey information.

L (Cont'd)

LARGE-SCALE INTEGRATION (LSI)

The simultaneous realization of large-area chips and optimum component packing density, resulting in cost reduction by maximizing the number of system connections done at the chip level. Circuit complexity above 100 gates.

LEVEL

The degree of subordination in a hierarchy.

LIBRARY

A collection of complete programs written for a particular computer, minicomputer, or microprocessor. For example, Second Order Differential Equation may be the name of a program in the Library of a particular computer; this program will contain all the subroutines necessary to perform the solution of second order differential equations written in machine language and using the instruction set of this machine.

LIFO

Last-In-First-Out. (See PUSH-POP Stack.)

LINK REGISTER

This is a register of 1 bit (sometimes called the Carry register) which acts as an extension of the Accumulator during rotation or carry operations.

L (Cont'd)

LOGIC

A mathematical treatment of formal logic in which a system of symbols is used to represent quantities and relationships. The symbols or logical functions are called AND, OR, NOT, to mention a few examples. Each function can be translated into a switching circuit, more commonly referred to as a "gate." Since a switch (or gate) has only two states - open or closed - it makes possible the application of binary numbers for the solution of problems. The basic logic functions obtained from gate circuits is the foundation of complex computing machines.

LOGIC SHIFT

A shift that affects all positions.

LOOK AHEAD

1. A feature of the CPU which allows the machine to mask an interrupt request until the following instruction has been completed.

2. A feature of adder circuits and ALUs which allow these devices to look ahead to see that all carrys generated are available for addition.

LOOP

A sequence of instructions that is executed repeatedly until a terminal condition prevails.

L (Cont'd)

LOOPING

Repetition of instructions at delayed speeds until a final value is determined (as in a weight scale indication) is called looping. The looped repetitions are usually frozen into a ROM memory location and then jumped to when needed. Looping also occurs when the CPU is in a wait condition.

LSB

Least Significant Bit. (See SIGNIFICANT BITS.)

LSI (LARGE SCALE INTEGRATION)

At the beginning of the LSI era a count of 100 gates qualified for LSI. Today an 8-bit CPU can be fabricated on a single chip.

M

MACHINE CODE

An operation code that a machine is designed to recognize.

MACHINE LANGUAGE

The only language the microprocessor can understand is binary. All other programming languages must be translated into binary code before entering the processor and decoded back into the original language after leaving it.

M (Cont'd)

MACRO COMMAND

A program entity formed by a string of standard, but related, commands which are put into effect by means of a single macro command. Any group of frequently used commands can be combined into a macro command. The many become one.

MACROINSTRUCTION

An instruction in a source language that is equivalent to a specified sequence of machine instructions.

MACROPROGRAMMING

Programming with macroinstructions.

MAIN FRAME

Same as Central Processing Unit.

MASK

1. A pattern of characters that is used to control the retention or elimination of portions of another pattern of characters.

2. A filter.

MATRIX

1. In mathematics, a two-dimensional rectangular array of quantities. Matrices are manipulated in accordance with the rules of matrix algebra.

M (Cont'd)

2. In computers, a logic network in the form of an array of input leads and output leads with logic elements connected in some of their intersections.

3. By extension, an array of any number of dimensions.

MICROPROGRAMMING

Control technique used to implement the stored program control function. Typically the technique is to use a preprogrammed read-only memory chip to contain several control sequences which normally occur together.

MNEMONIC CODE

These are designed to assist the human memory. The microprocessor language consists of binary words which are a series of 0's and 1's making it difficult for the programmer to remember the instructions corresponding to a given operation. To assist the human memory, the binary numbered codes are assigned groups of letters (or mnemonic symbols) that suggest the definition of the instruction. LDA for load accumulator, etc. Source statements can be written in this symbolic language and then translated into machine language.

MNEMONIC SYMBOL

A symbol chosen to assist the human memory, e.g., an abbreviation such as "mpy" for "multiply".

MICROCOMPUTER

(See MICROPROCESSOR.)

M (Cont'd)

MICROPROCESSOR

The microprocessor is a Central Processing Unit fabricated on one or two chips. While no standard design is visible in existing units, a number of well-delineated areas are present in all of them: Arithmetic & Logic Unit, Control Block, and Register Array. When joined to a memory storage system and peripheral I/O's, the resulting combination is referred to in today's usage as a Microcomputer. It should be added that each microprocessor is supplied with an Instruction Set listing the basic operations which the processor performs.

MICROPROGRAM

This word pre-dates the microprocessor and refers to computer instructions which do not reference the main memory storage. It is a computer technique which performs subroutines by manipulating the basic computer hardware and is often referred to as "computer within computer." The word has not changed its basic meaning when used in connection with microprocessors. A series of instructions stored in a ROM, any portion of which can implement a higher language program, is labeled a microprogram.

MICROINSTRUCTION: (See MICROPROGRAM)

M (Cont'd)

MEMORY

The part of a computer system into which information can be inserted and held for future use. Storage and memory are interchangeable expressions. Memories accept and hold binary numbers only. Memory types are core, disk, drum, and semiconductor.

MODEM

(MOdulator-DEModulator) A device that modulates and demodulates signals transmitted over communication facilities.

MOS TRANSISTOR (METAL-OXIDE-SEMICONDUCTOR TRANSISTOR)

An active semiconductor device in which a conducting channel is induced in the region between two electrodes by a voltage applied to an insulated electrode on the surface of the region.

MOS (METAL OXIDE SEMICONDUCTOR)

The structure of an MOS Field Effect Transistor (FET) is metal over silicon oxide over silicon. The metal electrode is the gate; the silicon oxide is the insulator, and carrier doped regions in the silicon substrate become the drain and source. The result is a sandwich very much like a capacitor, which explains why MOS is slower than bipolar since the 'capacitor sandwich' must charge up before current can flow. The three great advantages of MOS are its process simplicity because of reduced fabrication stages; the savings in chip real estate resulting in functional density; and the ease of interconnection on chip. These qualities enabled MOS to break the LSI barrier, something bipolar is

M (Cont'd)

just beginning to achieve. The hand-held calculator and the microprocessor are triumphs of MOS-LSI technology.

MSB

Most Significant Bit. (See Significant Bits.)

MULTIPLEX

To interleave or simultaneously transmit two or more messages on a single channel.

MULTIPLEXING

Multiplexing describes a process of transmitting more than one signal at a time over a single link, route, or channel. Of the two methods in use, one frequency-shares the bandwidth of a channel in the same way hurdlers run and jump in their assigned lanes thus permitting many contestants to compete simultaneously on the same track. The second way is to time-share multiple signals in the same way that pole vaulters jump over the same bar one after the other. The two methods may be described as parallel and serial processing. Time-sharing may not seem "simultaneous," but it should be remembered that the signal speed is so fast that it is possible to multiplex four different numbers through a single decoder-driver and have them appear on four different displays without a flicker to disturb the eye.

N

NEGATIVE LOGIC

Logic in which the more-negative voltage represents the "1" state; the less-negative voltage represents the "0" state.

N (Cont'd)

NESTING

Nesting is referred to when a subroutine is enclosed inside a larger routine, but is not necessarily part of the outer routine. A series of looping instructions may be nested within each other.

NOISE

A term referring to spurious or undesirable electrical signals.

NONDESTRUCTIVE READ OUT

A memory designed so that read-out does not affect the content stored. It is not necessary to perform a write after every read operation.

O

OBJECT CODE

Output from a compiler or assembler which is itself executable machine code or is suitable for processing to produce executable machine code.

OBJECT LANGUAGE

The language to which a statement is translated.

OBJECT PROGRAM

The end result of the source language program after it has been translated into machine language.

OCTAL

Whole numbers in positional notation using 8 as a base. The decimal or base 10 number, 125, becomes 175 in octal or base 8. Here is a

O (Cont'd)

convenient way to convert a decimal number into an octal number:

```
    1   7
8  15   5
8 125
```

Divide the decimal number by 8. The answer is 15 and 5 left over.

Divide the answer, 15, by 8 again. The answer is 1 and 7 left over.

The octal number is 175.

To prove your answer is correct, do the following:

```
5 x  1 =   5
7 x  8 =  56
1 x 64 =  64
        125
```

Arrange the octal number vertically with the significant digit on top.

The least significant digit represents one's, so multiply 5 x 1 = 5.

The next digit in the octal number represents 8's, so multiply 7 x 8 = 56.

The third digit of the octal number represents 64's, so multiply 1 x 64 = 64.

The sum is the decimal number 125.

OPERATING SYSTEM

Software which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

OPERAND

A quantity on which a mathematical operation is performed. One of the instruction fields in an addressing statement. Usually the statement consists of an operator and an operand. The operator may indicate an "add" instruction; the operand will indicate what is to be added.

O (Cont'd)

OPERATION

1. A defined action, namely, the act of obtaining a result from one or more operands in accordance with a rule that completely specifies the result for any permissible combination of operands.

2. The set of such acts specified by such a rule, or the rule itself.

3. The act specified by a single computer instruction.

4. A program step undertaken or executed by a computer, e.g., addition, multiplication, extraction, comparison, shift, transfer. The operation is usually specified by the operator part of an instruction.

5. The even or specific action performed by a logic element.

OPERATION CODE (OPCODE)

A code that represents specific operations. Synonymous with instruction code. Source statements which generate machine codes after assembly are referred to as operating codes.

OVERFLOW

Overflow results when an arithmetic operation generates a quantity beyond the capacity of the register. Also referred to as arithmetical overflow. An overflow status bit in the condition code register can be checked to determine if the previous operation caused an overflow.

P

PACK

To compress data in a storage medium by taking advantage of known characteristics of the data, in such a way that the original data can be recovered, e.g., to compress data in a storage medium by making use of bit or byte locations that would otherwise go unused.

P (Cont'd)

PAGE

A natural grouping of memory locations by higher-order address bits. In an 8-bit MPU, $2^8 = 256$ consecutive bytes may constitute a page. Then words on the same page only differ in the lower order 8 address bits.

PAGE MODE ADDRESSING

A technique used in MPU's to directly address memory locations stored within the page. (See PAGE.)

PARALLEL OPERATION

Processing all the digits of a word or byte simultaneously by transmitting each digit on a separate channel or bus line.

PARAMETER

A variable that is given a constant value for a specific purpose or process.

PARITY BIT

A check bit appended to an array of binary digits to make the sum of all the binary digits, including the check bit, always odd or always even.

PARITY CHECK

The technique of adding one bit to a digital word to make the total number of binary ones or zeros either always even or always odd. This type of checking will indicate an error in data but will not indicate the location of the error.

PERIPHERAL EQUIPMENT

Units which work in conjunction with a computer but are not part of it.

P (Cont'd)

PIPELINE REGISTER

A register placed on the output of the microprogram memory to essentially split the system in two. This allows the micro-instruction fetch to occur in parallel with the data operation rather than serially, allowing the clock frequency to be doubled. Stack processing is done on a last-in-first-out basis while pipeline processing is first-in-first-out.

POINTER

A register which serves as a reference point to a memory location.

POLLING

Polling is the method used to identify the source of interrupt requests. When several interrupts occur at one time, the control program decides which one to service first.

PORT

Device terminals which provide electrical access to a system or circuit. The point at which the I/O is in contact with the outside world.

POSITIVE LOGIC

Logic in which the more positive voltage represents the "1" state; the less positive voltage represents the "0" state.

PRIORITY INTERRUPT (See INTERRUPT.)

Designation given to method of providing some commands to have precedence over others thus giving one condition of operation priority over another.

P (Cont'd)

PROCESSOR

1. In hardware, a data processor.
2. In software, a computer program that includes the compiling, assembling, translating, and related functions for a specific programming language, COBOL processor, or FORTRAN processor.

PROGRAM

A procedure for solving a problem and frequently referred to as Software.

1. A series of actions proposed in order to achieve a certain result.
2. Loosely, a routine.
3. To design, write, and test a program as in definition 1 above.
4. Loosely, to write a routine.

PROGRAM COUNTER

One of the registers in the CPU which holds addresses necessary to step the machine through the program. During interrupts, the program counter value containing the address of the next instruction is saved. Branching also requires loading of the return address in the program counter.

PROGRAMMABLE READ ONLY MEMORY (PROM) (See ROM ALSO.)

A fixed program, read only, semiconductor memory storage element that can be programmed after packaging.

PROM

(See PROGRAMMABLE READ ONLY MEMORY.)

P (Cont'd)

PROPAGATION DELAY

The time required for a change in logic level to be transmitted through an element or a chain of elements.

PUSHDOWN LIST

A list that is constructed and maintained so that the item to be retrieved is the most recently stored item in the list, i.e., last in, first out.

PUSHDOWN STACK

A register which implements a pushdown list.

PUSH-POP STACK

A register that receives information from the Program Counter and stores the address locations of instructions on a last-in-first-out basis. Two operations are involved in stack processing: "Pushing" describes the filling of the stack from register; "Popping" involves emptying the stack for transfer to registers.

P-STACK (See PUSH-POP STACK.)

PUSHUP LIST

A list that is constructed and maintained so that the next item to be retrieved and removed is the oldest item still in the list, i.e., first in, first out.

APPENDIX

GLOSSARY OF BUZZ WORDS

R

RAM

(See RANDOM ACCESS MEMORY.)

RANDOM ACCESS MEMORY (RAM)

A memory from which all information can be obtained at the output with approximately the same time delay by choosing an address randomly and without first searching through a vast amount of irrelevant data.

READ ONLY MEMORY (ROM)

A fixed program semiconductor storage element that has been preprogrammed at the factory with a permanent program.

REAL TIME

1. Pertaining to the actual time during which a physical process transpires.
2. Pertaining to the performance of a computation during the actual time that the related physical process transpires, in order that results of the computation can be used in guiding the physical process.

REAL TIME OPERATION

Data processing technique used to allow the machine to utilize information as it becomes available, as opposed to batch processing at a time unrelated to the time the information was generated.

REDUNDANCY

The technique of using more than one circuit of the same type to implement a given function.

R (Cont'd)

REFRESH

Because electrical charges momentarily applied to Dynamic memory circuits (representing data or instructions) leak off rapidly, periodic recharging is necessary to keep memory voltage levels intact and determinable.

REGISTER

A register is a memory on a smaller scale. The words stored therein may involve arithmetical, logical, or transferral operations. Storage in registers may be temporary, but even more important is their accessibility by the CPU. The number of registers in a microprocessor is considered one of the most important features of its architecture.

RELATIVE ADDRESSING

The relative addressing mode specifies a memory location referenced to the CPU's Program Location Counter register. This addressing mode is mainly used for Branch instructions in which case an opcode is added to or subtracted from the Program Counter to complete the branching instruction.

RESIDENT ASSEMBLER (See CROSS-ASSEMBLER.)

ROM (READ ONLY MEMORY)

In its virgin state the ROM consists of a mosaic of undifferentiated cells. One type of ROM is programmed by mask pattern as part of the last manufacturing stage. Another, more popular type better known as P/ROM, is programmable in the field with the aid of programmer equipment. Program data stored in

R (Cont'd)

ROMs are often called firmware because they cannot be altered. However, another type of P/ROM is now on the market called EPROM which is erasible by ultra violet irradiation and electrically reprogrammable. (See READ ONLY MEMORY also.)

S

SCRATCHPAD

This term is applied to information which the Processing unit stores or holds temporarily. It is a memory containing subtotals for various unknowns which are needed for final results.

SCRATCH-PAD MEMORY

A small local memory utilized to facilitate local data handling on a temporary basis.

SEQUENCING

Control method used to cause a set of steps to occur in a particular order.

SEQUENTIAL LOGIC

A circuit arrangement in which the output state is determined by the previous state of the input. (See also COMBINATION LOGIC.)

SERIAL OPERATION

The organization of data manipulation within circuitry wherein the digits of a word are transmitted one at a time along a single line. The serial mode of operation is slower than parallel operation, but utilizes less complex circuitry.

S (Cont'd)

SHIFT

A movement of data to the right or left.

SHIFT REGISTER

A register in which the stored data can be moved to the right or left.

SIGNED NUMBERS (See TWO's COMPLEMENT NUMBERS.)

SIGNIFICANT BITS

The most significant bit (MSB) in a byte of 8 numbers is on the extreme left. The least significant bit (LSB) is on the extreme right. The remaining bits are weighted according to their position in the byte between the MSB and the LSB.

SIMULATE

1. To represent certain features of the behavior of a physical or abstract system by the behavior of another system.

2. To represent the functioning of a device, system, or computer program by another, e.g., to represent the functioning of one computer by another, to represent the behavior of a physical system by the execution of a computer program, to represent a biological system by a mathematical model.

SIMULATOR

A device, system, or computer program that represents certain features of the behavior of a physical or abstract system.

S (Cont'd)

SKIP

To ignore one or more instructions in a sequence of instructions.

SNAPSHOT

A CRT representation of a contiguous portion of a machine's memory addresses and data contained therein. A series of snapshots can capture the entire state of a machine including memory contents, registers, flags, etc.

SOFTWARE

What sheet music is to the piano, software is to the computer. Looked at from a practical point of view, one might say that software is the computer's instruction manual. The name, software, was obviously chosen to contrast with the formidable hardware which confronted the first programmers. Software is the language used by a programmer to communicate with the computer. Since the only language spoken by a computer is mathematical, the programmer must convert his verbal instructions into numbers. In the case of microprocessors, which vary from maker to maker, software libraries are assembled by the manufacturer for the benefit of the user.

SOURCE LANGUAGE

The language from which a statement is translated.

SOURCE PROGRAM

A computer program written in a source language.

APPENDIX

GLOSSARY OF BUZZ WORDS

S (Cont'd)

SOURCE STATEMENT

A program written in other than machine language, usually in three-letter mnemonic symbols, that suggest the definition of the instruction. There are two kinds of source statements: "executive instructions" which translate into operating machine code (opcode); and "assembly directives" which are useful in documenting the source program, but generate no code.

STACK

The stack is a block of successive memory locations which is accessible from one end on a last-in-first-out basis (LIFO). The stack is co-ordinated with the stack pointer which keeps track of storage and retrieval of each byte of information in the stack. A stack may be any block of successive information locations in the read/write memory.

STACK POINTER

The stack pointer is coordinated with the storing and retrieval of information in the stack. The stack pointer is decremented by one immediately following the storage in the stack of each byte of information. Conversely, the stack pointer is incremented by one immediately before retrieving each byte of information from the stack. The stack pointer may be manipulated for transferring its contents to the Index register or vice versa.

S (Cont'd)

STATE

The condition of an input or output of a circuit as to whether it is a logic "1" or a logic "0". The state of a circuit (gate or flip-flop) refers to its output. A flip-flop is said to be in the "1" state when its Q output is "1". A gate is in the "1" state when its output is "1".

STATUS WORD REGISTER

A group of binary numbers which informs the user of the present condition of the microprocessor. In most microprocessors, the Status Register provides the following five pieces of information: plus or minus sign of the value in Accumulator, overflow indication, carry bit, all zero's in accumulator, and interrupt bit status. (See also Condition Code Register. Also called Program Status Word.)

STORAGE

The word storage is used interchangeably with memory. In fact, it has been recommended as the preferred term by people who would rather not imply that the computer has any relationship with the human brain.

STORED PROGRAM

A set of instructions in memory specifying the operation to be performed.

SUBROUTINE

Part of a master routine which may be used at will in a variety of master routines. The object of a Branch or Jump command.

S (Cont'd)

SYNCHRONOUS CIRCUIT

A circuit in which all ordinary operations are controlled by equally spaced signals from a master clock.

SYSTEM

1. An assembly of methods, procedures, or techniques united by regulated interaction to form an organized whole.

2. An organized collection of men, machines, and methods required to accomplish a set of specific functions.

T

TABLE LOOK-UP

A procedure for obtaining the function value corresponding to an argument from a table of function values.

TEMPORARY STORAGE

In programming, storage locations reserved for intermediate results. Synonymous with working storage.

TERMINAL

A point in a system or communication network at which data can either enter or leave.

T (Cont'd)

THROUGHPUT

The speed with which problems or segments of problems are performed is called Throughput. Defined in this way, it is obvious that throughput will vary from application to application. As an index of speed, throughput is meaningful only in terms of your own application.

TRANSFER

Same as jump.

TRANSLATE

To transform statements from one language to another without significantly changing the meaning.

TRUTH TABLE

A chart that tabulates and summarizes all the combinations of possible states of the inputs and outputs of a circuit. It tabulates what will happen at the output for a given input combination.

TTL

Bipolar semiconductor transistor-transistor coupled logic circuits.

TWO'S COMPLEMENT NUMBERS

The ALU performs standard binary addition using the 2's complement numbering system to represent both positive and negative numbers. The positive numbers in 2's complement representation are identical to the positive numbers in standard binary.

T (Cont'd)

+ 127 in standard binary = 01111111 + 127 in 2's complement = 01111111.

Note that the eighth or most significant digit indicates the sign: 0 = plus; 1 = minus.

However, the negative 2's complement is the reverse of the negative standard binary plus 1.

- 127 in standard binary = 11111111. To form the 2's complement of - 127.

First reverse all the digits except the sign

= 10000000

Then add 1 1

10000001 = - 127 in 2's complement.

U

UART (UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER)

This device will interface a word parallel controller or data terminal to a bit serial communication network.

USASCII

United States of America Standard Code for Information Interchange. The standard code used by the United States for transmission of data. Sometimes simply referred to as the "as'ki" code.

# APPENDIX

## GLOSSARY OF BUZZ WORDS

V

VARIABLE

A quantity that can assume any of a given set of values.

VECTOR INTERRUPT

This term is used to describe a microprocessor system in which each interrupt, both internal and external, have their own uniquely recognizable address. This enables the microprocessor to perform a set of specified operations which are pre-programmed by the user to handle each interrupt in a distinctively different manner.

VOLATILE STORAGE

A storage device in which stored data are lost when the applied power is removed.

W

WORD

A group of "characters" treated as a unit and given a single location in computer memory. Presumably a byte is a group of 8 bits in contrast to a word which is a group of numeric and/or alphabetic characters and symbols, but the two words are used interchangeably more often than not.

W (Cont'd)

WRITE ENABLE

Also called read/write or R/W. The control signal to a storage element or a memory that activates the write mode or operation. Conversely when not in the write mode, the read mode is active.

WRITE TIME

The time that the appropriate level must be maintained on the write-enable line and that data must be present to guarantee successful writing of data in the memory.