CONTROL DATA CORPORATION

CYBER-70 and 6000

PP COMPASS SCOPE

CLASS TRAINING MANUAL

Advanced Software Education
CONTROL DATA INSTITUTE

DE4020-1

COURSE FH 4020

KRONOS/NOS PPU COMPASS

STUDENT HANDOUT

CONTROL DATA
EDUCATION
INSTITUTES

No. ~~A6002~~ DE 402

EDUCATION

6000 PERIPHERAL PROCESSOR (PP) COMPASS

PURPOSE

This course is designed to give the system programmer the capability of writing and debugging peripheral processor (PP) programs that interface with the SCOPE operating system.

OBJECTIVE

Upon completion the student will be expected to write a simple PP program of sufficient completeness to be executable in a running SCOPE system. He will be expected to understand the uses of monitor functions, PP resident routines, PP system macros, SCPTEXT symbols, PP memory layout, and PP interfacing with both central memory and peripheral equipment.

DESCRIPTION

This is a lecture/laboratory course where the student will be expected to write a PP program of sufficient completeness capable of being EDITLIBed into a running SCOPE operating system.

PREREQUISITE

6000 SCOPE Analysis (~~E6002~~ DE401) or ~~6000 SCOPE Workshop (E6004)~~

COURSE NO.	LENGTH	MAXIMUM STUDENTS
~~A6002~~ DE402	5 days	16

CONTENTS

Introduction

PP Hardware Review

Instruction Repertoire
- Assembler Mnemonics
- Pseudoinstructions
- Conditional Operators
- Data Channel Interface

PP Memory Layout
- Direct Cells
- PP Resident
- Transient Programs
- Overlays

Monitor Functions

SCPTEXT

PP Resident Routines

PP System Macros

Peripheral Equipment Interface
- Converters
- Console Display
- 3000 Type Controllers

PPU COMPASS WORKBOOK

CONTENTS

PPU COMPASS

COURSE MATERIALS

(12) ITEM NO.	(14) FORM OR PUBLICATION NO.	(15) CATALOG DESCRIPTION (INCLUDE PRODUCT DESIGNATION)
1	60183300	ASPER/COMPASS TRAINING GUIDE
2	60347400	CYBER SYSTEM DESCR VOL.1 CYBER [illegible]
3	60347300	INSTRUCTION DESCR VOL.2
4	60352500	CHANNEL I/O SPECS
5	60334400	6681/4 CHNL. CONV.
6	60333900	6602 CONSOLE DISPLAY
7	60375300	6642 DDP
8	60334600	6671 DATA SET CTLR.
9	60334500	6673/4 " "
10	38706000	6676 TTY MUX (A)
11	60332300	3248/3447/3549 CARD READ CTLR
12	60332100	3446/3644 CARD PUNCH CTLR
13	60237600	3528 MAG TAPE CTLR
14	60364400	7054/844 DISK & CTLR (GIM)
15	60353900	7054/844 OPTN. AND PROG.
16	60373500	3553 MASS STORAGE CTLR
17	60231300	3555 LINE PRINT CTLR
18	60361700	COMPASS INSTR. CARD
19	60164500	CP/PP INSTR. CARD
20	60361000	COMPASS 3. INSTANT
21	60347100	ECS REF MNL.
22	60087500	2" 3-RING BINDER
23	AA2987	CYBER CODING PADS

CYBER 70/6000 SERIES PERIPHERAL PROCESSOR COMPASS

(A6002)

Days	1	2	3	4	5
	Introduction Coding Format Hardware Review Inter-PP Instruction Set •Formats •Addressing Modes •The Non-I/O Instruction Set •CM I/O	Coding Conventions PP Macros System Tables And Pointers PP Resident	Monitor Functions	I/O Concepts I/O Instructions and Examples	Deadstart I/O Advanced Coding (Relocatable Overlays)

K - KRONOS 2.1 WORKSHOP MANUAL
H - 170 Hardware ref. man.
sec - Handout

CYBER 70/6000 SERIES PERIPHERAL PROCESSOR COMPASS

{A6002}

Days	1	2	3	4	5
projects →	Introduction Coding section 2 Format	SYSTEXT INSTANT Coding Conventions	Monitor Functions I 4-2 K3-10 SYSTEXT	~~sec~~ K20-1 I/O Concepts H5-30 section 10 H5-43 INSTANT 4-36 INSTANT - Funcodes	Deadstart I/O K24-1
	Hardware Review K1-3 H2-46 sec 3	SYSTEXT PP Macros K20-1		K8-1 I/O Instructions and Examples status/c reg. H5-32 DDP pg11 sec. 3 H4-73 H5-30 H5-42	Advanced Coding (Relocatable Overlays) K18-1
	Inter-PP Instruction Set • Formats • Addressing Modes • The Non-I/O Instruction Set • CM I/O H4-53 sec 4	System Tables And Pointers SYSTEXT K2-2 * INSTANT K2-9 *			
		PP Resident K4-1 I 3-39 PPR			

SECTION ONE

PROJECTS

SECTION ONE - ASSIGNMENTS

CONTENTS

PP CODING CLASS

ASSIGNMENTS

1. WORK THE EXERCISES (INSTRUCTIONS)
   - a. Problem Sets 1 and 2 - page 18
   - b. Problem Set 3 - page 19
   - c. Problem set 4 - page 27

2. STUDY TWO CHOSEN CODE EXAMPLES
   - a. 1AJ's RA+70 area code - page 33
   - b. PP Resident Routine R.DFM - page 34 or page 122a

3. WRITE A PP PROGRAM

   Choose one from the list of "Projects" -

   Preferred choices are #2 or #4

   Code the program, keypunch it, assemble it, and then prepare the deck for an Editlib run

PROJECTS

PP CODING CLASS
(Choose One)

Given:

1. Two data words in a central memory program:

```
BUF      DATA      123,456
```

a) Write a PP program to get these two numbers, add them together, and return the sum to the CP program at:

```
ANS      BSSZ      1
```

b) Write the CP program to check it out. Establish all the linkage necessary to call the PP program , receive the answer, and dump it out.

> Note:
>
> The CP and PP programs must obviously communicate with each other via all legal paths. The student must code them properly. However, the student does not have to write the Editlib control cards which would be necessary to put his PP program in the system library.

2. Write a PP Program to find the CP Program's Control Point number. Print the Job Name and CP Number in the job's dayfile. Write a calling CP routine to check it out.

3. Put #2 in 1EJ as a mod.

4. Write a relocatable PP overlay; and a transient program to call it, relocate it, and execute it.

5. Add a new control card. CTO (Comment to Operator). Modify 1AJ to process the new CTO card.

6. Write 2 PP Programs to communicate via the PP Interlock Register.

7. Modify MTR and R.IDLE to use the Interlock Register (#6 Mod).

SECTION TWO

CODING FORMAT

SECTION TWO - CODING FORMAT

CONTENTS

no

CODING FORMAT

1. On the coding form, PP programs are coded in the same columns as CP programs.

2. There is no entry point in a PP program.

   The code is ORG'd to begin at an absolute location in the PP -

   Usually, ie, 1000 - Transient Programs
   {also called Primary Overlays}

   2000 - Secondary Overlays
   3000,
   4000,
   ⌇
   7000...

3. The PP instruction set is used.

4. The same pseudo ops as in CP Compass are all available and are used extensively, especially conditional assembly.

5. The PP program must follow special coding rules -

   To interface with the system and not
   - clobber PP Resident
   - hang the PP
   - clobber central memory
   - steal channels or hang channels

6. To run the PP program and test it, it must be Editlibed into the running system.

   A CP test program must be written to call it.

2/72

SETTING UP THE CALLING CP PROGRAM

```
                IDENT     CPPGM

                ENTRY     TEST

TEST              }

                  code

                  }

                SA1       CALL      put call in RA+1

                BX6       X1

                SA6       1

WT              SA1       1         wait till picked up
                                    by MTR
                NZ        X1,WT
                  }

                  code

                  }
                                pp program name
                ENDRUN              recall bit if desired

CALL            VFD       24/4LPGMP,36/PARAM

                                        address in CM for
                                        pp program to
                                        find parameters or
                                        set complete bit
PARAM           BSSZ      1         PARAM in this case is a
                                    fake FET in which the PP pgm
                                    can set the complete bit to
                                    bring the CP out of recall.

                                    It may be any group of words,
                                    such as a FET for I/O

                END       TEST
```

test
XJ

2/72

SETTING UP THE PP PROGRAM

```
                                          program name
                                          ↙
                  IDENT        PGM,C.PPFWA PPFW
pp code    →      PERIPH                 ↖         address to start
get system  →     SST                              ↙
symbols
                  ORG          C.PPFWA PPFW
                                                  NAME    SUBR
           →      PPENTRY      D.PPIRB,D.TO
set up            {                  ↖       ↖
D.CPAD,D.RA,D.FL  {                             used by R.RAFL
do R.RAFL         {
                   code              PP Input Register Buffer
                  {                  in the PP
                  {

                  RJM          R.RAFL  PRL }
                  {                        }  RAFL's and TAFL's
                  {                        }  as needed to access
                  {                        }  CM field
                  RJM          R.TAFL  PBL }
                  {
                  {
                  {
                  {
                  {
                                  DPPM
                  LDN          M.DPP   ←————— drop me code
                                      FTN
                  RJM          R.MTR  ←————— go do it
                                      PPR
                  LJM          R.IDLE ←————— go to idle loop
                                              {a TAFL is also done
      {BUF        BSS          10             in case the
      {                                       programmer forgot}
      {
buffers           END
needed by
program
```

Note Upon entry to the PP program, the Field Access Flag {FAF} is clear. The program must do a PPENTRY or RJM R.RAFL

CONTROL CARDS

PP CODING CLASS PROJECT

ALSO V 3.2

```
**********  MASTER DECK  ---  PP CODING CLASS            SCOPE 3.3
**********  THIS IS THE CARD DECK TO USE FOR THE COMPASS ASSEMBLY RUN
**********  USE YOUR INITIALS AS 2ND  TWO DICITS OF SEQ NO (CC 12 AND 13)
$SEQUENCE.L
$CHARGE.
YOURJOB.CM60000.T10.P6.
COMPASS(S=SCPTEXT)
0000000000000000000000
**********  PUT YOUR PP AND CP PROGRAMS HERE  ---  TAKE THIS CARD OUT
0000000000000000000000
0000000000000000000000
**********  THIS IS THE CARD DECK TO USE FOR THE EDITLIB RUN
YOURJOB.T100.
COMPASS(B=PPTEST.S=SCPTEXT)
REWIND(PPTEST)
EDITLIB.
COMPASS.
LGO.
DMP(100.105)
DMP.
DMP(100.400)
EDITLIB(RESTORE)
EXIT.
DMP(100.105)
DMP(100.400)
EDITLIB(RESTORE)
0000000000000000000000
**********  PUT YOUR PP PROGRAM HERE  ---  TAKE THIS CARD OUT
0000000000000000000000
READY(SYSTEM)
ADD(*,PPTEST)
COMPLETE.
0000000000000000000000
**********  PUT YOUR CP CALLING PROGRAM HERE  ---  TAKE THIS CARD OUT
0000000000000000000000
```

```
********** MASTER DECK  ---  PP CODING   SCOPE 3.4 CLASS
**********  THIS IS THE CARD DECK TO USE FOR THE COMPASS ASSEMBLY RUN
YOURJOB.
COMPASS(S=SCPTEXT)
0000000000000000000000
PUT YOUR PP AND CP PROGRAMS HERE  ---  TAKE THIS CARD OUT
0000000000000000000000
0000000000000000000000
```

```
**********  THIS IS THE CARD DECK TO USE FOR THE EDITLIB RUN
YOURJOB.T100.
COMPASS(B=PPTEST,S=SCPTEXT)
REWIND(PPTEST)
EDITLIB(SYSTEM)
COMPASS(S=CPCTEXT)
LGO.
DMP.
DMP(100,105)
DMP(100,400)
EDITLIB(SYSTEM,RESTORE)
EXIT.
DMP(100.105)
DMP(100,400)
EDITLIB(SYSTEM.RESTORE)
0000000000000000000000
********** PUT YOUR PP PROGRAM HERE  ----   TAKE THIS CARD OUT
0000000000000000000000
READY(SYSTEM.OLD)
ADD(*,PPTEST)
COMPLETE.
ENDRUN.
0000000000000000000000
********** PUT YOUR CP CALLING PROGRAM HERE   ---   TAKE THIS CARD OUT
0000000000000000000000
```

CD
11/72

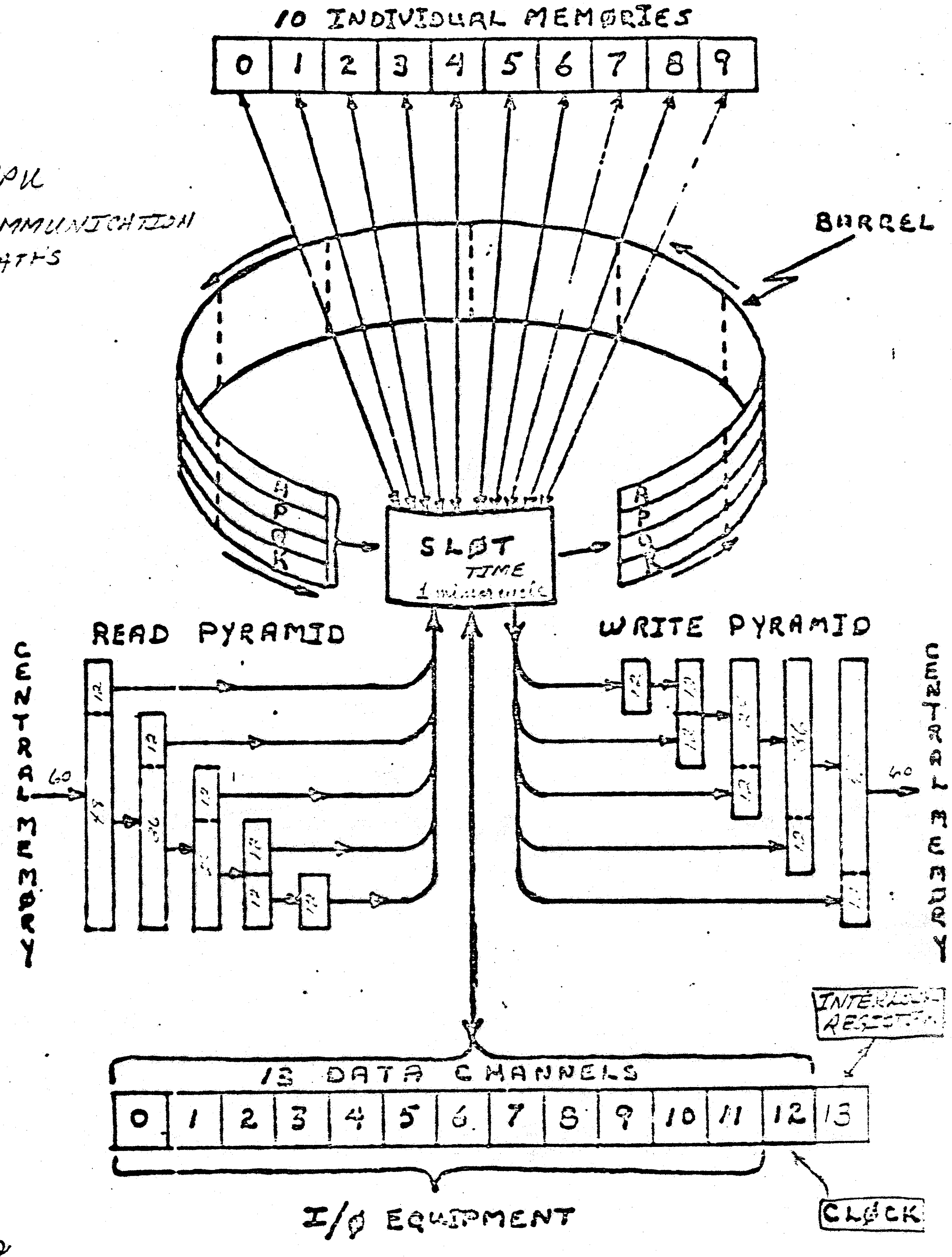
SECTION THREE

HARDWARE

SECTION THREE - HARDWARE

CONTENTS

PPU COMMUNICATION PATHS



dan cone

REGISTERS

A

18 BITS

- MAIN ARITHMETIC REGISTER (ALSO ADDER)
  NO SIGN EXTENSION
  UPPER BITS ARE ZERO WHEN PP 12-BIT OR 6-BIT QUANTITIES ARE PUT THERE
- ALSO USED IN SOME I/O
  AND SHIFT, LOGICAL, INSTRUCTIONS
- HOLDS 18-BIT CM ADDRESSES (ABSOLUTE) FOR CENTRAL I/O

P

12 BITS

- HOLDS ADDRESS OF CURRENT INSTRUCTION
- ALSO USED IN I/O
  (TO HOLD DATA ADDRESS OF WHERE IN THE PP
  DATA IS GOING TO OR FROM)

Q

12 BITS - HOLDING REGISTER

- HOLDS ADDRESS
  - DELTA OF A 12-BIT INSTRUCTION
  - ADDRESS FROM DELTA FOR INDIRECT ADDRESSING
- HOLDS DELTA OF OTHER INSTRUCTIONS
- ALSO AN ADDER (ADDS +1 or -1 TO ITSELF)

K

9 BITS

- OP CODE - UPPER 6 BITS
- TRIP COUNT - LOWER 3 BITS
  (WHICH TRIP THE INSTRUCTION IS ON, AROUND THE BARREL)

OTHER REGISTERS:
- I/O CHANNEL REGISTERS
- READ/WRITE PYRAMID REGISTERS

* ONLY THE A REGISTER IS PROGRAMMABLE *

ADDER

- BOTH OPERANDS ARE COMPLEMENTED GOING INTO THE ADDER
- THE ANSWER IS RECOMPLEMENTED COMING OUT

ie ADD 3+4

```
 original      complemented                            recomplemented
 number        numbers                                   answer

  0003 ------->  7774
 +0004 ------->  +7773
                 -----
                  7767
                     1
                 -----
                  7770 ------------------------------->  0007
```

ie SUBTRACT 3-4

```
  0003 ------->  7774                7774
 -0004 ------->  -7773  ---------->  +0004
 -----           -----        ↗      -----
                             /        0000
          To subtract,      /            1
          complement       /          -----
          and add         /           0001 ----------->  7776
```

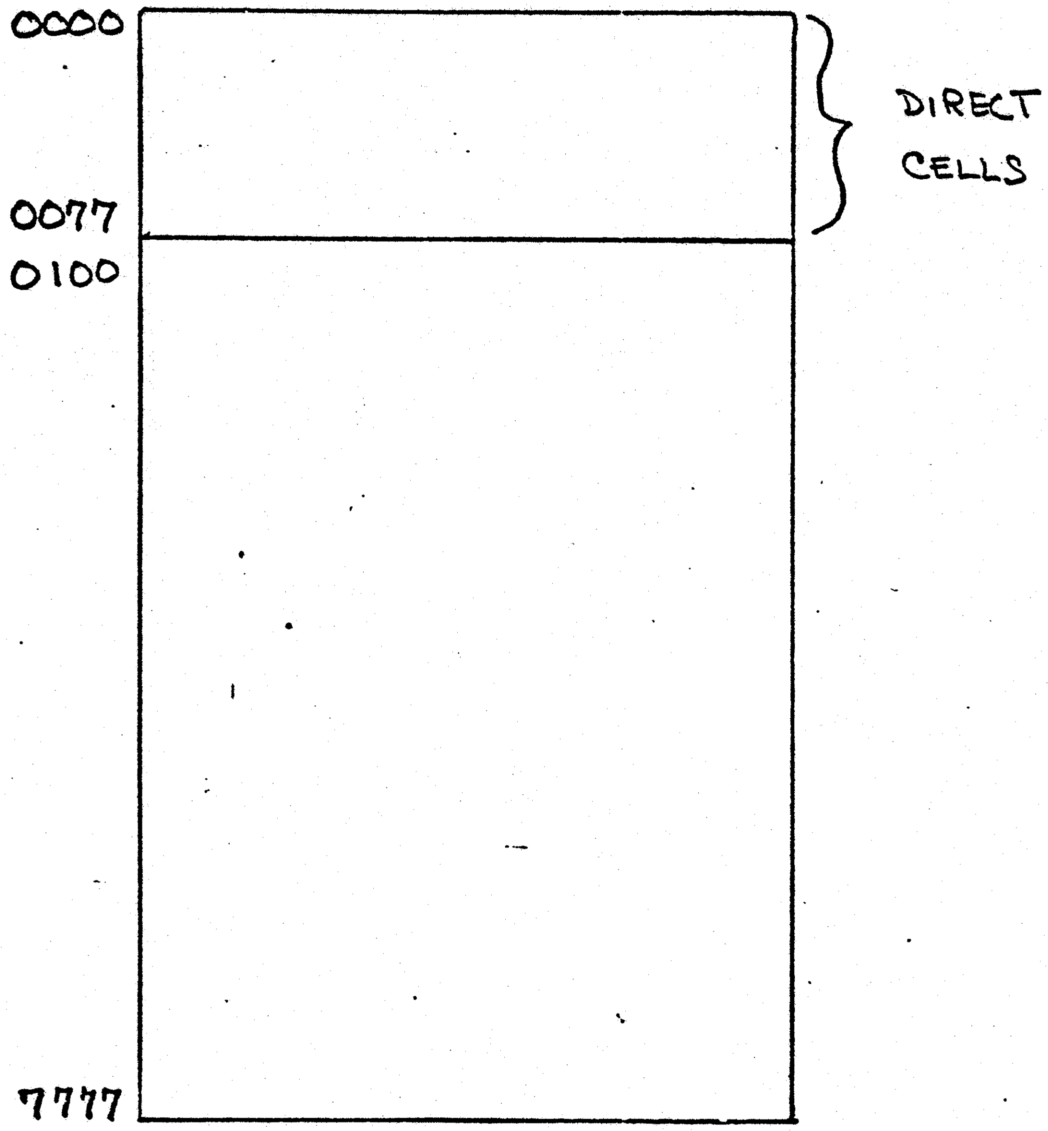
ie ADD +3 and -3

```
  0003 ------->  7774
 +7774 ------->  +0003
 -----           -----
                  7777 ------------------------------->  0000
                                                           ↖
                                          Note a +0 is       \
                                          returned by the adder
```

Customer Engineering
PP Training Manual - p 5-32
CDI 020267 March 67

c/ 2/1/12

PP MEMORY



CYBER 70 HARDWARE FEATURES {What makes them different from 6TTT's}

Contents

3-5

{THE FOLLOWING SET OF
PAGES IS EXTRACTED
FROM THE SCOPE 3.4
IMS, M655

CYBER 70 HARDWARE FEATURES

INTRODUCTION

The CDC CYBER 70 series computers have several new hardware features which were not available or optional on the 6000 series machines. Among these new features are the Central Exchange Jump/Monitor Exchange Jump {CEJ/MEJ} capability, the Compare/Move Unit {CMU}, the Distributive Data Path {DDP}, and the Interlock Register. All these hardware features are supported by the SCOPE 3.4 Operating System, and a thorough knowledge of their concepts is essential for full utilization of the CYBER series computers and their associated software.

This chapter is intended as an introduction to these hardware features and the software instructions that use them.

CENTRAL EXCHANGE JUMP/MONITOR EXCHANGE JUMP {CEJ/MEJ}

BACKGROUND REVIEW

In most 6000 systems operated under SCOPE 3.3 or older, Monitor runs in a dedicated PP{PP0}. A job accesses or relinquishes the Central Processor each time Monitor issues an exchange jump instruction {EXN}. This is done when a job has used the Central Processor for the maximum interval allowed by SCOPE. Each time the EXN instruction is executed, any job using the Central Processor is interrupted and the job with the next highest priority has access to the Central Processor. When an active job is interrupted, all pertinent information about the job {register contents, RA, FL, etc.} are saved in a 16 word exchange package where they can be picked up later for execution again by another EXN instruction issued by Monitor.

CEJ/MEJ

In the CYBER series computer, however, a new type of exchange jump is available with the CEJ/MEJ hardware and the SCOPE 3.4 Operating system. In the CYBER series machine, the Central Processor has, in the Central Memory control section, a Monitor mode flag bit. The flag is cleared by Deadstart. Thereafter, it can be set or cleared only by the Monitor exchange jump {MXN} on the central exchange jump {XJ} instructions supplied by SCOPE 3.4. There is no instruction with which to test the status of this flag directly or independently. We can now distinguish two types of Central Processor operations. The CPU executes in either Monitor Mode or User Mode depending on whether this Monitor Mode flag is set or clear. All user programs, as well as many system programs, run in User Mode. The only programs that do run in Monitor Mode are CP Monitor, SPM, and CPCIO. Programs running in User Mode can be interrupted at any time, either because the CPU is needed by a Monitor Mode program or upon the expiration of the time slice. In Monitor Mode, however, the CPU is not interruptable and is permitted to execute until a task has been completed.

To utilize fully the CEJ/MEJ hardware and to provide Monitor with a processor more powerful than a PPU, the Monitor in SCOPE 3.4 is divided into two separate parts, a CP Monitor and a PP Monitor, with each performing different functions. CP Monitor, which resides in Central Memory Resident, controls CPU Monitor Mode execution and CPU scheduling. PP Monitor, which is in general control of the system, operates in PP0. {For details, please refer to the chapter on Monitor.}

When a User Mode program has used up its time slice, or when a PP {e.g., PP Monitor or other PP routine} needs the CP Monitor to perform a certain task it initiates an MXN exchange jump instruction. This will activate the CP Monitor immediately if the Central Processor is running in User Mode. The job that was running is forced to relinquish the CPU to CP Monitor. At the same time the Monitor Mode flag is set putting the CPU in Monitor Mode execution. If, however, the CPU is already in Monitor Mode when MXN is initiated by a PP, it will be ignored and treated as a PASS instruction. The CPU is allowed to execute without interruption until a task is completed.

The Central Exchange Jump instruction {XJ} is used in conjunction with MXN. As mentioned before, the CPU is not interruptable while in Monitor Mode. Hence, the Monitor Mode program must exit itself. When a task is completed, the Monitor Mode program initiates an XJ exchange jump. This will release the Central Processor to a User Mode job and at the same time clear the Monitor Mode flag returning the computer to normal program mode execution. When a User Mode CP program needs the CP Monitor, it too can initiate an XJ. This will activate the CP Monitor immediately {as in the case of the MXN for a PP program} and put the computer in Monitor Mode. Hence, the mode of execution of the Central Processor changes every time upon the completion of the XJ exchange jump.

The CEJ/MEJ hardware operation is enabled or disabled by a control switch on the deadstart panel. If it is enabled, the CEJ/MEJ feature will operate as above. However, if it is disabled or in an installation without the MXN/XJ instruction set, the EXN instruction is used. This is a PP initiated exchange jump which occurs independently of the mode of the CPU and has no effect on the Monitor Mode. PP Monitor is the only program that may perform an EXN. In fact, it simulates the MXN for all PPs in the system and also simulates XJ for the Central Processor as SCOPE 3.3.

SCOPE 3.4 requires either the combination of MXN/XJ or EXN to run.

The different exchange jumps are summarized below:

NAME	INSTRUCTION CODE
PPU Regular Exchange Jump - EXN	260d
PPU Monitor Exchange Jump - MXN	261d
CPU Central Exchange Jump - XJ	013jk

Table 1 summarizes the operational differences between the Normal exchange jump instruction {260} and the Monitor and Central Exchange jumps {261 and 013}.

EXCHANGE INSTRUCTION DIFFERENCES

INSTRUCTION	CONDITIONAL/ UNCONDITIONAL	OPERATIONAL DIFFERENCES	
		Effect on Monitor Flag Bit	Location of Starting Address of Exchange
260 {Normal Peripheral Processor Exchange Jump}	Unconditional	No effect on Flag	Peripheral Processor A Register
261 {Peripheral Processor Monitor Exchange Jump}	Conditional {occurs only if Monitor Flag bit is clear; passes if flag is set}	Sets Flag	Peripheral Processor A Register
013 {Central Exchange Jump} with Monitor Flag bit clear	Unconditional	Sets Flag	Central Processor Monitor Address Register
013 {Central Exchange Jump} with Monitor Flag bit set	Unconditional	Clears Flag	Address formed by K+{Bj}

TABLE 1

Their instruction formats are as follows:

OPERATION	VARIABLE	DESCRIPTION	SIZE	OCTAL CODE
EXN	d	Exchange jump tp CPU d	12 bits	260d
MXN	d	Monitor exchange jump CPU d to {A}	12 bits	261d
MAN	d	Monitor exchange jump CPU d to {MA}	12 bits	262d
XJ		Exchange jump to MA if in Program Mode	30 bits	0130G 00000
XJ	Bj	Exchange jump to {Bj}; flag set	30 bits	013j0 00000
XJ	K	Exchange jump to K; flag set	30 bits	0130K
XJ	BJ+K	Exchange jump to {BJ+K; flag set	30 bits	013jK

In 6500 or 6700 systems {or CYBER 70/Model 72-2Z, 73-2Z, or 74-2Z} with dual Central Processors, d can be 0 or 1 and specifies which CPU the exchange jump will interrupt. In single processor systems, this value is not interpreted.

Please also note that the assembler forces upper before and after assembling an XJ instruction.

OTHER EXCHANGE JUMP

Besides the MXN/XJ and EXN exchange jump, two other exchange jump instructions are available.

1. MAN

   The MAN exchange jump {octal code 262} is a PPU instruction that executes just like the MXN. However, the exchange package address is taken from the 18 bit Monitor Address {MA} Register in the CPU rather than the A register of a PP. Which instruction is set to use {MXN/XJ or MAN/XJ} is determined by an installation parameter {IP.XJ}.

2. Program Stop/Error Exit Operation

   The Program Stop instruction PS could execute an exchange jump on the CEJ/MEJ panel switch.

   The DISABLE position disables the Central exchange jump or the Monitor exchange jump. In this case, PS halts the Central Processor unit at the current step in the program. An exchange jump is necessary to restart the Central Processor unit. The

ENABLE position enables the jump capabilities. In this case, PS causes an exchange jump to monitor address {MA} in the exchange package.

The contents of the location field become a sub-subtitle on the assembler listing. The assembler forces upper before and after assembling a PS instruction.

Instruction Format:

OPERATION	VARIABLE	DESCRIPTION	SIZE	OCTAL CODE
PS		Program stop or exchange jump to {MA}	30 bits	00000 00000
PS	K	Program stop or exchange jump to {MA}	30 bits	0000K

Its operation is summarized as follows {CEJ/MEJ enabled}:

Monitor Flag Clear	Store P+1 at RA Clear P Exchange Jump to {MA} Set Monitor Flag
Monitor Flag Set	Store P+1 at RA Clear P Stop CPU Monitor Flag Remains Set

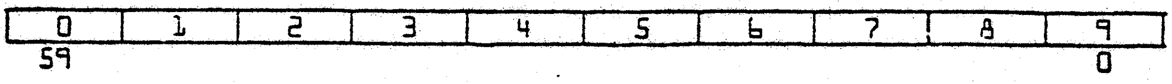
Program errors can also cause an Exchange Jump to happen. Hardware action during an attempted execution of an illegal instruction will effect the following {CEJ/MEJ switch enabled}:

Monitor Flag Clear	Store P+1 at RA Clear P Exchange Jump to {MA} Set Monitor Flag
Monitor Flag Set	Store P+1 at RA Clear P Stop the CPU

COMPARE MOVE UNIT {CMU}

The Compare Move Unit is a standard CPU hardware component of the CYBER 70 series Model 72 and 73 and optional on the Model 76 computer system. It provides the capability to move and compare data fields in storage without having to use the registers.

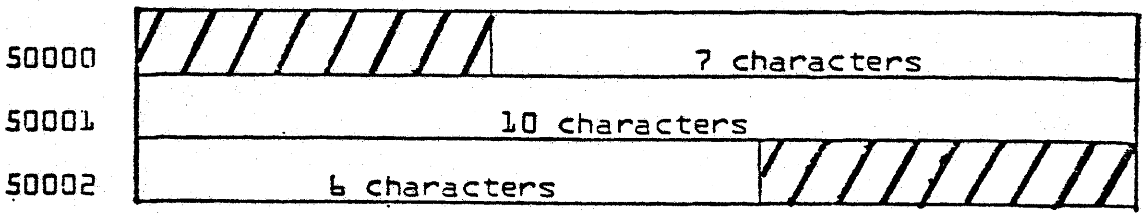
There are ten 6-bit character positions in each 60-bit word. These positions are numbered 0 through 9 from left to right respectively. The 4-bit character addresses of these positions are {in binary} 0000, 0001, ..., 1000, and 1001. Character addresses 1010 through 1111 are illegal and cause the instructions to give an address out of range condition.



STORAGE WORD

Data fields may span word boundaries and may start or end at any position in a 60-bit word.

Example:



The field above starts at character 3 in word 50000 and ends with character 5 in word 50002. The field has a length of 23_{10} characters. One limitation for using the CMU is that the data field must not be in an operating register or in ECS/LCM.

COMPASS 3.0 provides symbolic forms of four CMU instructions. They are:

1. Indirect Move - IM
2. Direct Move - DM
3. Compare Collated - CC
4. Compare Uncollated - CU

Of the above, only the Indirect Move {IM} instruction has the same type of syntax and semantics as other CPU instructions. The others are treated as pseudo instructions by COMPASS.

INDIRECT MOVE {IM}

This is a 30-bit instruction that moves the content of a data field to another data field according to a descriptor word. Maximum length of the data field that could be moved by this instruction is 8191_{10} characters. The descriptor word contains the length and addresses of the data fields. COMPASS forces the instruction to the upper left of a word because it is executed as a pass by the hardware if it is not the first instruction of a word. The next instruction is also forced upper in the next word, because the lower half of a word containing an indirect move is not executed.

Format:

OPERATION	VARIABLE	DESCRIPTION	SIZE	OCTAL CODE
IM	Bj	Move per descriptor at Bj	30 bits	464j000000
IM	K	Move per descriptor at K	30 bits	4640K
IM	Bj±K	Move per descriptor at Bj±K	30 bits	464jK

Execution: The descriptor word is fetched from storage location {Bj}±K. If the data field length is zero, the instruction is executed as a pass but the execution time is longer. Otherwise, the content of the source field is moved to the destination field. If the two fields overlap, the results are undefined. The X0 register is used for intermediate storage during execution of the instruction and is cleared upon completion of the instruction.

A pseudo instruction MD is used to generate a descriptor word for use by the indirect move instruction. The MD instruction has the following format:

LOCATION	OPERATION	VARIABLE
locsym	MD	L, K_S, C_S, K_D, C_D

L is the absolute address expression; its value, in the range $0 \leq L \leq 8191$, is the data field length in characters. The upper 9 bits are placed in bits 56-48 of the descriptor word while the lower 4 bits are placed in bits 29-26.

K_S is any expression, the first word address of the source field.

C_S is the absolute expression, the starting character position of the source field within the word at location K_S.

K_D is any expression, the first word address of the destination field.

C_D is the absolute expression, the starting character position of the destination field within the word at location K_D.

Indirect Move Descriptor Word Format:

59	56	47	29	25	21	17 0
0	L_u	K_S	L_L	C_S	C_D	K_D

Where:

L_u: Upper 9 bits of value of L.

L_L: Lower 4 bits of value of L.

DIRECT MOVE (DM)

The direct move pseudo instruction generates a CMU instruction that moves a data field in storage to another location in storage. This instruction differs from the indirect move in several ways. It is a 60-bit instruction that cannot be split between words and the descriptor word is part of the instruction. Furthermore, the length of the data field it can move is limited to a maximum of 127_{10} characters.

Instruction Format:

LOCATION	OPERATION	VARIABLE
locsym	DM	L, K_S, C_S, K_D, C_D

59	50	47	29	25	21	17	0
465	L_u	K_s	L_L	C_s	C_D	K_D	

L is the absolute address expression; its value, in the range $0 \leq L \leq 127$, is the data field length in characters.

L_u is the upper 3 bits of the value of L.

L_L, K_S, C_S, K_D, C_D: Same as in the MD instruction.

Execution: Same as IM, except that the descriptor is in the instruction word itself.

COMPARE COLLATED (CC)

The compare collated instruction compares the contents of two data fields, one character at a time, from left to right, until a pair of corresponding characters are found to have unequal collating values, or until the data fields are exhausted. It is a 60-bit instruction that occupies one full word (it cannot be split between two words) and contains its own data field descriptor.

It uses register A0 to contain the first word address of a table in storage that contains the collating values to be used in comparing characters. The result of the comparison is placed in register X0.

Format:

LOCATION	OPERATION	VARIABLE
locasym	CC	L, K_A, C_A, K_B, C_B

59	50	47	29	25	21	17 0
466	L_U	K_A	L_L	C_A	C_B	K_B

L, L_U, L_L are same as in the DM instruction.

K_A is any expression, the first word address of the first data field.

C_A is the absolute expression, the starting character position of the first data field within the word at location K_A.

K_B is any expression, the first word address of the second data field.

C_B is the absolute expression, the starting character position of the second data field within the word at location K_B.

Execution: The first word address of the collating table is obtained from register A0. The contents of the data fields are compared from left to right, one character at a time from each field, until two unequal characters are found. The collating value of each character is obtained from the collating table. If these values are equal, the compare continues until another character pair is unequal or until all characters have been compared. If the collating values are unequal, the two data fields are unequal and the field with a larger collating value is the greater of the two fields. The collating values are treated as 6-bit unsigned integers.

Note that two unequal characters could have the same collating value and would compare equal. Upon completion, register X0 contains a 60-bit signed integer as follows:

$X0 = L - N > 0$ if field A > field B

$X0 = +0$ if field A = field B

$X0 = N - L < 0$ if field A < field B.

where N is the number of pairs of characters that compared equal.

If L=0, then X0=+0.

The format of the collating table is as follows:

	59	53	47	41	35	29	23	17	11 0
{A0}	00	01	02	03	04	05	06	07	
{A0} +1	10	11	12	13	14	15	16	17	
⋮									
{A0} +7	70	71	72	73	74	75	76	77	

COMPARE UNCOLLATED {CU}

The compare uncollated instruction compares the contents of two data fields, one character at a time, from left to right, until a pair of corresponding characters are found to have unequal values, or until the data fields are exhausted. It is a 60-bit instruction that occupies one full word {it cannot be split between two words} and contains its own data field descriptor. The result of the comparison is placed in register X0.

Format:

LOCATION	OPERATION	VARIABLE
locasym	CU	L, K_A, C_A, K_B, C_B

59	50	47	29	25	21	17 0
467	L_U	K_A	L_L	C_A	C_B	K_B

Execution: Same as the CC instruction except that A0 and the collating table are not used. Instead, the characters are compared directly with each character regarded as a 6-bit unsigned binary integer. Register X0 is set in the same manner as by the CC instruction.

DISTRIBUTIVE DATA PATH - DDP

PHILOSOPHY

The Distributive Data Path is a new hardware feature designed to increase the performance and throughput of systems equipped with Extended Core Storage (ECS). If DDP is not available, data transfers between ECS and peripheral equipment must pass through central memory using a system double buffer. (For a description of ECS I/O buffering please see the chapter on ECS Extensions.) The Distributive Data Path provides a data path between a PPU and ECS, allowing direct PPU to/from ECS data transfers. The DDP utilizes one access of an ECS controller to communicate with ECS. A PPU in turn communicates with the DDP via I/O data channels. Data is transferred across this channel in 12-bit bytes at a maximum rate of up to one million bytes per second.

The DDP is expandable from one to a maximum of four identical PPU data channel interfaces. Each of these PPU interfaces, called ports, operated independently while sharing a common ECS interface. The first interface is part of the DDP. The second, third, and fourth interfaces are the optional DDPRI's. These interfaces each contain a buffer which is used to assemble 12-bit bytes into an ECS record or to disassemble an ECS record. When 480 bits of the buffer are available, a request for ECS transfer is made. An equal-priority scanner monitors the four Port-ECS-Request signals and connects a requesting port to the ECS Controller interface for an ECS transfer. At the completion of one ECS Record transfer, the scanner moves on to check for a request from the next port.

It takes at least 40 microseconds to transfer a 480-bit ECS record between a PPU and a DDP port. The DDP port has buffering to allow the data channel to maintain its one mega-byte per second transfer rate while data is being transferred between the port and ECS, if no more than two devices are actively accessing ECS. For example, two DDP ports can maintain a one mega-byte rate if nothing else is accessing ECS; one DDP port can maintain a one mega-byte rate if no more than one other ECS controller access is busy, such as, if the CPU is accessing ECS.

Restrictions on the DDP port are that it must be either the first or the second device out of a data channel to maintain a one MHZ transfer rate and it must be the last device on a data channel.

DDP PROGRAMMING

FUNCTION CODES

The DDP is controlled via functions from the PPU I/O channel. Function codes are sent to the DDP PPU port with the upper three bits containing the equipment select code {5}, and the remaining bits designating the function to be performed. Functions are sent out on an inactive channel by the PPU and the DDP responds to valid functions by disconnecting the data channel. The function codes used to control a DDP port are:

5001 - ECS Read
5002 - ECS Write
5004 - Status
5010 - Clear Port

All other function codes are either illegal or ignored by the DDP. The DDP will respond to all function codes with the correct equipment select code and the remainder of the upper eight bits equal to zero. The remaining four bits of the function code must have only one bit set to select the required function. More than one bit set is illegal and the results of such a condition are undefined.

5001 - ECS READ

This function causes the DDP port to read data from ECS and to present this data to the I/O channel for input to the PPU. The DDP responds to this function by disconnecting the data channel. When the channel is activated by the PPU, the DDP requires an output of two 12-bit bytes from the channel. These bytes are loaded into the 24-bit address register, with the first byte going into the upper twelve bits of the register and the second byte into the lower twelve bits. The address register now designates the ECS address of the first 60-bit word to be presented to the PPU.

The ECS read function has three selectable modes.

A. The first mode is referred to as Block Read Mode. In this mode a new request to ECS is made whenever a sufficient amount of buffer register space is available for a new ECS Record. The DDP increments the address register once for each 60-bit word it receives such that each subsequent request is made at the next higher ECS record address. The DDP will continue to make these requests until the data channel is made inactive.

B. The second mode of operation is the Read 1 Mode. In the read 1 mode, a second ECS request is never made. The purpose of this mode is to eliminate the wasteful second request that would be made to ECS under a Block Read when data from only one ECS record is needed. This mode is terminated when the data channel goes inactive.

C. The third ECS Read mode does not cause data to be read from ECS. This mode is called Function Flag Register mode. When this mode is selected, the DDP sends the contents of the address register to ECS and terminates the ECS read condition within the DDP. The PPU must disconnect the data channel. This is the manner in which a flag register operation is performed.

The selection of these modes of ECS Read is determined by the two most significant bits (2^{23} and 2^{22}) of the ECS address given to the DDP at the start of the ECS Read. Any time bit 2^{23} is a "1", Function Flag Register mode is selected. If bit 2^{23} is a "0" and 2^{22} is a "1", then Read 1 mode is selected. If both 2^{23} and 2^{22} are "0", a Block Read is performed.

Graphically, this is:

2^{23}	2^{22}	Mode
0	0	Block Read
0	1	Read 1
1	0	Function Flag Register
1	1	Function Flag Register

As stated in the descriptions of the three modes of an ECS Read, this function is terminated by the data channel going inactive. The channel can be disconnected by either the PPU or the DDP. In the Block Read and Read 1 modes, two error conditions exist that will cause the DDP to disconnect the data channel. They are:

ECS Abort
ECS Parity Error

If either of these two conditions is received from the ECS Controller in response to an ECS request for data, then the DDP will disconnect the data channel after the last byte of the previous ECS Record has been transferred to the PPU and when the data channel is in the Empty state. The disconnect signal is sent out by the DDP on an Empty channel rather than a Full and data. This is done to give the PPU the ability to determine whether the DDP is going to send a disconnect or not. If the channel is Full, the PPU can send a disconnect without risking a hang-up condition. If the channel is Empty, a PPU-generated disconnect is illegal on the basis that the DDP may disconnect.

When the data channel is disconnected by the DDP, the status word must be read to determine the reason for the disconnect. In the case of Parity Error, the PPU may issue a Read 1 function in Maintenance Mode to input the data contained in the buffer register. The only way to read more data beyond that is to issue a new "ECS Read" function. Note that any ECS Read function must have an address sent to the DDP before data can be input by the PPU.

The issuance of a Read 1 in Maintenance Mode causes the data in the buffers to be presented to the data channel for input by the PPU without sending a request to ECS. Maintenance Mode is selected by setting bit 2^{21} of the address to a logical "1".

The upper three bits of the Address Register provide these variations of the "ECS Read" function.

2^{23}	2^{22}	2^{21}	Function
0	0	0	ECS Block Read
0	0	1	Block Read in Maintenance Mode
0	1	0	ECS Read 1
0	1	1	Read 1 in Maintenance Mode
1	0	0	Function Flag Register - Ready Select
1	0	1	Function Flag Register - Selective Set
1	1	0	Function Flag Register - Status
1	1	1	Function Flag Register - Selective Clear

These various modes are selected or deselected according to the most recent address sent to the DDP.

It can be seen from the above chart that when bit 2^{23} is set, Maintenance Mode is not selected. 2^{23} dictates a Flag Register operation and Maintenance Mode does not exist for a Flag Register operation.

An instruction sequence to do an ECS Read is:

```
     FCN     5001

     ACN

     OAM     XXXX, Where {A} = 2

     IAM     XXXX, Where {A} = 12-bit byte count

     NJN     Error:  Channel disconnected via DDP:  Read Status
             {See para. 3.2.1.5.5}

B    IJM     A }  Wait for change from Active and

     EJM     B }  Empty State

     DCN

A    XXX
```

The ECS Read condition within the DDP may be cleared out by a PPU disconnect, a DDF disconnect, a power-on Master Clear, a deadstart Master Clear, or by a functioned port clear.

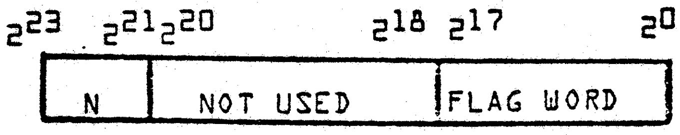
FUNCTION FLAG REGISTER

This function is performed any time bit 2^{23} of the Address Register is set when the address for an ECS Read is loaded into the DDP. The Flag Register in the ECS Controller cannot be read directly but may be interrogated and/or written into.

Interrogation is accomplished by selecting and reading status from the DDP after the Function Flag Register operation has been performed. The status word shows whether an Abort or an Accept has been received from the ECS Controller in response to the Flag Register word.

Four Flag Register operations may be performed with the three most significant bits of the Address Register {N} determining which operation is to be performed.

The Flag Word Format is:



The possible operations are:

N=4 {Ready Select}

A bit by bit comparison is made in the ECS Controller of the lower eighteen bits of the Flag Register in the ECS Controller and the Flag Word received from the DDP. If all bits set in the Flag Word are clear in the Flag Register, then the ECS Controller responds with an Accept to the DDP and enters the set Flag Word bits into the Flag Register. The clear bits in the Flag Word have no effect on the Flag Register.

If any of the bits set in the Flag Word are set in the Flag Register, then the ECS Controller responds with an Abort and does not modify the Flag Register.

Examples, using only three bits, are:

Flag Register = 010

Flag Word = 101

Result: Accept and Flag Register = 111

Flag Register = 010

Flag Word = 011

Result: Abort and Flag Register = 010

N=5 {Selective Set}

Selectively sets bits in the Flag Register from bits set in the Flag Word. The only response is an Accept.

N=6 {Status}

Same as Ready Select, except that the contents of the Flag Register are not changed. {NOTE: This is a Flag Register Status and has nothing to do with DDP Status, function code 5004.}

N=7 {Selective Clear}

Selectively clears bits in the Flag Register from bits set in the Flag Word. The only response is an Accept.

5002 - ECS Write

This function causes the DDP port to assemble bytes from the data channel and to write data in ECS. The DDP responds to an ECS Write function by disconnecting the data channel. When the channel is activated by the PPU, the DDP will begin accepting data from the PPU. The first two 12-bit bytes received from the channel must be the address at which the first ECS Record is to be written. These bytes are loaded into the 24-bit Address Register. The bytes that come after the second byte are regarded as data and sent to ECS.

The first byte received by the DDP is put into the upper twelve bits of the Address Register. The second byte is put into the lower half. The Address Register now designates the address of the first 60-bit word to be written into ECS. This address is presented to the ECS Controller along with a request signal after the buffer in the DDP is filled by the PPU or after a disconnect is received from the PPU. The Address Register is incremented as the buffer empties into ECS. Unless an error condition is encountered, data will continue to be transmitted in this fashion. A disconnect from the PPU will cause accumulated data to be written into ECS, and the ECS Write condition within the DDP to clear out. If the PPU disconnects the DDP with less than an integer multiple of 60-bit words assembled in the DDP buffer registers, then the partial 60-bit word will be written into ECS with zeros in the missing byte{s}.

A program sequence such as the following will produce a partial ECS Write with zero fill.

FNC ECS Write

ACN

LDC 20_{10} .

OAM

Only one error condition is possible on ECS Write. If the ECS Controller returns an Abort signal to the DDP, the DDP will disconnect the I/O channel. This disconnect will be sent to the I/O channel in the place of an Empty response to a Full signal from the data channel. This will eliminate the possibility of hanging the channel when the PPU performs a disconnect. However, if an Abort comes after the PPU disconnects the channel, the only way to detect it is to do a status check after the disconnect, waiting for the Write status to drop.

An instruction sequence to do an ECS Write is:

```
FNC     5002    ECS Write

OAM     XXXX    Where {A}=2 + the number of 12-bit bytes of
                data to be sent {first 2 words are address}

NJN     Error:  Channel disconnected via DDP:  Read Status

DCN     Keeping in mind that the channel must be Empty.

FCN     Read Status:  Check for Abort or Accept:  continue to
        read status until one or the other is detected {may
        take as long as 50 microseconds}.
```

5004 - Select Status

This function makes the status of a port available for PPU input after the channel is activated by the PPU. The DDP responds to this function code by disconnecting the data channel. The PPU then activates the channel and inputs a 12-bit word. Status may be repeated at this point simply by doing another input. When it is desired terminate the reading of status, the PPU must issue a disconnect to the data channel.

Status bits are assigned to indicate the following

2^0	ECS Abort
2^1	ECS Accept
2^2	ECS Parity Error
2^3	ECS Write

2^0 - ECS Abort

This status bit indicates that an Abort signal has been received from ECS.

2^1 - ECS Accept

This status bit indicates that an Accept signal has been received from ECS.

2^2 - ECS Parity Error

This status bit indicates that a Parity Error signal has been received from ECS.

2^3 - ECS Write

This status bit indicates that the DDP port is busy with a write to ECS. When the write terminates, this status bit will clear out.

Besides being cleared by a port or master clear, the status bits {Abort, Accept, and Parity Error} are cleared out either by a new request to ECS or by reading status. Status must be read to clear out a DDP generated disconnect due to an ECS Abort or an ECS Parity Error.

5010 - Port Clear

This function is a programmable master clear for the data buffers and control logic within the DDP associated with the port to which this function is issued. This function, as does Deadstart Master Clear, clears only that DDP port to which the clear is issued.

INTERLOCK REGISTER - ILR

INTRODUCTION

The Interlock Register {ILR} is another new hardware feature that is available on all CYBER 70 machines. It is a 64-bit register which could be expended to 128 bits. It can be accessed by the PPU's through two data paths. An Interlock {channel 15_8} will be added to each set of 10 PPU's to enable up to 20 PPU's to access the ILR. Initial software utilization of the ILR will include primarily I/O channels and pseudo-channels interlocking.

Interlock Register

*					**					
Word 10	Word 9	Word 8	Word 7	Word 6	Word 5	Word 4	Word 3	Word 2	Word 1	Word 0

127 119 107 95 83 71 63 59 47 35 23 11 0

* - Word 10 is 8 bits.
** - Word 5 is 4 bits in the 64-bit Interlock Register.

OPERATIONS

Eight operations can be performed on the Interlock Register from the PPU.

1. Set

   Sets a bit specified by the octal translations $0 \geq 77_8$ or $0 \geq 177_8$.

2. Clear

Clears a bit specified by the octal translations $0 \geqslant 77_8$ or $0 \geqslant 177_8$.

3. Test

Checks a bit specified by the octal translations $0 \geqslant 77_8$ or $0 \geqslant 177_8$ and sends the PPU a status of "1" or "0", depending on if the bit is set or clear. The status bit will be located in the bit zero position in the 12-bit word. The other 11 bits in the status word will be zero.

4. Read

One of the 6 or 11 words specified by the octal translations $0 \geqslant 5_8$ or $0 \geqslant 12$ are read into the PPU. The upper four bits will be zero if word 10 is read. The upper eight bits will be zero if word 5 is read in a 64-bit register.

5. External Set or Clear

Sets or clears one of the lower 13 bits from external sources. In the 128-bit register, bits 64 - 76 may also be set from external sources. Bit 0 will be assigned as the "power off bit" and will set when the input power to the MG drops. The power to the computer will drop approximately 500 milliseconds after bit 0 sets.

6. Clear All

Clears all 64 or 128 bits.

7. Test All

Tests all 64 or 128 bits and sends the PPU a status of "1" if one or more bits are set.

8. Simultaneous Operations

Test/Set	A test is made on the bit with the bit ending up set.
Test/Clear	A test is made on the bit with the bit ending up clear.

INSTRUCTION FORMATS

Instruction Format		Instruction Code	
LDC	2000	0XXX	Read
	XXXX	1XXX	Test
OAN 15s	7215	2XXX	Clear
IAN [illegible]	7015	3XXX	Test/Clear
		4XXX	Set
		5XXX	Test/Set
		6XXX	Clear All
		7XXX	Test All

Where XXXX is the descriptor word:

Instruction Code	N.U.	N.U.	Octal Translations
11 10 9	8	7	6 5 4 3 2 0

Bits 7 and 8 of the Descriptor Word are reserved for future enhancements and should be zero.

On the Set, Clear, and Clear All operations, zero is returned to the PPU.

The only way bits can be cleared in the Interlock Register is by doing a 2XXX, 3XXX, or a 6XXX.

CENTRAL MEMORY ACCESS PRIORITY - CMAP

CMAP's Effect on ECS Transfer

The Central Memory Access Priority {CMAP} is another standard hardware feature on the CYBER 70 machines models 72,73 and 74. Its primary function is to improve CM-ECS transfer rate by allowing only °priority° CM accesses {read/write} by a PPU to be honored during an ECS transfer. It thus ensures that CM-ECS transfers are maintained at the maximum rate for a given configuration. Without CMAP, any PPU CM request can interrupt an ECS transfer. A maximum of one PPU request can be honored every ECS record or eight CM words. This could reduce the transfer rate of a large ECS system up to 75 percent, although a small {128K} system is not affected.

CMAP prevents non-priority reads or writes from entering the read or write pyramid while ECS is active. But when ECS becomes active, it is possible that some writes could be trapped in the write pyramid. This writes that are hung in the write pyramid will not be serviced until ECS transfer is complete. However, if a priority write appears during this time, CMAP will interrupt the ECS transfer and the priority write as well as any non-priority writes that were trapped in the write pyramid will be serviced.

CMAP's Effect on PPU Read/Write

By allowing only priority read/write to access central memory, CMAP provides a PPU with an opportunity to access CM at a much improved rate during an ECS transfer. When ECS is inactive, CMAP allows a PPU to place a reservation for the read/write pyramid if it failed to gain access to the pyramid on its initial request. This ensures that a PPU with a priority request will gain entrance to the pyramid within a few major cycles {microseconds}. To achieve this, a basic change is made to the read pyramid to allow data to flow unrestricted through the pyramid and give all PPU an equal chance of getting into the read pyramid. This change applies to both priority and non-priority CM accesses.

CMAP's Effect on SCOPE 3.4

The SCOPE 3.4 rotating mass storage device stack processor {1SP/1EP} uses CMAP priority for one word CM accesses within certain time critical loops. Such priority CM accesses may interrupt an ECS transfer, but will not delay it significantly, since only a single word is being read/written. At the same time, the stack processor can continue executing, rather than waiting for the ECS transfer to complete. The stack processor would be required to wait unnecessarily if CMAP priority was not used for the CRD/CWD instructions.

The stack processor does not use priority for CM block transfers. This means that CMAP will prevent stack processor CRM/CWM instructions from starting if an ECS transfer is in progress. The reasoning here is that if the PPU CM block transfer and the ECS transfer were allowed to occur simultaneously, both would be slowed and lost disk revolutions would probably result. It is more efficient to allow the ECS transfer to complete, and then honor the PPU CM block transfer.

SECTION FOUR

INSTRUCTIONS

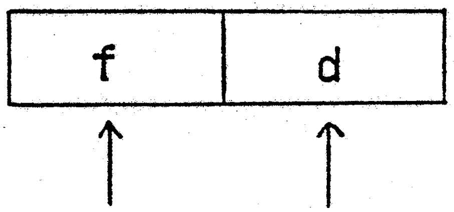
SECTION FOUR - INSTRUCTIONS

CONTENTS

INSTRUCTION FORMATS

INSTRUCTIONS OCCUPY ONE OR TWO WORDS

1.

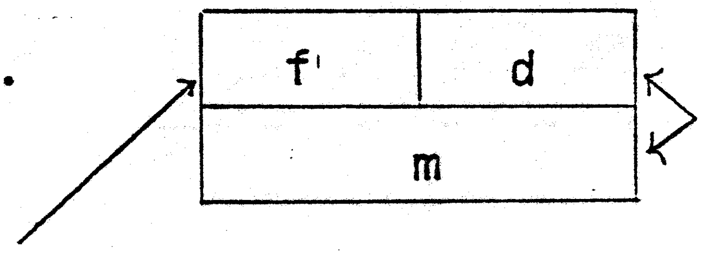


OP CODE

DELTA: OPERAND OR
OPERAND ADDRESS

THESE ARE FOR INSTRUCTIONS CONTAINING
SMALL CONSTANTS OR
REFERENCING DIRECT CELLS

2.



OP CODE

- 18-BIT OPERAND <u>OR</u>
- OPERAND ADDRESS:
  - DELTA = 0
    M IS DIRECT ADDRESS
  - DELTA ≠ 0
    D CELL CONTAINS
    INDEX FOR ADDRESS M

ca
2/12

INSTRUCTION EXAMPLES

14 | 12	LDN	10
30 | 02	LDD	2
40 | 20	LDI	20B
20 | 12 3456	LDC	123456B
20 | 00 0012	LDC	10
50 | 00 0773	LDM	773B
50 | 00 0074	LDM	74B
50 | 70 0773	LDM	773B,70B

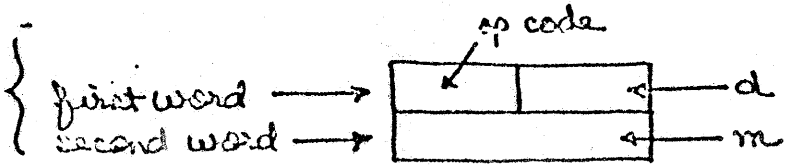
2/1/72

ADDRESSING MODES

MODE	OPERAND ADDRESS	OPERAND	EXAMPLE			
NO ADDRESS	P lower 6 bits	d	LDN	2	14 02	number 2 → A
CONSTANT	P lower 6 bits and P+1	d+m	LDC	123456B	20 12 34 56	number 123456B → A
DIRECT	d	(d)	LDD	74B	30 74	contents of loc 74 → A
MEMORY d=0	m	(m)	LDM	773B	50 00 07 73	contents of loc 773 → A
MEMORY INDEXED d≠0 *	m+(d)	(m+(d))	LDM	773B,70B	50 70 07 73	if contents of loc 70 is 1, then contents of loc 774 → A
INDIRECT **	(d)	((d))	LDI	20B	40 20	if contents of loc 20 is 1500, then contents of loc 1500 → A

* indexing is only thru direct cells and only on memory mode || there is no indexed indirect addressing
** indirect addressing is only thru direct cells + only one level

instruction format



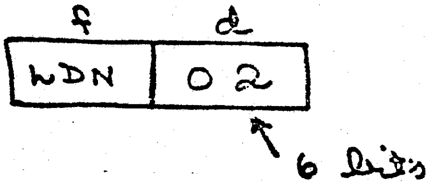
Address Modes

R p. 4-6
H –

- 5 modes of addressing
- difference is
  "where the operand comes from"

N - **N**o address - one word

– LD**N** 2

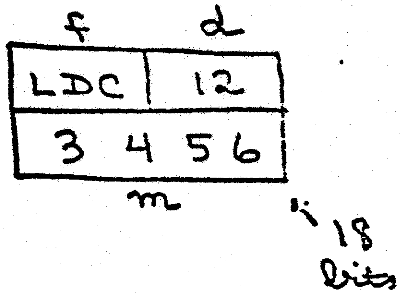


operand is in delta

N mode is for small constants (6 bits)

C - **C**onstant mode - two words

LD**C** 123456B



operand is in d+m

C mode is for up to 18-bit constants

D - __D__irect - one word

LD__D__ 74B

f	d
LDD	74

delta is the address of the operand
(contents of loc 7~~5~~4 goes into A)

D mode is for loading
from the direct cells

M - __M__emory mode - two words

LDM 773B

f	d
LDM	00
0773	
m	

m is the address of the operand
(contents of loc 773 goes into A)

* address 7777B may not be addressed *
(the instruction is assembled, but
is a NØP at execution time)

indexed memory mode:

LDM 773B, 70B

f	d
LDM	70
0 7 7 3	

m

d is a direct cell - 70 -
its contents are added to 773
to get address of operand

ie if (70) = 1
load will be from 774

I - Indirect - one word

LDI 20B

f	d
LDI	20

d is the address of the address of op.

d is a direct cell - it contains addr.
ie if (20) is 1500
load will be from 1500

* Indirect addressing goes only thru direct cells
* It is not chained (only one level)

INSTRUCTIONS

Loads and Stores

LDN	LCN	
LDC		
LDD		STD
LDM		STM
LDI		STI

Adds and Subtracts

ADN	SBN
ADC	
ADD	SBD
ADM	SBM
ADI	SBI

Replace Adds

RAD	AOD	SOD
RAM	AOM	SOM
RAI	AOI	SOI

Shift

SHN

Logicals

LMN	LPN	SCN
LMC	LPC	
LMD		
LMM		
LMI		

Jumps

UJN	ZJN
LJM	NJN
RJM	PJN
	MJN

Miscellaneous

PSN
EXN
MXN
RPN

MAN (Cybers only)

Central Memory Reads/Writes

Peripheral I/O

cw
2/77

ADC 6/71

ACN	D	7400	ACTIVATE CHANNEL DD (PRECEDES INSTRUCTION CODES 70-73)
ADC	C	21CC CCCC	ADD 18-BIT CONSTANT, CCCCCC, TO (A)
ADD	D	3100	ADD 12-BIT POS. QUANTITY, (DD), TO (A)
ADI	D	4100	ADD 12-BIT POS. QUANTITY, ((DD)), TO (A)
ADM	M,D	5100 MMMM	ADD 12-BIT POS. QUANTITY, (MMMM+(DD)), TO (A)
ADN	D	1600	ADD 6-BIT POS. QUANTITY, DD, TO (A)
AJM	M,D	6400 MMMM	JUMP TO LOCATION MMMM IF CHANNEL DD IS ACTIVE
AOD	D	3600	REPLACE ADD 1 TO (DD). RESULT ALSO IN LOW 12-BITS OF A
AOI	D	4600	REPLACE ADD 1 TO ((DD)). RESULT ALSO IN LOW 12-BITS OF A
AOM	M,D	5600 MMMM	REPLACE ADD 1 TO (MMMM+(DD)). RESULT ALSO IN LOW 12-BITS OF A
CRD	C	6000	CENTRAL READ FROM (A) TO LOCATION DD.
CRM	M,D	6100 MMMM	CENTRAL READ (DD) WORDS FROM (A) TO LOCATION MMMM.
CWD	D	6200	CENTRAL WRITE TO (A) FROM LOCATION DD.
CWM	M,D	6300 MMMM	CENTRAL WRITE (DD) WORDS TO (A) FROM LOCATION MMMM.
DCN	D	7500	DISCONNECT CHANNEL DD. I/O EQUIP. STOPS AND BUFFER TERMINATES
EJM	M,D	6700 MMMM	JUMP TO LOCATION MMMM IF CHANNEL DD IS EMPTY
FAN	D	7600	SEND FUNCTION CODE, IN LOWER 12-BITS OF (A), ON CHANNEL DD.
FJM	M,D	6600 MMMM	JUMP TO LOCATION MMMM IF CHANNEL DD IS FULL
FNC	C,D	7700 CCCC	SEND FUNCTION CODE, CCCC, ON CHANNEL DD.
IAM	M,D	7100 MMMM	INPUT (A) WORDS TO LOCATION MMMM FROM CHANNEL DD.
IAN	D	7000	INPUT 1 WORD FROM CHANNEL DD TO LOW 12-BITS OF A REG.
IJM	M,D	6500 MMMM	JUMP TO LOCATION MMMM IF CHANNEL DD IS INACTIVE.
LCN	D	1500	LOAD COMPL. OF 6-BIT CONSTANT DD (UPPER 12-BITS OF A REG.=1)
LDC	C	20CC CCCC	LOAD A REG. WITH 18-BIT CONSTANT CCCCCC
LDD	D	3000	LOAD A REG. WITH (DD). UPPER 6 BITS OF A REG.=0
LDI	D	4000	LOAD A REG. WITH ((DD)). UPPER 6 BITS OF A REG.=0
LDM	M,D	5000 MMMM	LOAD A REG. WITH (MMMM+(DD)). UPPER 6 BITS OF A REG.=0
LDN	D	1400	LOAD A REG. WITH 6-BIT CONSTANT, DD. UPPER 6 BITS OF A REG.=0
LJM	M,D	0100 MMMM	JUMP TO LOCATION MMMM+(DD). (DD MAY BE 0 OR OMITTED)
LMC	C	23CC CCCC	EXCLUSIVE OR, (A) AND 18-BIT CONSTANT CCCCCC
LMD	D	3300	EXCLUSIVE OR, LOWER 12-BITS OF (A) AND (DD)
LMI	D	4300	EXCLUSIVE OR, LOWER 12-BITS OF (A) AND ((DD))
LMM	M,D	5300 MMMM	EXCLUSIVE OR, LOWER 12-BITS OF (A) AND (MMMM+(DD))
LMN	D	1100	EXCLUSIVE OR, LOWER 6-BITS OF (A) AND 6-BIT CONSTANT DD
LPC	C	22CC CCCC	AND, (A) AND 18-BIT CONSTANT CCCCCC
LPN	D	1200	AND, (A) AND 6-BIT CONSTANT DD. UPPER 12 BITS OF A REG.=0
MJN	R	07RR	JUMP, RR LOCATIONS, IF (A) ≤ -0, (-31 ≤ RR ≤ 31)
NJN	R	05RR	JUMP, RR LOCATIONS, IF (A) ≠ 0, (-31 ≤ RR ≤ 31) (-0 ≠ +0)
OAM	M,D	7300 MMMM	OUTPUT (A) WORDS FROM LOCATION MMMM ON CHANNEL DD
OAN	D	7200	OUTPUT 1 WORD FROM LOW 12-BITS OF A ON CHANNEL DD
PJN	R	06RR	JUMP, RR LOCATIONS, IF (A) ≥ 0, (-31 ≤ RR ≤ 31)
PSN		0000	PASS (NOP), (MAY ALSO BE INSTRUCTION CODES 2400 OR 2500)
RAD	D	3500	(DD) + (A) TO DD AND A REG. LOWER 12-BITS OF (A) ARE STORED
RAI	D	4500	((DD)) + (A) TO (DD) AND A REG. LOWER 12-BITS OF A ARE STORED
RAM	M,D	5500 MMMM	(MMMM+(DD)) + (A) TO LOCATION MMMM+(DD) AND A REG.
RJM	M,D	0200 MMMM	STORE P+2 AT LOCATION MMMM+(DD) AND JUMP TO LOC. MMMM+(DD)+1
SBD	D	3200	SUBTRACT 12-BIT POS. QUANTITY, (DD), FROM (A)
SBI	D	4200	SUBTRACT 12-BIT POS. QUANTITY, ((DD)), FROM (A)
SBM	M,D	5200 MMMM	SUBTRACT 12-BIT POS. QUANTITY, (MMMM+(DD)), FROM (A)
SBN	D	1700	SUBTRACT 6-BIT POS. QUANTITY, DD, FROM (A)
SCN	D	1300	SELECTIVELY CLEAR THE LOWER 6-BITS OF (A) CORRESPONDING TO DD
SHN	D	1000	SHIFT (A), DD BITS. +=LEFT CIRC; -=RIGHT, END-OFF NO SIGN EXT
SOD	D	3700	REPLACE SUBTRACT 1 TO (DD). RESULT ALSO IN LOW 12-BITS OF A
SOI	D	4700	REPLACE SUBTRACT 1 TO ((DD)). RES. ALSO IN LOW 12-BITS OF A
SOM	M,D	5700 MMMM	REPLACE SUBTR. 1 FROM (MMMM+(DD)). RESULT ALSO IN A REG.
STD	D	3400	STORE LOWER 12-BITS OF (A) IN LOCATION DD
STI	D	4400	STORE LOWER 12-BITS OF (A) IN ((DD))
STM	M,D	5400 MMMM	STORE LOWER 12-BITS OF (A) IN (MMMM+(DD))
UJN	R	03RR	UNCONDITIONAL JUMP, RR LOCATIONS (-31 ≤ RR ≤ 31)
ZJN	R	04RR	JUMP, RR LOCATIONS, IF (A)=0 (-31 ≤ RR ≤ 31)

48

Reference List of All Instructions

MO

```
0000             PSN          PASS  (NOP),  (MAY ALSO BE INSTRUCTION CODES 2400 OR 2500)
0100 MMMM        LJM  M,D     JUMP TO LOCATION MMMM+(DD).   (DD MAY BE 0 OR OMITTED)
0200 MMMM        PJM  M,D     STORE P+2 AT LOCATION MMMM+(DD) AND JUMP TO LOC. MMMM+(DD)+1
03RR             UJN  R       UNCONDITIONAL JUMP, RR LOCATIONS  (-31 ≤ RR ≤ 31)
04RR             ZJN  R       JUMP, RR LOCATIONS, IF (A)=0  (-31 ≤ RR ≤ 31)
05RR             NJN  R       JUMP, RR LOCATIONS, IF (A) ≠ 0, (-31 ≤ RR ≤ 31)     (-0 ≠ +0)
06RR             PJN  R       JUMP, RR LOCATIONS, IF (A) ≥ 0,  (-31 ≤ RR ≤ 31)
07RR             MJN  R       JUMP, RR LOCATIONS, IF (A) ≤ -0, (-31 ≤ RR ≤ 31)
1000             SHN  D       SHIFT (A), DD BITS. +=LEFT CIRC; -=RIGHT, END-OFF NO SIGN EXT
1100             LMN  D       EXCLUSIVE OR, LOWER 6-BITS  OF (A) AND 6-BIT CONSTANT DD
1200             LPN  D       AND, (A) AND 6-BIT CONSTANT DD.  UPPER 12 BITS OF A REG.=0
1300             SCN  D       SELECTIVELY CLEAR THE LOWER 6-BITS OF (A) CORRESPONDING TO DD
1400             LDN  D       LOAD A REG. WITH 6-BIT CONSTANT, DD. UPPER 6 BITS OF A REG.=0
1500             LCN  D       LOAD COMPL. OF 6-BIT CONSTANT DD (UPPER 12-BITS OF A REG.=1)
1600             ADN  D       ADD 6-BIT POS. QUANTITY, DD, TO (A)
1700             SBN  D       SUBTRACT 6-BIT POS. QUANTITY, DD, FROM (A)
20CC CCCC        LDC  C       LOAD A REG. WITH 18-BIT CONSTANT CCCCCC
21CC CCCC        ADC  C       ADD 18-BIT CONSTANT, CCCCCC, TO (A)
22CC CCCC        LPC  C       AND, (A) AND 18-BIT CONSTANT CCCCCC
23CC CCCC        LMC  C       EXCLUSIVE OR, (A) AND 18-BIT CONSTANT CCCCCC
3000             LDD  D       LOAD A REG. WITH (DD).  UPPER 6 BITS OF A REG.=0
3100             ADD  D       ADD 12-BIT POS. QUANTITY, (DD), TO (A)
3200             SBD  D       SUBTRACT 12-BIT POS. QUANTITY, (DD), FROM (A)
3300             LMD  D       EXCLUSIVE OR, LOWER 12-BITS OF (A) AND (DD)
3400             STD  D       STORE LOWER 12-BITS OF (A) IN LOCATION DD
3500             RAD  D       (DD) + (A) TO DD AND A REG.  LOWER 12-BITS OF (A) ARE STORED
3600             AOD  D       REPLACE ADD 1 TO (DD).  RESULT ALSO IN LOW 12-BITS OF A
3700             SOD  D       REPLACE SUBTRACT 1 TO (DD).  RESULT ALSO IN LOW 12-BITS OF A
4000             LDI  D       LOAD A REG. WITH ((DD)).  UPPER 6 BITS OF A REG.=0
4100             ADI  D       ADD 12-BIT POS. QUANTITY, ((DD)), TO (A)
4200             SBI  D       SUBTRACT 12-BIT POS. QUANTITY, ((DD)), FROM (A)
4300             LMI  D       EXCLUSIVE OR, LOWER 12-BITS OF (A) AND ((DD))
4400             STI  D       STORE LOWER 12-BITS OF (A) IN ((DD))
4500             RAI  D       ((DD)) + (A) TO (DD) AND A REG. LOWER 12-BITS OF A ARE STORED
4600             AOI  D       REPLACE ADD 1 TO ((DD)).  RESULT ALSO IN LOW 12-BITS OF A
4700             SOI  D       REPLACE SUBTRACT 1 TO ((DD)).   RES. ALSO IN LOW 12-BITS OF A
5000 MMMM        LDM  M,D     LOAD A REG. WITH (MMMM+(DD)).  UPPER 6 BITS OF A REG.=0
5100 MMMM        ADM  M,D     ADD 12-BIT POS. QUANTITY, (MMMM+(DD)), TO (A)
5200 MMMM        SBM  M,D     SUBTRACT 12-BIT POS. QUANTITY, (MMMM+(DD)), FROM (A)
5300 MMMM        LMM  M,D     EXCLUSIVE OR, LOWER 12-BITS OF (A) AND (MMMM+(DD))
5400 MMMM        STM  M,D     STORE LOWER 12-BITS OF (A) IN (MMMM+(DD))
5500 MMMM        RAM  M,D     (MMMM+(DD)) + (A) TO LOCATION MMMM+(DD) AND A REG.
5600 MMMM        AOM  M,D     REPLACE ADD 1 TO (MMMM+(DD)). RESULT ALSO IN LOW 12-BITS OF A
5700 MMMM        SOM  M,D     REPLACE SUBTR. 1 FROM (MMMM+(DD)).  RESULT ALSO IN A REG.
6000             CRD  D       CENTRAL READ FROM (A) TO LOCATION DD.
6100 MMMM        CRM  M,D     CENTRAL READ (DD) WORDS FROM (A) TO LOCATION MMMM.
6200             CWD  D       CENTRAL WRITE TO (A) FROM LOCATION DD.
6300 MMMM        CWM  M,D     CENTRAL WRITE (DD) WORDS TO (A) FROM LOCATION MMMM.
6400 MMMM        AJM  M,D     JUMP TO LOCATION MMMM IF CHANNEL DD IS ACTIVE
6500 MMMM        IJM  M,D     JUMP TO LOCATION MMMM IF CHANNEL DD IS INACTIVE.
6600 MMMM        FJM  M,D     JUMP TO LOCATION MMMM IF CHANNEL DD IS FULL
6700 MMMM        EJM  M,D     JUMP TO LOCATION MMMM IF CHANNEL DD IS EMPTY
7000             IAN  D       INPUT 1 WORD FROM CHANNEL DD TO LOW 12-BITS OF A REG.
7100 MMMM        IAM  M,D     INPUT (A) WORDS TO LOCATION MMMM FROM CHANNEL DD.
7200             OAN  D       OUTPUT 1 WORD FROM LOW 12-BITS OF A ON CHANNEL DD
7300 MMMM        OAM  M,D     OUTPUT (A) WORDS FROM LOCATION MMMM ON CHANNEL DD
7400             ACN  D       ACTIVATE CHANNEL DD  (PRECEDES INSTRUCTION CODES 70-73)
7500             DCN  D       DISCONNECT CHANNEL DD. I/O EQUIP. STOPS AND BUFFER TERMINATES
7600             FAN  D       SEND FUNCTION CODE, IN LOWER 12-BITS OF (A), ON CHANNEL DD.
7700 CCCC        FNC  C,D     SEND FUNCTION CODE, CCCC, ON CHANNEL DD.
```

C2
2/72

LOAD AND STORE INSTRUCTIONS

LDN - load a 6-bit constant into A

LCN - load a 6-bit (complemented) constant into A

LDC - load an 18-bit constant into A

LDD - load contents of direct cell into A

STD - store contents of A into direct cell

LDM - load contents of memory cell into A

STM - store contents of A into memory cell

LDI - load contents of any cell into A, indirectly thru a direct cell

STI - store contents of A into any cell, indirectly thru a direct cell

ADD and SUBTRACT INSTRUCTIONS

ADN	SBN
ADC	
ADD	SBD
ADM	SBM
ADI	SBI

Adds:

a 6-, 12-, or 18-bit number
is added to the contents of A

Subtracts:

a 6- or 12-bit number
is subtracted from A

* 18-bit arithmetic is used *

co 2/72

REPLACE ADD INSTRUCTIONS

RAD	AOD	SOD
RAM	AOM	SOM
RAI	AOI	SOI

- a 12-bit number (unsigned)
  may be added to the A register
  and to the memory location
- the number 1
  may be added to or subtracted from
  the A register
  and the memory location

* 18-bit arithmetic is used *

* the result is in A and memory *

2/72

LOGICAL INSTRUCTIONS

LMN LPN SCN
LMC LPC
LMD
LMM
LMI

0101
0011 (A)
0110

exclusive or:

a 6-, 12, or 18-bit number
may be exclusive ored with A

and:

only a 6- or 18-bit constant
may be anded with A

selective clear:

a 6-bit constant
may selectively clear bits in A

ce
2/72

SHIFT INSTRUCTION

SHN

LEFT shifts:

- end-around
- ie SHN 6
  shifts A left 6 bits

```
before:  (A) = 000005
after:   (A) = 000500
```

RIGHT shifts:

- end-off
  <u>no</u> sign extension
- ie SHN -6
  shifts A right 6 bits
  zeroes are filled in from the left

```
before:  (A) = 777774
after:   (A) = 007777   ← note!
```

ce
2/72

62

JUMP INSTRUCTIONS

unconditional:

UJN - jumps up to 37_8 locations
forward or backward

LJM - jumps any number of locations
forward or backward

RJM - jumps to any location
plants return address there
executes instruction in next word

conditional:

ZJN - jumps if A is +0

NJN - jumps if A is not +0

PJN - jumps if A is positive

MJN - jumps if A is negative

* conditional jumps are
up to 37_8 locations
forward or backward

* address field may contain
a number
(jump is to P+n)

or a location symbol
(jump is to the location)

aw
2/72

mo

Instruction Problem Set 1

Contents of the following core locatiors are given.

All numbers are in octal.

{0025} = 1234
{0034} = 1111
{0100} = 1111
{0125} = 2222
{1111} = 7777
{1234} = 4321
{1334} = 3333
{1361} = 1234
{2345} = 4444

Work <u>each question indepently.</u>

Show the contents of the A register after the instruction has been executed.

		{A} Register
1)	LDN 25	
2)	LDD 25	
3)	LDM 100	
4)	LDM 100,25	
5)	LDI 25	
6)	LDM 0,25	
7)	LDC -100	
8)	LDN -25	
9)	LDM 1111,25	
10)	LCN 25	

11. How many words of code do the above instructions generate? ____________

On the following store instructions, indicate:

The contents of the A register after the store

The contents of the core location

The location at which the data is stored

```
12}  LDC  1234568          {A}   = ________________
     STD  258              {loc} = ________________
                           loc is  ________________

13}  LDC  1234568          {A}   = ________________
     STM  10008            {loc} = ________________
                           loc is  ________________

14}  LDC  1234568          {A}   = ________________
     STI  258              {loc} = ________________
                           loc is  ________________

15}  LDC  1234568          {A}   = ________________
     STM  10008,258        {loc} = ________________
                           loc is  ________________

16}  LDD  258              {A}   = ________________
     ADN  258              {A}   = ________________
```

Instruction Problem Set 2

Use the data given in Problem Set 1

Work each problem independently

{Note thac each problem has severel instructions to be worked cumulatively}

Indicate the contents of A after each instruction is executed

{The problems are numbers 17-33 on the following page}

```
17}  LDI   34B              {A} = ____________
     SBN   6                {A} = ____________

18}  LDM   100B             {A} = ____________
     ADD   100B             {A} = ____________

19}  LDD   34B              {A} = ____________
     SBD   34B              {A} = ____________

20}  LDM   1234B            {A} = ____________
     ADI   25B              {A} = ____________
     SBI   34B              {A} = ____________

21}  LDM   1234B,34B        {A} = ____________
     ADC   123456B          {A} = ____________
     STM   25B            {loc} = ____________

22}  LDM   100B             {A} = ____________
     ADM   100B             {A} = ____________
     ADM   100B,25B         {A} = ____________

23}  LDM   125B,25B         {A} = ____________
     SBM   125B             {A} = ____________
     SBM   100B,25B         {A} = ____________

24}  LDC   123456B          {A} = ____________
     SHN   14B              {A} = ____________
     SHN   -11B             {A} = ____________
     SHN   36B              {A} = ____________

25}  LDM   1361B            {A} = ____________
     LMN   46B              {A} = ____________

26}  LDC   2345B            {A} = ____________
     LPN   1234B            {A} = ____________

27}  LDI   25B              {A} = ____________
     SCN   25B              {A} = ____________

28}  LDC   123456B          {A} = ____________
     LMD   25B              {A} = ____________

29}  LDN   22B
     SHN   14B
     ADN   55B
     LMI   34B              {A} = ____________

30}  LDC   123456B
     SHN    -6
     LPC   123456B          {A} = ____________

31}  LDC   123456B
     SHN   -11B
     LMC   123456B          {A} = ____________

32}  LDM   100B
     SHN   6
     LMM   34B              {A} = ____________

33}  LDM   100B,25B
     SHN   6
     STM   34B,25B          {A} = ____________
```

Instruction Problem Set 3

Use the data given in the core locations in Problem Set 1.

The A register contains $\boxed{123456_8}$

Work each problem independently.

Indicate the contents of the core location and A register after each instruction is executed.

1. RAD 34B {34} = ____________ {A} = ____________
2. AOD 34B {34} = ____________ {A} = ____________
3. SOD 25B {25} = ____________ {A} = ____________
4. RAI 25B {loc} = ____________ {A} = ____________
   loc is ____________
5. AOI 343 {loc} = ____________ {A} = ____________
   loc is ____________
6. SOI 343 {loc} = ____________ {A} = ____________
   loc is ____________
7. RAM 1234B {1234} = ____________ {A} = ____________
8. AOM 1234B {1234} = ____________ {A} = ____________
9. SOM 1111B {1111} = ____________ {A} = ____________
10. RAM 100B,25B {loc} = ____________ {A} = ____________
    loc is ____________
11. AOM 253,34B {loc} = ____________ {A} = ____________
    loc is ____________

CU 2/72

ANSWERS - PROBLEM SETS

SET #1

1.	25
2.	1234
3.	1111
4.	3333
5.	4321
6.	4321
7.	777~~767~~677
8.	error
9.	4444
10.	777752
11.	15
12.	123456 3456 loc 25
13.	123456 3456 loc 1000
14.	123456 3456 in 1234
15.	123456 3456 loc 2234
16.	1234 1261

SET #2

17.	008888 007771
18.	001111 error
19.	001111 000000
20.	004321 010642 000643
21.	004444 130122 0122 in loc 25
22.	001111 002222 005555
23.	001234 777011 773456
24.	123456 561234 000561 610005
25.	001234 001272
26.	002345 error
27.	004321 004300
28.	123456 122662
29.	227722
30.	001014
31.	123575
32.	110011
33.	333300 3300 in 1270

SET #3

1.	4567	124567
2.	1112	001112
3.	1233	001233
4.	7777 1234	127777 ~~010000~~
5.	0000 loc 1111	010000
6.	7776 loc 1111	007776
7.	7777	127777
8.	4322	004322
9.	7776	007776
10.	7011 loc 1334	127011
11.	? loc 1136	(1136+1)

CODING EXAMPLES

1.

```
             MJN      ABT
          (  code        )
          {  100         }
          {instruction   }
          (  words       )
     ABT     :
```

What does this code do?

2. How to fix a jump that won't reach:

```
             PJN      *+3
             LJM      ABT
          {  code:  }
          {  100    }
          {  words  }
     ABT     ⌇
```

3. HOW THE RETURN JUMP INSTRUCTION WORKS:

```
             ORG      10008
             LDN      1
             STD      4  SUB
             RJM      TAG,4
             :
    SUB
     TAG     DATA     100
             DATA     0
```

a. {TAG} = ____________

b. {TAG+1} = ____________

c. Where does execution continue? ____________

d. Why is P+2 the return address?

____

4.

```
             :
             STM      *+3
             LJM      TAG
             :
```

Where does the STM store?

____

ce 8/71

5. HOW TO MAKE A LOOP

```
         LDN      4
         STD      14B
LP       -
         -
         SOD      14B
         NJN      LP
```

How many times is this loop executed?

____

6. HOW NOT TO MAKE A LOOP

```
         LCN      4
         STD      10B
LP       -
         -
         AOD      10B
         NJN      LP
```

Why is this a hung loop?

____

____

7. HOW TO MOVE CORE

```
      Given:  {15} = 1000B
              {17} = 2000B
```

In each example, how many words are moved, and from what locations to where?

```
a.             LDM      0,15B
               STM      1000B,15B

b.             LDI      15B
               STI      17B

c.             LDC      1000B
               STD      1
     LOOP      LDI      1
               STM      1000B,1
               AOD      1
               ADC      -1077B
               MJN      LOOP
```

a. ________________________

b. ________________________

c. ________________________

8. HOW TO MAKE AN INCLUSIVE OR

```
        LDM     TAG1
        STM     AND+1
        LDM     TAG
AND     LPC     **
        STD     2
        LDM     TAG
        LMM     TAG1
        ADD     2

TAG     DATA    1264B
TAG1    DATA    4444B
```

a. What is the inclusive OR of 1264 and 4444?

____

b. Is it necessary to load TAG1 and store it in the LPC instruction? ______

c. What value does ** assemble as?

____

9. HOW TO ADD 18-BIT NUMBERS

Given:

```
(IN   )= 0012
(IN+1 )= 7756
```

IN and IN+1 represent an 18-bit relative CM address which must be updated

```
LDC     100B
RAM     IN+1
SHN     -12
RAM     IN
```

(ie $\begin{array}{r} 127756 \\ \underline{100} \\ 130056 \end{array}$ ← 100_8 is 64_{10} 1 pru)

10. ALGORITHM FOR WORKING WITH 12-BIT SIGNED NUMBERS IN MEMORY

```
a.  DATA1        DATA       1234B,7643B,0005B
    DATA2        DATA       2000B,7523B,7766B
    ANSWER       BSSZ       3
                  }
                 LDN        2
                 STD        2
    ADD          LDC        770000B
                 ADM        DATA1,2
                 ADM        DATA2,2
                 STM        ANSWER,2
                 SOD        2
                 PJN        ADD

b.  X            VFD        12/DATA1
    Y            VFD        12/DATA2
    Z            VFD        12/ANSWER
                  }
                 DUP        3,7
                 LDC        770000B
                 ADI        X
                 ADI        Y
                 STI        Z
                 AOD        X
                 AOD        Y
                 AOD        Z
                  }
    DATA1        DATA       1234B,7643B,0005B
    DATA2        DATA       2000B,7523B,7766B
    ANSWEF       BSSZ       3
```

Question: What type of numbers are in:

DATA1 __________ __________ __________

DATA2 __________ __________ __________

ANSWER __________ __________ __________

11. INTEGER MULTIPLY

```
         LDC      800
         STD      1
         SHN      4
         SBD      1
         SBD      1
         SBD      1
         STM      ANS+1
         SHN      -12
         STM      ANS
          ≀
ANS      BSSZ     2
```

a. What are the numbers multiplied?

____

b. How many words of core does this code occupy?

____

12. INTEGER DIVIDE

```
DVDN     EQU      10B
DVSR     EQU      11B
ANS      EQU      12B
REM      EQU      13B
          ≀
         LDN      0
         STD      ANS
DIV      LDD      DVDN
         SBD      DVSR
         MJN      OUT
         STD      DVDN
         AOD      ANS
         UJN      DIV
OUT      ADD      DVSR
         STD      REM
          ≀
DVDN     DATA     30
DVSR     DATA     13
ANS      BSSZ     1
REM      BSSZ     1
```

a. What numbers are divided?

____

b. What is the answer?

____

c. What is the remainder?

____

{10} = 30

{11} = 13

ANSWERS TO CODING EXAMPLES

1. Assembly error.
   ABT is too far away.

2. The PJN jumps around the LJM.
   The LJM can reach ABT.

3. a. 100
   b. 1004
   c. TAG+2
   d. RJM is two words

4. Second word of LJM instruction.

5. four

6. The AOD in this case always leaves (A) $\neq 0$

7. a. one word - from 1000 to 2000
   b. one word - from 1000 to 2000
   c. 77_8 words - from 1000+ to 2000+

8. a. 5664
   b. yes - there is no LPM instruction
   c. 0

9. -

10.

```
+1234     -0134     +0005
+2000     -0254     -0011
+3234     -0410     -0004
```

Solution formula:

```
770000    770000    770000
 +1234     +7643     +0005
------    ------    ------
771234    777643    770005
 +2000     +7523     +7766
------    ------    ------
773234    007366    777773
               1
          ------
          007367
```

11. a. 800 x 13
    b. 12

12. a. 30 ÷ 13
    b. 2
    c. 4

no

CENTRAL MEMORY

READ and WRITE INSTRUCTIONS

CRD - reads one word

CRM - reads a block

CWD - writes one word

CWM - writes a block

cw
2/72

FORMAT OF THE CRD

1. Reserve a PP buffer to read the data into:

   (Locations 10-14 will be used)

2. Load CM absolute address into A:

```
LDC     123400B
```

3. Execute the Central Read instruction:

```
CRD     10B
```

PP memory location 10
is beginning of a
5-word PP buffer
to contain 1 CM word

* After the Read, (A) is not destroyed *

(in this case, 123400_8)

FORMAT OF THE CRM

1. Reserve a PP buffer to read the data into:

```
BUF     BSSZ    20
```

2. Store the block length (CM words) in a direct cell:

```
        LDN     4
        STD     10B
```

3. Load the absolutized CM address in A

```
        LDC     123400B
```

4. Execute the Central Read instruction:

```
        CRM     BUF,10B
```

PP Buffer
to read into

Direct Cell
containing
CM word count

* After the Read, (A) is CM LWA+1 *

(in this case, 123404_8)

cw
2/72

7

no

Instruction Problem Set #4

1.

```
        LDC     400B
        CRD     50B
```

a. (A) = ______________________

b. What does this code do?

____

2.

```
        LDC     60000B
        CWD     10B
```

a. (A) = ______________________

b. What does this code do?

____

3.

```
        LDN     16
        STD     2
        LDC     400B
        CRM     BUF,2

BUF     BSS     80
```

When the Read is finished, but before the original P is restored, what is:

(A) = ______________________

(Q) = ______________________

(P) = ______________________

(loc 0) = ______________________

4.

```
ADDR    DATA    0021B,2222B

        LDN     5
        STD     2
        LDM     ADDR
        SHN     12
        ADM     ADDR+1
        CWM     BUF,2

BUF     BSS     25
```

What does this code do?

ce
2/72

SECTION FIVE

SYSTEM DESIGN
CODING CONVENTIONS

SECTION FIVE - CODING CONVENTIONS

CONTENTS

CODING CONVENTIONS

All the previous coding examples have been slanted toward the <u>use</u> of the instructions. Note that they used absolute CM and PP addresses and constants rather than system symbols.

<u>Good</u> PP programs should <u>always</u> be coded using system symbols. This is for two reasons:

1. The names will become recognizable so one can see what the program is doing.
2. <u>Most important</u>: When the system is changed, a program which used system symbols and proper coding setup can be reassembled and will still run.

The symbols are accessed on the COMPASS control card:

```
COMPASS (S= SCPTEXT)
```

*THIS CANNOT BE EMPHASIZED STRONGLY ENOUGH**

Symbol names on pp. 111, 112
100--103

SYMBOL DEFINITIONS

go to SYSTEXT

		EXAMPLE
D.XX	A DIRECT CELL IN PP MEMORY	D.FFO is loc 50 D.PPIRB is loc 50
R.XX	A PP RESIDENT ROUTINE	R.IDLE is loc 103
M.XX	A MONITOR FUNCTION	M.DPP is 12 M.ABORT is 13
T.XX	A SYSTEM TABLE IN CMR	T.FNT is FILE NAME TABLE
P.XX	A POINTER IN CMR TO A TABLE	P.FNT byte 0 is ADDRESS OF FNT
C.XX	A BYTE IN A CMR WORD (0-4)	C.CPRA is 3
	or	
	A PP CONSTANT	C.PPFWA is 1000
L.XX	LENGTH OF A CMR TABLE	L.TAPES is LENGTH of TAPES TABLE
	or	
	A PP LENGTH CONSTANT	L.PPHDR is 5
W.XX	A WORD IN CMR:	
	PP COMMUNICATION AREA	W.PPMES1 is WORD 2 of PP COMM AREA
	CONTROL POINT AREA	W.CPSTAT is WORD 20 of CP AREA
F.XX	AN ERROR FLAG VALUE	F.ERPP is 3 (PP Abort code passed by M.ABORT)

ca
2/72

EXAMPLES OF GOOD CODE

1. HOW TO READ THE PP INPUT REGISTER FROM CENTRAL MEMORY AND GET THE CP NUMBER AND CP AREA ADDRESS

```
LDD    D.PPIR
CRD    D.PPIRB
LDD    D.PPIRB+1
LPN    7
SHN    7
STD    D.CPAD
```

2. HOW TO GET RA AND FL AND BRING THEM INTO THE PP

```
LDD    D.CPAD
ADN    W.CPSTAT
CRD    D.TO
LDD    D.TO+C.CPRA
STD    D.RA
LDD    D.TO+C.CPFL
STD    D.FL
```

3. HOW TO ABSOLUTIZE A CENTRAL MEMORY ADDRESS

```
LDD    D.IN
SHN    6
ADD    D.RA
SHN    6
ADD    D.IN+1
```

8 2/73

4. HOW TO CLEAR 5 PP WORDS

```
LDN     P.ZERO          CM ADDR 0 CONTAINS 0
CRD     D.TO            CLEAR D.TO-D.T4
```

This is faster and uses less code than:

```
LDN     0
STD     D.TO
STD     D.T1
STD     D.T2
STD     D.T3
STD     D.T4
```

5. HOW TO CHECK TO SEE IF THE CPU IS RUNNING:

```
        RPN
        STD     2
        RPN
        LPC     7777B
        SBD     2
        NJN     OK
        LJM     TIMOUT
OK
```

6. HOW TO WRITE A PP BUFFER TO CENTRAL MEMORY

- 32 PP WORDS -

```
          LDN     7
          STD     7
          LDN     0
          STM     IN+32
          STM     IN+33
          STM     IN+34
          LDD     D.IN
          SHN     6
          ADD     D.RA
          SHN     6
          ADD     D.IN+1
          CWM     IN,7
          LDN     7
          RAD     D.IN+1
          SHN     -12
          RAD     D.IN


IN        BSS     35
```

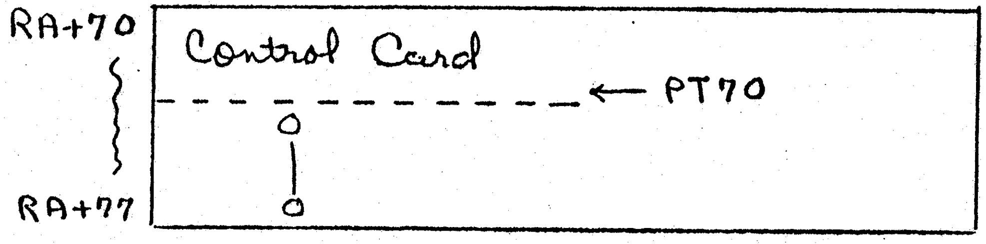
The above code wrote 35 PP words, which is 7 CM words, to Central Memory. The last 3 bytes were zero, because only whole CM words may be written.

The code also updates the IN pointer.

GOOD CODE TO STUDY
(also study R.DFM)

5-6

Given: PT70 contains a number in the range 70-77. 1AJ has just moved a control card into the RA+70 area. Examine the following code to see how 1AJ clears the rest of the area. PT70 contains the pointer to the last word stored.



```
1400                    LDN    F.ZERO          CLEAR REST OF RA+70B AREA          1AJ
6010                    CRD    D.T0                                               1AJ
3665          UNP6      ADD    PT70                                               1AJ
1377                    SCN    77B                                                1AJ
0506                    NJN  UNP5                                                 1AJ
3055                    LDD    D.RA                                               1AJ
1006                    SHN    6                                                  1AJ
3165                    ADD    PT70                                               1AJ
6210                    CWD    D.T0                                               1AJ
0370                    UJN  UNP6                                                 1AJ
```

SCOPE 3.3 & 3.4

Given: (A) contains a flag and a message buffer address: | 0X | loc. |

flag — message address in PP

R.DFM moves the message to pp locs 13+. It then moves 5 pp words (1 CM word) at a time to the PPMES area in Central Memory. Study the code.

```
                          *       R.DFM     TRANSMIT DAYFILE MESSAGE                                   STL
                          *                                                                            STL
                          *       CALLING SEQUENCE                                                     STL
                          *                                                                            STL
                          *       LOAD      (FLAG)LOCATION OF MESSAGE                                  STL
                          *       RJM       R.DFM                                                      STL
                          *                                                                            STL
                          *       ACTIONS                                                              STL
                          *                                                                            STL
                          *       TRANSMIT MESSAGE TO PP MESSAGE AREA                                  STL
                          *       CALL MONITOR FUNCTION M.DFM+(FLAG)                                   STL
                          *                                                                            STL

   0664                   R.DFM   ENM       *                                                          STL
   0665     3410                  STD       D.T0          LOCATION OF MESSAGE                          STL
   0667     1065                  SHN       -12                                                        STL
   0670     3411                  STD       D.T1          STORE FLAG                                   STL
   0671     3075                  LDD       D.PPMES1                                                   STL
   0672     3412                  STD       D.T2          SET STORAGE (+1) ADDRESS                     STL
   0673     1413          DFM2    LDN       D.T3                                                       STL
   0674     3400                  STD       0             SET ASSEMBLY ADDRESS                         STL
   0675     4010          DFM1    LDI       D.T0          MOVE BYTE                                    STL
   0676     4400                  STI       0             TO ASSEMBLY AREA                             STL
   0677     0402                  ZJN       *+2           SENSE END OF MESSAGE                         STL
   0700     3510                  AOD       D.T0          ADVANCE IN MESSAGE                           STL
   0701     3500                  AOD       0             AND ASSEMBLY AREA                            STL
   0702     1120                  LMN       D.T3+5                                                     STL
   0703     0571                  NJN       DFM1          SENSE ASSEMBLY NOT FULL                      STL
   0704     3012                  LDD       D.T2                                                       STL
   0705     6213                  CWD       D.T3          WRITE ASSEMBLY TO MESSAGE AREA               STL
   0706     3512                  AOD       D.T2          ADVANCE STORAGE ADDRESS                      STL
   0707     1207                  LPN       7                                                          STL
   0710     0403                  ZJN       DFM3          JUMP IF END OF MESSAGE AREA                  STL
   0711     3017                  LDD       D.T3+4                                                     STL
   0712     0560                  NJN       DFM2          LOOP IF NOT END OF MESSAGE AREA              STL
                                                                                                       STL
   0713     1401          DFM3    LDN       M.DFM                                                      STL
   0714     0200 6515             RJM       R.MIF         SEND DAYFILE MESSAGE                         STL
   0716     0345                  UJN       R.DFMX                                                     STL
```

57

3

SECTION SIX

PP MACROS

SECTION SIX - PP MACROS

CONTENTS

PP Macros

The following descriptions of the PP Macros are from the PSI Scope 3.3 Handbook, pages 2-104 & 2-105. The actual code is listed from the system routine SCPTEXT, Version 3.3

SCPTEXT MACROS

PPENTRY Macro

Used as first instruction following ORG in a primary level overlay. PPENTRY generates code to set up low core parameter as follows:

D.PPIRB through D.PPIRB+4	Input register contents
D.CPAD	Control point address
D.RA	Reference address/100B
D.FL	Field length/100B

Address field of the PPENTRY macro should contain: D.PPIRB, D.TO.

Code:

note T parameter required in V 3.3

```
~~PPENTRY MACRO S,T~~
   LDD D.PPIR
~~   ORG S~~
   IFEQ T,D.TO
~~   RJM R.RAFL~~
   ELSE
~~   ERR MACRO CALL ERROR~~
   ENDIF
~~PPENTRY ENDM~~
```

note a RAFL is also done

The PPENTRY macro is normally used as the first instruction in a primary overlay, for upward system compatibility.

ce 2/71

70

ENM Macro

Generates standard subroutine entry and exit lines. The name of the subroutine is that declared in location field of ENM; the subroutine may be entered by an RJM to that name. If address field of ENM is blank, no exit symbol is defined; otherwise, contents of address field are appended to location symbol to generate subroutine exit symbol. (Typically, address field contains only an X) An exit from subroutine may then be made by jumping directly to the generated symbol.

Code:

```
    MACRO ENM,N,X
    ~~IFG NE,+X++,2~~
N•X LJM
    ~~IFGP 1~~
    LJM
~~N   EQU *-1~~
~~ENM ENDM~~
```

See example on next page

HOW TO GET IN AND OUT OF A SUBROUTINE

1.

```
            RJM    SUB

SUB         ENM    X   generates: {SUBX   |0100|
                                  {SUB    |0000|

            LJM    SUBX
```

2.

```
            RJM    SUB

SUBX        LJM    **
SUB         EQU    *-1   <--- put tag SUB
                              on second word
                              of LJM instruction

            LJM    SUBX
```

cw
2/72

72

UJK Macro

Generates UJN or LJM instruction, depending on length of jump. In general, the jump must be backward, since symbols used in address field must have been previously defined. Macro is useful for exiting from small subroutines subject to expansion.

T.O.C

Code:

```
UJK MACRO P
~~  IFLT *-P,/,63,2~~
  UJN P
~~  IFCP 1~~
  LJM P
~~UJK ENDM~~
```

UJK makes a UJN or LJM

Example: UJK SUBX

LDCA Macro

Load PP A register with absolute 18-bit central address. Relative CM address is obtained from two consecutive PP low core locations, the first of which is specified in address field of LDCA macro; CM address is assumed to be right justified within these two words. Contents of D.RA are added to CM address. Macro is useful for loading many different CM addresses. Space may be conserved by using a subroutine rather than a macro if the same address is to be loaded three or more times.

Code:

```
~~LDCA MACRO A~~
  LDD A
~~  LPN 77B~~
  SHN 6
~~  ADD D.RA~~
  SHN 6
~~  ADD A+1~~
~~LDCA ENDM~~
```

Example:

LDCA D.IN

The relative cm address is obtained from D.IN + D.IN+1 and then absolutized

CRI Macro "Central Read Indirect"

Reads contents of a CM word the address of which is contained in a central memory pointer. Address field of CRI macro contains X, Y, and Z subfields, in that order.

X	6-bit CM pointer word address
Y	First of five PP low core cells which will contain the desired CM word.
Z	Byte within CM pointer word containing 12-bit CM address of desired word.

Code:

```
CRI MACRO X,Y,Z
~~    LDN  X~~
    CRD  Y
~~    LDD  Y+Z~~
    CRD  Y
~~CRI ENDM~~
```

Example:

CRI . P.EST, I.TO , O

The first word of the EST will be read into D.TO - D.T4. To search the EST, one can repeatedly add 1 to (A) and reread, since the CRD did not destroy the address of T.EST in A

BIT Macro

Generates no code; merely defines a symbol in the location field. Value assigned to symbol is a 1-bit mask where the bit is positioned according to the value of address field. Bits are counted from right to left, beginning with zero. Thus, the statement MASK BIT 2 would set MASK equal to 4. Macro is useful for generating 1-bit flag values with the S.x SCPTEXT symbols.

Code:

```
    ~~MACRO BIT,R,V~~
R SET 1
    ~~DUP V,1~~
R SET R+R
~~BIT ENDM~~
```

Example:

```
MASK    BIT    2
```

generates a constant called MASK containing a bit in bit position 2. Same as:

```
MASK    SET    4
```

LDK Macro

Generates LDN, LDC, or LCN instruction depending on size of its argument, which may be any valid address expression. This macro is recommended for referencing SCPTEXT symbols for CM pointer words.

Code:

```
~~LDK MACRO A~~
   LOCAL X
~~IFD IF DEF,A~~
X SET A
   ~~IFLT X,0,5~~
X SET -Y
   ~~IFLT X,100B,2~~
   LCN X
   ~~IFCP 6~~
   IFCP 4
   ~~IFLT X,100B,2~~
   LDN X
   ~~IFCP 2~~
IFD ENDIF
   ~~LDC A~~
   ENDM
```

ADK Macro

Generates ADN, ADC, SBN, or no code depending on size of its argument, which may be an address expression. This macro is recommended for referencing SCPTEXT symbols for control point additives {W.x symbols}.

Code:

```
~~ADK MACRO A~~
   LOCAL X
~~IFD IF DEF,A~~
 X SET A
   IFLT X,0,3
 X SET -X
   ~~SBK X~~
   IFCP 6
   ~~IFLT X,100B,3~~
   IFNE X,0,1
   ~~ADN X~~
   IFCP 2
~~IFD ENDIF~~
   ADC A
   ~~ENDM~~
```

SBK Macro

Generates SBN or ADC depending on size of its argument. All symbols in its argument must be defined.

Code:

```
SBK MACRO ARG
   ~~LOCAL CON~~
CON SET ARG
   ~~IF DEF,ARG,3~~
   IFLT CON,100B,2
   ~~SBN CON~~
   IFCP 1
   ~~ADC CON~~
   ~~ENDM~~
```

SECTION SEVEN

SYSTEM TABLES
AND POINTERS
(SCOPE 3.3 & 3.4)

SECTION SEVEN-SYSTEM TABLES AND POINTERS

CONTENTS

100

SCOPE 3.3
CMR

0			
57		Pointers	
60	T.PPC 1	PP Communications Area	
70	2		
⋮	⋮		
200	T.CPA 1	Control Point Area	
400	2		
⋮	⋮		
		CP Resident	
		Stack Processor Manager	▨
	T.EST	Equipment Status Table	
	T.CHTIM	Channel I/O Time (Accounting Package)	▨
	T.CST	Channel Status Table	
	T.FNT	File Name Table	
*	T.APF	Attached Permanent File Table	
	T.SDT	Permanent File Subdirectory Table	
	T.DAT	Device Activity Table	▨
*	T.RQS	Request Stack	▨
	T.RBR	RBR Header Words	▨
	T.RBRBIT	RBR Bit Tables	▨
	T.DST	Device Status Table	▨
	T.SEQ	Sequencer Table	
	T.RMS	RMS Preallocation Table	▨
	T.INS	Installation Area	
	T.ITABL	INTERCOM Table	▨
	T.NOVA	NOVA Table (GP250)	
	T.TAPES	Tapes Table	▨
	T.DFB	Dayfile Buffers	
	T.ECSBUF	ECS Storage Move Buffer	
	T.ICEBUF	Icebox Buffer	
	T.ECST	ECS Information Table	
	T.ECSTAT	ECS Statistic Table	
	T.ECFLAW	ECS Flaw Table	
	T.LRD	Logical Record DEF Table	
	T.ECLBUF	ECS Resident Library Buffer	▨
~6200_8	T.LIB	Library Directory	

▨ NEW OR RELOCATED FOR SCOPE 3.3

* T.APF cannot start after 100008
T.RQS cannot start after 200008

FIG 3. SCOPE 3.3 CMR

SCOPE 3.3

P.ZERO	ZEROS					0
P.LIB	C.DIRFWA 0.0	FWA LIB DIRECTORY		LWA+1 LIB DIRECTORY	DEAD START LOAD FLAG	1
P.RBR P.RBT	C.RBRAD FWA OF RBR AREA		RBT ORDINAL OF EMPTY CHAIN	LENGTH/100B OF RBT AREA	(LWA+1)/100B OF CENTRAL MEMORY	2
P.NPP/P.NCP P.DFB/P.NOVA	FWA/8 OF DAYFILE BUFFER	C.NOVA (CP250) T.NOVA/8	C.NOVAL L.NOVA	▨ C.NPP NO. OF PPU'S	▨ C.NCP NO. OF CTL PTS	3
P.SEQ/P.FNT P.HEC	FWA OF FNT/FST	LWA+1 OF FNT	C.SEQ T.SEQ/8	C.SEQL L.SEQ	C.HEC HARDWARE ERR CT	4
P.CST P.EST	FWA OF EST	LWA+1 OF EST	C.CST FWA OF CST	C.CSTL LWA+1 OF CST	▨ C.CHRQ CHAN REQ. TABLE	5
P.PFM1	C.SDTL N.SD	C.APFL L.APF	C.SDT FWA OF SDT	C.APF FWA OF APF	C.PFMCH PF INTERLOCKS	6
P.PFM2	C.SDL N.ESD	C.RBTCL N.RBTC	C.RBTC1 C.RBTC2 C.RBTC3	POINTER TO RBTC CURRENT EOI		7
P.INS	RESERVED FOR INSTALLATIONS					10
P.ECST	C.ICEBUF T.ICEBUF/8	C.ECSTAT T.ECSTAT/8	C.ECFLAW T.ECFLAW/8	C.LRD T.LRD/8	C.ECST T.ECSTAT/8	11
P.LECST	C.ICEBUF L.ICEBUF	C.ECSTAT L.ECSTAT	C.ECFLAW L.ECFLAW	C.LRD L.LRD	C.ECST L.ECST	12
P.RQS	▨ T.DAT	▨ L.DAT	▨ C.RQSFS FWA/2 REQ STACK	▨ NUMBER OF DST ENTRIES	FWA/8 OF DST	13
P.TAPES P.RMS	▨ T.TAPES/8	▨ L.TAPES	▨ T.RMS/8	▨ L.RMS		14
P.AVTAPE	▨ C.AVMT	▨ C.TMT	▨ C.AVNT	▨ C.TNT		15
P.INT	▨ C.INT/C.IFL	▨ C.ITABL	▨ C.IDED C.IBUFF	▨	▨ C.IRES	16
						17
			STORAGE MOVE FLAG		MACHINE FL/100B	20
	ΔSYSTEMΔΔΔΔ					21
					C.CPECFL ECS FL/1000B	22
	77770		CPU A IDLE TIME	SECONDS	MILLISECONDS	23
			CPU B IDLE TIME	SECONDS	MILLISECONDS	24
			ECS TIME	SECONDS	MILLISECONDS	25
T.CPJOBN	JOB SEQUENCE NUMBER		JOB COUNT			26
T.JDATE	(LEADING ZEROS) YYDDD					27
T.CLK	HH . MM . SS					30
T.SLAB1 T.DATE	MM / DD / YY					31
T.SLAB2 T.SLAB6	SCOPE VERSION 3.X					32 36
T.MSP	RESERVED FOR CDC			DEBUGGER	STEP FLAG	37
T.MSC	COUNT OF PP JOB QUEUE ENTRIES	SECONDS	MILLISECONDS	MILLISECOND CLOCK		40
T.PPS1 T.PPSO	D.CPAD	D.PPSTAT	▨ R.FAF	▨ IO CHANNEL TIME	▨	41 52
	CDC					53
	CDC					54
T.RCHN	SPM-RCN COMMUNICATION WORD				1ST RBT WORD TO BE RELEASED	55
T.CPT1/T.UAS T.STATCP	UNASSIGNED CM/100B	UNASSIGNED ECS/1000B	C.STATCP	CPU B STATUS	CPU A STATUS	56
T.ECSPAR			ECS FLAW TABLE FLAG	ECS PARITY FLAG	ECS PARITY ADDRESS/1000B	57

Fig. 6. CMR POINTER AREA

▨ New or Relocated for SCOPE 3.3

SCOPE
3.3

W.PPIR	↑C.CPNAM PROGRAM NAME OR 0	0	C P .		0
W.PPOR					1
W.PPMES 1					2
W.PPMES 2					3
W.PPMES 3					4
W.PPMES 4					5
W.PPMES 5					6
W.PPMES 6					7

~~FIG.~~ PP COMMUNICATIONS AREA

SCOPE 3.3

	EXCHANGE PACKAGE	0 17
W.CPSTAT (NOTE 1)	C.CPSTAT (NOTE 2) W X A J Y C.CPEF ERROR FLAG C.CPSM STORAGE MOVE FL C.CPRA RA/100B CM C.CPFL FL/100B CM	20
W.CPJNAM	JOB NAME C.CPNCSP NEXT CTL. STMT.	21
W.CPPRI W.CPECS	C.CPPRI PRIORITY C.CPECRA ECS RA/1000B C.CPECFL ECS FL/1000B	22
W.CPTIML W.CPTIME	C.CPTIML TIME LIMIT SECONDS CPUA TIME SECONDS MILLISECONDS	23
W.CPTIMB	CPUB TIME SECONDS MILLISECONDS	24
W.PPTIME	PPU TIME SECONDS MILLISECONDS	25
W.SSW	SENSE SWITCHES	26
W.EQP	CDC Reserved	27
W.CPDFM	LAST DAYFILE MESSAGE (1-5 WORD LINE) (1-3 WORD LINE) (WORD 37 USED BY INTERCOM AS LAST WORD OF FL)	30
W.CKP W.CPERT	C.CPFST FST ADDRESS C.CFFP d e k u p n l o g f S.CPE S.CPR S.CPEOR S.CPOP S.CPX S.CPN S.CPS S.CPA S.CPG S.CPF C.CKP # OF CKPTS	40
W.CPJCP	C.CPTLI INITIAL TIME LIMIT C.CPECSI INITIAL ECS FL C.CPFLI INITIAL FL	41
W.CPOPV	C.CPOPV	42
W.CPCC	C.CPRPV C.CPRPA C.CPLIB	43
W.CPTAPE	C.CPTTPS MT NT C.CPATPS MT NT C.CPRTPS MT NT	44
W.ID (WD 50)	OWNER ID	50
W.CPCAF	CONTROL CARD BUFFER	51
W.CPCAL		150
W.FSTCC	FST ENTRY FOR NEXT CONTROL CARD PRU	151
W.CPRO W.CPLDR	C.CPSO C.CPLDR2 LOADER FLAG C.CPLDR3 w s t c l r m C.CPRO ROLLED OUT FLAG	152
W.CPOAE (NOTE 3)	C.CPREQ REQ FLAG C.CPFLAG C.CPRPRI RERUN PRIORITY C.CPOUT JANUS FLAG C.CPOAE EQUIP. ASSIGNED	153
W.CPVRNO	PRIVATE PACK ASSIGNMENT	154
W.CPDFMC	C.CPDFMC DAYFILE MSG COUNT	155
W.CPRES1	RESERVED FOR E/I 200-SENTRY	156
W.CPAR	Last Auto-Recall Request	157
W.CPPF1	RET. ADDR. SUBD 0 CYCLE 0 PERM 0 APF ← PFD POINTER →	160
W.CPPF2	RBTA RBTO PRU ENTRY MODE ABCO ← PFD POINTER →	161
W.CPPF3	RBTA RBTO PRU ENTRY EF	162
W.CPMSLM	MASS STORAGE LIMIT IN PRU'S RUNNING PRU COUNT	163
W.CHTIM	SECONDS MILLISECONDS	164
W.CPIN	C.ICPST/C.INT STATUS C.ICPUTP USER TABLE ADDRESS C.ICPUID USER ID C.ICPFST ADDR. OF SWAP FST	165
W.CPNS	INTERCOM PP RECALL REGISTER	166
W.CPSR	C.CPSR	67
W.CPINS	RESERVED FOR INSTALLATIONS	170 177

☐ NEW OR RELOCATED FOR SCOPE 3.3

CONTROL POINT AREA

SCOPE
3.3
&
3.4

0		P	A0	B0	0
1		CMRA	A1	B1	1
2		CMFL	A2	B2	2
3		EM	A3	B3	3
4	ECS RA		A4	B4	4
5	ECS FL		A5	B5	5
6		MA	A6	B6	6
7			A7	B7	7
10	X0				8
11	X1				9
12	X2				10
13	X3				11
14	X4				12
15	X5				13
16	X6				14
17	X7				15

FIG. 8 EXCHANGE PACKAGE

NOTES: CONTROL POINT AREA CHART

SCOPE
3.3

a	Rerun
b	Rerun priority
c	Exchange dump
d	EXPORT/IMPORT
e	INTERCOM
f	Reprocess
g	Abort bit for exit
h	Clear flag
i	Sequencer flag
l	Labeled dump
m	00-no map, 01-full map, 10-partial map
n	CHECKPOINT has been taken
p	EXIT card encountered
q	Control card end of record
r	REDUCE flag
s	SNAP
t	TRACE
u	Private disk pack flag
w	LOADER 00-PP LOADER, 01-CP LOADER
oq	Private pack overflow

Note 1: also W.CPEF, W.CPSM, W.CPFL
Note 2: W-wait, X-recall, A-requires CPU A, B-requires CPU B, Y-auto-recall, M-storage move
Note 3: also W.CPOUT, W.CPFLAG

Reference Words W.CPPF2 and W.CPPF3

A	HICY
B	INCOM
C	NEWNAME
D	PP COMP. FLAG
E	WAIT
F	PRIORITY

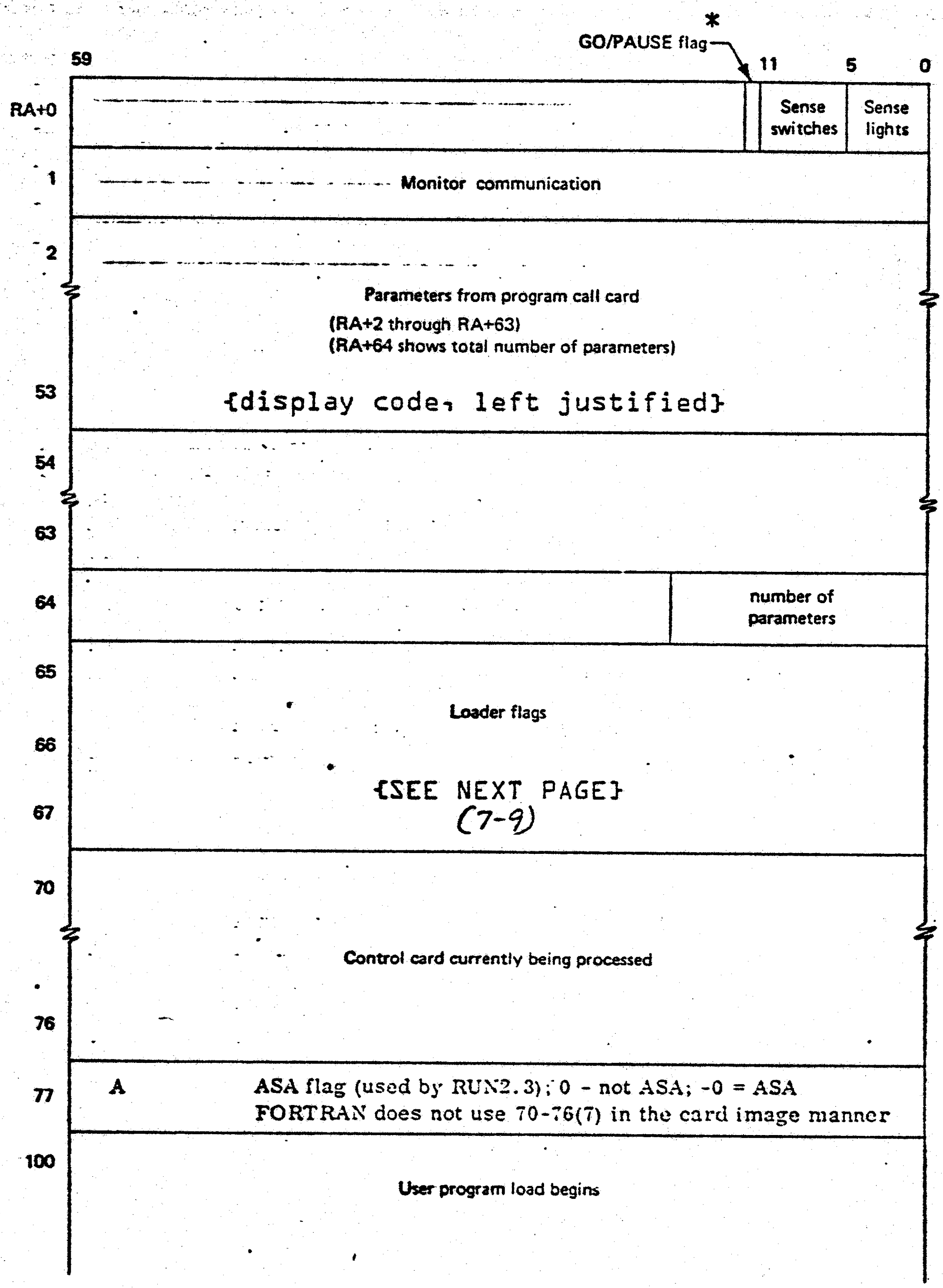
CONTROL POINT AREA ERROR FLAG VALUES {3.3}

SCOPE
3.3

1	F.ERTL	TIME LIMIT
2	F.ERAR	ARITHMETIC ERROR
3	F.ERPP	PP ABORT {M.ABORT}
4	F.ERCP	CP ABORT {ABT IN RA+1}
5	F.ERFCE	PP CALL ERROR
6	F.EROD	OPERATOR DROP
7	F.ERK	KILL
10	F.ERRN	RERUN
11	F.EREX, F.ERCC	CONTROL CARD ERROR
12	F.ERECP	ECS PARITY ERROR
13	F.ERJC, F.ICCD	JOB CARD ERROR
14	F.ERPA	PRE-ABORT
15	F.ERRCL	AUTO RECALL ERROR
16	F.ERHANG	JOB HUNG IN AUTO RECALL
17	F.ERMSL	MASS STORATE LIMIT EXCEEDED
20	F.IUABT	USER ABORT
21	F.IHEAD	REQUEST FOR HEADER LINE TOO
22		LONG. ROLLOUT RECALL ERROR
23		NOT AUTHORIZED TO USE
24		PROG. DO NOT UNDERSTAND
25		CONTROL CARD
26		ERROR.
27		LOADER ERROR
30	F.IMYGE	MYSTERY GUEST LOGIN
31		
32		OUT OF TIME
33	F.ITL	TIME EXCEEDS AUTHORIZATION
34	F.IFL	F.L. EXCEEDS AUTHORIZATION
35		
36		
37		
40	F.IOUT	OUTPUT FILE ERROR
41	F.IPNRD	PRINTER NOT READY
42	F.ISYS	SYSTEM ERROR
43	F.ICRD	CARD READ ERROR
44	F.IJBCRD	JOB CARD ERROR
45	F.IINP	INPUT FILE ERROR
46	F.IFMT	FORMAT ERROR {READER NOT READY}
47	F.IFNF	FILE NOT FOUND {ACCOUNTING HEADER}
50	F.IDSPF	ILLEGAL DISPOSITION FOR PERMANENT FILE
51	F.ILLFN	ILLEGAL FILE NAME
52		FILE QUOTA EXCEEDED

SCOPE 3.3

CONTENTS OF RA+0---RA+77 IN USER's FIELD



GO/PAUSE flag; 0 = GO, 1 = Pause, wait for GO

* GO bit can be used effectively for operator communication. Set G {bit 12} in the CP program. It will be cleared when the operator hits GO. A CP or PP program can, ie, loop in periodic recall waiting for the bit to be cleared.

2/12

LOADER FLAGS

(Detail from User's Field, P. 7-8)

<table>
<tr><td>+63</td><td colspan="12"></td><td>17</td></tr>
<tr><td>+64</td><td colspan="12">Program or File Name</td><td>No. of Parameters</td></tr>
<tr><td>+65</td><td></td><td>Segment Table Pointer</td><td colspan="10">Reserved</td><td>Next Available CM</td></tr>
<tr><td>+66</td><td></td><td>FWA Loader Tables</td><td>MP</td><td>O</td><td>S</td><td>T</td><td>C</td><td>L</td><td colspan="3">OVLVL</td><td>2d</td><td>FWA of User Object Program</td></tr>
<tr><td>+67</td><td></td><td>S.OFLAG</td><td colspan="2"></td><td>P</td><td>E</td><td>Q</td><td>M</td><td>R</td><td>CT</td><td colspan="2">2221</td><td>FWA Loader</td></tr>
<tr><td>+70</td><td>59</td><td>53</td><td>35</td><td></td><td>31</td><td></td><td>29</td><td></td><td>27</td><td></td><td>23</td><td></td><td>17 00</td></tr>
</table>

In the above diagram RA+64 through RA+67 are reserved for the SCOPE system-loader interface and bytes are assigned as follows:

2d	Indicator for loader directive
OVLVL	Level of incoming overlay
CT	Control card type NOGO, LOAD, EXECUTE, PROGRAM
R	RSS mode indicator
M	No map flag for library load
Q	Request flag - communication between LDR and LOADER
E	End of load flag
S.OFLAG	Segment - FWA of tables for lowest segment in user's job area
	Overlay - FWA blank common
P	Partial map flag
S	SNAP
T	TRACE
C	Change dump
L	Labeled dump
O	REDUCE (OBESE) flag
MP	Map flag: 001 = Full map, 010 = No map, 100 = Partial map
Card image	Upon initial entry from a named routine call or an EXECUTE card, these locations will contain the card image (in display code) of the card which called for execution.

CENTRAL MEMORY RESIDENT

SCOPE 3.4

0		Pointers
100		Channel Status Table
154		PP Status Words
200	T.CPA$_n$	Control Point Areas
	T.XPIOLA	System Exchange Packages
	T.PPC$_n$	PP Communication Areas
*	T.EST	Equipment Status Table
*	T.FNT	File Name Table
		CIO-CPCIO Special FNTs
		Permanent File FNTs
*	T.ITABL	INTERCOM Table
*	T.DAT	Device Activity Table
*	T.STG	Tape Staging Table
*	T.APF	Attached Permanent File Table
‡	T.RQS	Request Stack
	T.RBR	Record Block Reservation Table (Headers)
	T.RBRBIT	RBR Bit Table
	T.DST	Device Status Table
	T.DPT	Device Pool Table
	T.SEQ	Sequencer Table
	T.RMS	Rotating Mass Storage Diagnostic Table
	T.INS	Installation Area
	T.VSNBUF	VSN Buffer
	T.TAPES	Tapes Table
	T.RPT	Removable Pack Table
	T.MAIL	Scheduler Mailbox Buffer
	T.DFB	Dayfile Buffers
	T.PJT	Parameter Storage for Delayed PP Jobs
	T.SCHPT	(Optional) Scheduler Statistics
	T.SCHJCA	Scheduler Job Control Area
	T.SCHJDT	Scheduler Job Descriptor Table
	T.BCFAP	CPMTR CEFAP Buffer
	T.EPAGE	Empty Page Stack
	T.ECSPRM	ECS Parameters
	T.SUBPG	Subpage Buffer
	T.ECTL	Description of T.EBUF Area
	T.EBUF	ECS Buffer for RMS—ECS Transfer
		CM Resident Programs
	T.LIB	Library Directory
		INTERCOM Pointer Area
		INTERCOM Small Buffers and User Tables
		INTERCOM Large Buffers

*Table Must Begin Before 10000_8

‡Table Must Begin Before 20000_8

▨ New or Relocated for SCOPE 3.4

SCOPE 3.4

CMR TABLE FORMATS

CMR tables are defined as follows:

The CMR Pointer Area which contains 100 words of various flags and pointers to CM tables, many of which are new.

The Channel Status Table which contains one word entry per channel (hardware or pseudo) in the system, has been lengthened to accomodate the new hardware and software channels for SCOPE 3.4 and INTERCOM. The table length is now 44_8 words.

The PP Status Words have been removed from the pointer area and now reside beginning at location 154_8. These words are defined in the same manner as SCOPE 3.3, i.e., one word per PP defined in the system.

The Control Point Areas are the only areas remaining in the same location as in SCOPE 3.3. There are now fifteen possible areas of 200_8 words, one for each control point. Each area contains the exchange package, job name and information about the job running at the control point.

System Job Exchange Package Area is the exchange package and register area for all Central Processor CM resident systems programs which run in user mode. There will also be one or two (depending upon number of CPUs) idle exchange package areas.

The PP Communications Area contains up to twenty 8-word areas, one for each PP defined in the system. The format is unchanged from SCOPE 3.3 in its function as communication area for the PPs and Monitor.

The Equipment Status Table is comprised of a one word entry for each device attached to the system. The format of the EST has been changed to define each entry type of device rather than allocation.

The File Name Table has one entry for each file in the system, excluding unattached permanent files. These entries may be three or six words in length.

Immediately following the FNT are the special CIO-CPCIO FNTs as well as the Permanent File FNTs. These entries are mentioned separately since they are not included in the length pointers for the FNT; and they are referenced outside the routine system.

The INTERCOM Table, increased to two words, provides software information to INTERCOM. The multiplexor subtables have also been modified, especially to include the new hardware features.

SCOPE 3.4

The Device Activity Table contains one-word entries for each mass storage device in the system. In addition to recording device activity, a reservation has been made for dual access.

The Tapes Staging Table has four words of information needed to stage tapes. This includes number of units defined, number of units unassigned, units being held up and units needed.

The Attached Permanent File Table is made up of a two-word entry for each permanent file attached to a control point. These words contain status information, PFD pointer, etc.

The Request Stack holds requests for I/O on a mass storage file. Each request entry is not three words long with various pointer and operation code information.

The Record Block Reservation Table {Headers} has two parts to describe; the headers and the RBR Bit Table. The headers in the RBR table are two-word entries each associated with an RBR Bit Table and with an RMS device. The headers contain device and record block information.

The RBR Bit Table contains bit representation of record blocks on a device. The setting of the bit indicates non-availability.

The Device Status Table, as before, contains two-word entries; one for 1SS and one for each rotating mass storage controller in the system.

The Device Pool Table is a table of internal information used by 1EP and 1SP. There is an eight word entry for each RMS device in the system.

The Program Sequencer Table contains two-word entries for each job running under the control of the program sequencer. The first two-word entry is for the Automatic Program Sequencer.

The Rotating Mass Storage Diagnostic Table contains one 30-bit entry for each RMS device that was preallocated for C. E. Diagnostics programs at deadstart time.

The Installation Area is reserved for just that.

The VSN Buffer is used for tape staging via the P display. Job information is recorded in six-word blocks.

The Tapes Table has eight-word entries, one for each tape unit in the system. This area supplies label and status information to insure sufficient tapes for the configurations.

The Removable Pack Table contains status and label information on disk packs which have been designated as sequential packs. Each entry consists of two words.

SCOPE 3 4

The Scheduler Mailbox Buffer is a variable length buffer used to hold dayfile messages for swapped=out jobs.

The Dayfile Buffer Area contains a pointer word (FET) and a buffer area for each control point in the system, the system dayfile, and the C. E. error file.

The Peripheral Job Table is used by PP monitor and consists of a four-word entry for each job entered in the Peripheral Job Queue, the Delay Stack or the Event Stack.

The Scheduler Statistics Table is an optional table containing performance testing information on the Scheduler.

The Scheduler Job Control Area is used by the Scheduler to acquire job class information. Each entry is two words in length. There is an INPUT queue entry as well as entries from five possible classes.

The Job Descriptor Table is another area used by Scheduler. Each job to be executed will have a five-word entry in the table containing status information needed for job scheduling. This information includes field length, priority, RBT information, class, etc.

The CPMTR-CEFAP Buffer is used to communicate between CEM and the CM system routines, mostly ECS. When CPMTR or any other such program detects an ECS or DDP parity error on a read or write the information is put into the buffer and picked up by CEM for the C. E. error file.

The Empty Page Stack in CMR refers to a table used for indexing available ECS pages. The table reflects the empty page stack in the ECS system area.

The ECS Parameter Table contains parameters that would normally be in the CMR pointer area. But space would not allow this. Therefore, the parameters are in the ECS table area.

The Subpage Buffer is an area used to hold a files current subpage, therefore making it available to I/O buffering routines in CM. Each file buffered through ECS will have such system subpages.

The RMS-ECS Transfer table describes the RMS-ECS transfer buffer which follows it. The firstword contains total and available buffers. Then there are two entries for each buffered device with first word address information.

The RMS-ECS Transfer Buffer is the system double buffer for ECS. It is used by LEP and CM resident ECS drivers for transfers between the two storages.

SCOPE 3.4

The CM Resident Programs are those programs such as SPM, CPMTR, etc., which are required to be CM resident. At present, they are:

CP.MTR	Central Processor Monitor
CP.SM	Central Memory Storage Move
CP.ECSM	ECS Storage Move {optional assembly IP.MECS}
CP.SPM	Stack Processor Manager
CP.SCH	Memory Manager
CP.SCH1	Integrated Scheduler
CP.ECOVL	ECS Read/Write PP Overlay
CP.CIO	Central Processor I/O Controller

The Library Directory consists of the Library Name Table, the PP Program Name Table, the PP Program bodies, and finally the CM resident libraries.

The INTERCOM Pointer Area contains some twenty-seven words of pointers to buffers, tables, and interlocks used by INTERCOM.

The Low Speed User Table is used by INTERCOM for CRT and TTY user information such as buffer pointers, FNT pointers, field length, time, etc.

The MUJ Table contains some of the same information as is in the User Tables, such as, ID, pointers, 1CI timer, etc.

The High Speed User Table is analogous to the Low Speed User Table except that it is used for high speed import devices. It contains data pointers, buffer pointers and types, etc.

The Auxiliary High Speed User Table is an additional table which is used by each active HS terminal to accomodate the data streams which need extra data and information space

The 274 IGS User Table is a table assigned to a graphics job by routine GBJ. This table holds information such as data pointers, job class, and descriptor information.

SCOPE 3.4

CMR POINTER AREA

	59	47	35	23	11 0	
P.ZERO	Zeros					0
P.LIB	a C.DIRFWA FWA of Library Directory			LWA+1 Library Directory	C.DSFLAG Deadstart Load Flag	1
P.RBR P.RBT	C.RBRAD FWA of RBR Area		RBT Ordinal of Empty Chain	Length/100B of RBT Area	C.CMLWA (LWA+1)/100B of CM	2
P.NPP P.NCP P.DFB	FWA/8 of Dayfile Buffer	(Reserved for 250 Graphics Package)		C.NPP No. of PPs	C.NCP No. of CPs	3
P.SEQ P.FNT P.HEC	FWA of FNT	LWA+1 of FNT	C.SEQ T.SEQ/8	C.SEQL L.SEQ	C.HEC Hardware Error Count	4
P.CST P.PCOM P.EST	FWA of EST	LWA+1 of EST	C.CST FWA of CST	C.CSTL LWA+1 of CST	C.PCOM Address of Comm Area PP1	5
P.PFM1	C.SDTL N.SD	C.APFL N.EAPF	C.PFACT Activity Count C.APF T.APF		C.PFMCH Toggle Byte	6
P.PFM2	C.SDL C.ESD					7
P.INS	(Reserved for Installations)					10
P.EIRPR	C.LEPAGE L.ECSTK+1		C.ECSPRM T.ECSPRM	ICC Area Address		11
P.ELBST	Max. Length/1000B of ECS Library File	ECS Flaw Table Address		ECS Page Stack Address		12
P.RQS	T.DAT	L.DAT	C.RQSFS FWA/2 of Request Stack	No. of DST Entries	FWA/8 of DST	13
P.DPT P.TAPES P.RMS	T.TAPES/8	L.TAPES	C.RMS T.RMS/8	C.RMSL L.RMS	C.DPT T.DPT/8	14
P.STG					C.STG T.STG	15
P.INT	C.INT/C.IFL Control Point 0 FL	C.ITABL	C.IBUFF Start of INTERCOM pointer area		C.ILTABL	16
P.PFM3	C.RBTC1 ← RBTC Pointer →	C.RBTC2	C.RBTC3	C.RBTCL N.RBTC	C.PFFNT FWA of PF FNTs	17

▨ New or Relocated for SCOPE 3.4

SCOPE 3.4

CMR POINTER AREA

	59 47 41 35 23 11 0	
T.JDATE	(Leading Zeros) Y Y D D D	20
P.NRBR	Number of Request Stack Entries (N.RQS) Number of RBR Headers (N.RBR) Size of Total RBR Area (L.RBR)	21
T.BJDT	Julian Date in Binary (YYYDDD) Time in Binary (HHMMSS)	22
P.EVICT	Trace Buffer T.TRB/8	23
P.CMFL	Machine FL/100B	24
	∧ S Y S T E M ∧ ∧ ∧	25
T.CPJOBN P.PJT P.SPDROP	Job Sequence Number C.SPDROP DST Ordinal for 1SP Drop Job Count C.PJTFWA T.PJT/8 C.PJTLWA T.PJT/8+L.PJT/8	26
T.EPBL P.ECSFL	C.ECSPL ECS Page Length C.ECSBL ECS Buffer Length C.CPECFL Direct Access ECS FL/1000B	27
T.CLK	H H . M M . S S	30
T.SLAB1 T.DATE	M M / D D / Y Y	31
T.SLAB2	System Label SCOPE Version 3.4	32
		33
		34
		35
T.SLAB6		36
T.MSP	Debugger Step Flag	37

■ New or Relocated for SCOPE 3.4

SCOPE 3.4

CMR POINTER AREA

	59	47	35	23	11	0	
T.MSC	Count of PP Job Queue Entries	Number of Idle PPs	Number of Seconds*4096				40
P.CHRQ				C.CHRQ First 10 Channels	C.CHRQ2 Second 10 Channels		41
P.PPLIB	Position of CIO	0 0 0 0	Number of Programs		Address of First Entry		42
P.VRNBUF	C.VRNFWA T.VRNBUF/8	C.VRNFIN Pointer to First VSN	C.STGFLG Stage ON/OFF	C.VRNINT Buffer Interlock	C.VRNFUL Buffer Full Flag		43
T.CPSTA	Idle Exchange Package Address	* * * * * * * *	Next Slice Time	2 0 0 3 0 3 * * * * 0 0 0 L * * * *	Active XP Address		Rescheduling 44 OFF
T.CPSTB							45
T.MXNCTL	0 0 0 0 0 0 0 0 0 STL Code	0 0 0 0 0 0 0 0 0	0 0 0 0 20	Active XP Address * * * * * *	0 0 P 2 6 1 P 2 6 2 P		EXN MXN 46 MAN
T.PPID	CP-MTR Requests				PP Input Register Address		47
T.PPIP	PP-MTR Requests				PP Input Register Address		50
	(Reserved)						51
	(Reserved)						52
T.SPF	Control Point Number				EST Ordinal		53
	(Reserved)						54
T.RCHN	SPM-1RN Communications Word				First RBT Word Pair to Release		55
T.CPT1 T.UAS	Unassigned CM/100B	Unassigned ECS/1000B	ECS Size		Initial CMTR P Address		56
T.ECSPAR P.EPAGE		C.EPAGE T.EPAGE	ECS Flaw Table Flag	ECS Parity Flag	ECS Parity Address/1000B		57

L = { 0 Turned Off
      1 Locked Off

P = { 0 CPUA
      1 CPUB

▨ New or Relocated for SCOPE 3.4

SCOPE 3.4

CMR POINTER AREA

	59	47	35	23	11 0	
P.SCH	C.SRSL C.LEJDT LE.JDT	C.SRS T.XPSCH/10B	C.JCA T.SCHJCA/8	C.LJDT L.SCHJDT	C.JDT T.SCHJDT/8	60
P.STR	C.NFL Needed FL/100B	C.JQP Queue Priority of Job in Counter	C.RFL Reserved FL/100B	C.STMF SCH Recall	C.AFL Available FL/100B	61
T.SCHCP	Interlock Word (Scheduler)					62
T.SCHPP	Interlock Word (PP Routines)					63
P.RPT	T.RPT/8	L.RPT				64
P.MAIL **P.SWPECS** **P.SCHPT**	C.MAILF T.MAIL/8	C.MAILL L.MAIL	C.SWPECS L.ECSSWP	C.SCHPT T.SCHPT/8		65
	(Reserved)					66
	(Reserved)					76
P.ILR **P.PPOVL**			C.ILR IP.ILR	C.PPOVL T.PPOVL/10B	T.ELIBD	77

IP.ILR:
0 = No ILR
1 = 64-bit ILR
2 = 128-bit ILR

▨ New or Relocated for SCOPE 3.4

SCOPE 3.4

CONTROL POINT AREA

59 47 44 41 35 29 23 17 14 11 5 0

	Exchange Package					0 17
W.CPUST W.CPLINK	C.CPSTAT Status Byte	C.CPSLIC M.RCLCP Time or Slice Period	* *	C.CPLINK Previous Active Control Point	Next Active Control Point	20
W.CPTIME	C.CPUQS CPU-A	C.CPUQMS Seconds*4096 This Quantum	C.CPUAS Total CPU-A Time as Number of Seconds*4096			21
W.CPTIMB	C.CPUQS	C.CPUQMS CPU-B Seconds*4096 This Quantum	C.CPUBS Total CPU-B Time as Number of Seconds*4096			22
W.PPTIME W.CPPTM	C.CPPQS	C.CPPQMS PP Seconds*4096 This Quantum	C.CPPTS Total PP Time as Number of Seconds*4096			23
W.CPSTAT W.CPFL W.CPEF		C.CPEF Error Flag	C.CPSM Storage Move	C.CPRA RA/100B	C.CPFL FL/100B	24
W.CPJNAM	C.CPJNAM Job Name			JDT Ordinal		25
W.CPCC	C.CPRPV Reprieve CKSM Value	C.CPRPA Reprieve Address	C.CPNFL Nominal FL/100$_8$	C.CPNCSP Next Control Card Pointer		26
W.CPECS		C.CPECRA ECS RA/1000B	C.CPECFL ECS FL/1000B			27
W.CPDFM	Last Dayfile Message					30 37
W.CPPRI W.CPJCP W.CPTIML	C.CPTIML Current Time Limit (15 Bits)	C.CPTLI Initial Time Limit (15 Bits)	C.CPRPRI Job Class	C.CPECSI Initial ECS FL/1000B	C.CPFLI Initial FL/100B	40
W.CPSWP W.CPINT	C.CPQNT Quantum	C.CPUTA User Table Address	C.CPORG C.CPEVNT Flags	Job Origin		41
W.CPSCH W.CPRO	C.CPFLG Swap Flags	C.CPJQP Job Queue Priority	C.CPRFL Reserved FL		C.CPJDA JDT Address (Absolute)	42
W.SSW W.CPSSW		C.CPSSW Sense Switches				43
	(Reserved)					44
	(Reserved)					45
	(Reserved)					46
	(Reserved)					47

▨ New or Relocated for SCOPE 3.4

SCOPE 3.4

CONTROL POINT AREA

	59	47	42	35	23	17	11	5	
W.CPFACT	Account Parameter for Permanent Files								50
W.CPFST	FST Entry for Next Control Card PRU								51
W.CKP W.CPCKP					C.CPCON Console Checkpoint Flag		C.CPCKP Number of Checkpoints		52
W.CPOAE	C.CPREQ Req Flag	1 = Extended Label Format	Relative Address of Tape Label Info				C.CPOAE Equipment Assigned		53
W.CPVRNO	Family Pack VID Assignment								54
W.CPLDR1	C.CPLW C.CPLT Loader Flags				Global Library				55
W.CPLDR2 W.CPLS					Set				56
W.CPLDR3					Indicators				57
W.CPAR	RA+1 Contents (and Control Point Number) of Last Auto-recall Request					C.CPAR Reply Word Address			60
W.CPSTG	C.CPTMT MT	C.CPTNT NT					C.CPMNT MT Max	NT Max	61
W.CPDFMC W.CPDPV W.CPIRB	C.CPDFMC Dayfile Msg Count			C.CPRBID INTERCOM Batch Routing ID			C.CPDPV Job Dep. ID		62
W.CPFP W.CPOUT W.CPFLAG W.CPERT	C.CPFLAG Flags			C.CPFST FST Address			C.CPFP C.CPOUT Flags		63
W.CPMSLM	C.CPMSLO MS Limit Saved During Swap		C.CPMSLM MS Limit in PRU's		C.CPMSRC	Running PRU Count			64
W.CHTIM				Channel Time as Number of Seconds*4096					65
W.CPMSI	C.CPSITM Time of Swap-In								66
W.CPSR		C.CPSR					C.CPESR (Non-zero During ECS-Disc Transfer)		67
W.CPCAF W.CPCAL	Control Card Buffer								70 167
	Reserved for Installations								170 177

▇ New or Relocated for SCOPE 3.4

118

SCOPE 3.4

Control Point Area

W.CPSLIC
W.CPLINK

Word 20
Byte 0 {C.CPSTAT} - Status Byte bit values are as follows:

Bit 0	S.CPUSTM	Move flag - move in progress
1	S.CPUSTY	Auto Recall
2	S.CPUSTA	CPUA assigned only
3	S.CPUSTB	CPUB assigned only
4	S.CPUSTX	Recall status
5	S.CPUSTW	Wait status
6	S.CPUSTR	Real time job
7	S.CPUSTC	Active CPUA
8	S.CPUSTD	Active CPUB
9	S.CPUSTS	Control point activity suspended
10	S.CPUSTP	Suspended by check point

Bytes 1-2 {C.CPSLIC} - Total time slice allotted to the CPU.
Bytes 3-4 {C.CPLINK} - Link to other active control points.

W.CPTIME

Word 21
Bytes 0-1 {C.CPUQS/C.CPUQMS} - CPU A quantum in seconds *4096
Bytes 2-4 {C.CPUAS} - CPU A time as number of seconds *4096

W.CPTIMB

Word 22
Byte 0-1 {C.CPUQS/C.CPUQMS} - CPU B quantum in seconds *4096
Bytes 2-4 {C.CPUBS} - CPU B time as number of seconds *4096

W.PPTIME

Word 23
Bytes 0-1 {C.CPPQS/C.CPPQMS} - PP quantum in PP seconds *4096
Bytes 2-4 {C.PPIS} - Total PP time in seconds *4096

W.CPSTAT
W.CPFL

Word 24
Byte 1 {C.CPEF} - Error flag values are as follows:

W.CPEF C.CPEF Values:

0001	F.ERTL	Time limit exceeded
0002	F.ERAR	Arithmetic error
0003	F.ERPP	PPU abort {M.ABORT}
0004	F.ERCP	CPU abort {ABT in RA+1}
0005	F.ERPCE	PP call error {garbage in RA+1}
0006	F.EROD	Operator drop
	F.IUABT	INTERCOM user abort
0007	F.ERK	Operator kill {batch job only}
0010	F.ERRN	Rerun {batch job only}
0011	F.EREX	Control card error
	F.ERCC	
0012	F.ERECP	ECS parity error
0013	F.ERJC	Job card error
	F.ICCD	
	F.IJBCRD	
0014	F.ERPA	Pre-abort {batch job only}
0015	F.ERRCL	Auto-recall error
0016	F.ERHANG	Job hung in auto-recall
0017	F.ERMSL	Mass storage limit exceeded {batch job only}
0020	F.EROVL	PP overlay not in PP LIB

Byte 2 {C.CPSM} - Storage move flag set to non- zero by MTR is a storage move requested.
Byte 3 {C.CPRA} - Control point RA/100B
Byte 4 {C.CPFL} - Control point field length/100B

W.CPJNAM Word 25
Bytes 0-3 {C.CPJNAM} - Job name {7 characters}
Byte 4 contains the Job Descriptor Table Ordinal.

W.CPCC Word 26
Byte 0 {C.CPRPV} - Reprieve {RPV} checksum value
Bytes 1-2 {C.CPRPA} - Reprieve {RPV} FWA
Byte 3 {C.CPNFL} - nominal FL/100B
Byte 4 {C.CPNCSP} - pointer to next control card

W.CPECS Word 27
Byte 3 {C.CPECRA} - RA/1000B of ECS
Byte 4 {C.CPECFL} - FL/1000B of ECS

W.CPDFM Word 30 through 37
Last day file message. Words 30-34 contain the first display line. Words 35-36 contain the second display line. Word 37 is the last word of FL {INTERCOM}.

W.CPPRI
W.CPJCP
W.CPTIML

Word 40
Byte 0 {C.CPTIML} - Current time limit
Byte 1 {C.CPTLI} - Initial time limit
Byte 2 {C.CPPIR} - Job Class {Priority}
Bytes 2-3 {C.CPECSI} - Initial ECS FL/1000B
Byte 4 {C.CPFLI} - Initial FL/100B

W.CPSWP
W.CPINT

Word 41
Bytes 0-1 {C.CPQNT} - Quantum Value
Bytes 2-3 {C.CPUTA} - User Table Address {INTERCOM}
Byte 4 {C.CPORG} - Job Origin

C.CPORG Values {octal}:

4	Real time
10	Graphics
20	Multi-user
40	INTERCOM
Bit 6	Swap Out Event Bit

W.CPSCH
W.CPRO

Word 42
Byte 0 {C.CPFLG} - Swap flag

C.CPFLG Values:

Bit 0	Unused	
1	Unused	
2	Unused	
3	S.CPLIB	LIB bit
4	S.CPFFL	FNTs in positive FL
5	S.CPEOJ	End of job
6	S.CPCLR	Control point area clear reques
7	S.CPRFL	Storage request
8	S.CPROP	Roll out
9	S.CPSIP	Swap in
10	S.CPSOP	Swap out
11	S.CPSWC	Swap out complete

Byte 1 {C.CPJQP} - Job Queue Priority
Byte 2 {C.CPRFL} - Reserved field length
Byte 3 {C.CPJCA} - Job class {JCA} index
Bytes 3-4 {C.CPJDA} - Absolute address of Job Description Table {JDT}

W.SSW
W.CPSSW

Word 43
Byte 4 upper 6 bits {C.CPSSW} - sense switch settings.

Words 44 through 47 - RESERVED

W.CPFACT

Word 50
ACCOUNT parameter for Permanent Files

W.CPFST
Word 51
FST entry for next control card PRU

W.CKP
W.CPCKP
Word 52
Byte 3 {C.CPCON} - Console checkpoint flag.
First bit on if there is a checkpoint request.
Byte 4 {C.CPCKP} - Number of checkpoints requested.

W.CPOAE
Word 53
Byte 0 contains the request flag for operator assigned equipment.
Bit 47 {EL} = 1 for extended label processing
Bytes 1-2 is the relative address of tape label information.
Byte 4 has the equipment assignment.

W.CPVRNO
Word 54
Sequential pack VID assignment

W.CPLDR1
Word 55
Byte 0 {C.CPLW or C.CPLT} - loader flags

<u>C.CPLW Values</u>:

Bit	1	S.CPLP	Program loaded from non-system library
	2-3	L	Library set indicator
	4	S.CPLT	Debugging aid flag
	5	R	Reduce flag
	6-9	M	Map Options
	10-11	W	Indicator for loader to be used

W.CPLS
W.CPLDR2
Word 56
The Library Set

W.CPLDR3
Word 57
User library name or indices to LNT entries.

W.CPLDR1{55}
W.CPLDR2{56}
<u>W.CPLDR3{57} Global Library Set Indicators</u>:

00	End of global library set
01-76	LNT ordinal of system library
77	User library; lfn of first user library in W.CPLDR3; lfn of second user library

Sep 2 8.4

W.CPAR	Word 60	
	Bytes 3-4 {C.CPAR}	- Last auto-recall request pointer to program address. Program remains in recall until low order bit of the word addressed is set to 1, indicating operation complete.
W.CPSTG	Word 61	
	Byte 0 {C.CPTMT}	
	Byte 1 {C.CPTNT}	
	Byte 4 {C.CPMNT}	
W.CPDFMC	Word 62	
W.CPDPV	Byte 0 {C.CPDFMC}	- Dayfile message count
W.CPIRB	Byte 2 {C.CPRBID}	- INTERCOM BATCH routine ID.
	Byte 4 {C.CPDPV}	- Job Dependency ID
W.CPFP	Word 63	
W.CPOUT	Byte 0 {C.CPFLAG}	- Flags
W.CPFLAG		

C.CPFLAG Values:

Bit	0	S.CPLDAF	MDI interlock
	1		
	2	Reserved	Private pack overflow
	3	S.CPNFNT	If on, do not search FNT
	4-12	Reserved	

Byte 2 {C.CPFST} - FST address
Byte 3 Lower 6 bits {C.CPRERN} - Rerun priority
Byte 4 {C.CPFP} - Flags

C.CPFP Values:

Bit	0	S.CPL	Reprocess
	1	S.CPG	Abort
	2	S.CPA	No rerun
	3	S.CPS	Sequencer
	4	S.CPN	Checkpoint taken
	5	S.CPX	EXIT card encountered
	6	S.CPDP	Private disk pack
	7	S.CPEOR	Control card EOR unused
	8	S.CPJFL	Job card field length assigned
	9	S.CPJ	JANUS
	10	S.CPR	Remote Batch
	11	S.CPE	INTERCOM

W.CPMSLM Word 64
Bytes 1-2 contains the mass storage limit in PRUs
Bytes 3-4 {lower 18 bits} - Running PRU count

W.CHTIM	Word 65 Bytes 2-3 contain channel time in seconds for accounting. Byte 4 - channel time {milliseconds}	
W.CPMSI	Word 66 Word is RESERVED	
W.CPSR	Word 67	
	Byte 1 {C.CPSR}	- Number of stack requests which cause access for this control point.
	Byte 4 {C.CPESR}	- 1 indicator which is non-zero during ECS-Disk transfer.
W.CPCAF	Words 70 through 167 Control Card Buffer	
W.CPCAL	Word 70 {W.CPCAF} contains the relative first word address within control point area of the 100B word buffer containing the current control card PRU. Word 167 {W.CPCAL} contains the relative last word address within the control point area of the 100B word buffer containing the current control card PRU. Words 170 through 177 RESERVED for installations	

SYSTEM JOB EXCHANGE PACKAGE AREA {XPIDLA}

This is the area used by all CP CM resident systems programs, including IDLE, which run in user mode. The area contains the system exchange jump package, status words, input and output register information and request stacks for the Scheduler and CPMTR/MTR. The exchange package for CPMTR will reside in the System Job Exchange Package Area of the job that is executing.

scope 3.4

SYSTEM JOB EXCHANGE PACKAGE AREA

	59	47	35	29 23 14	11 0
T.XPSCH	Exchange Package for Scheduler and Storage Move				
W.CPUST/W.CPSLIC	C.CPSTAT Status Byte	C.CPSLIC Total CPU Time Slice Alotted		C.CPLINK Previous Active Control Point	Next Active Control Point
W.CPTIME	CPU-A Scheduler/Storage Move Time				
W.CPTIMB	CPU-B Scheduler/Storage Move Time				
W.PPTIME	PPU Idle Time				
W.XPIR	Scheduler Input Register				
W.XPOR	Scheduler Output Register				
Scheduler Only T.SCHRS	Scheduler Request Stack Table			L.SCHRS=4	
T.MTRRS	CPMTR MTR Request Stack			L.MTRRS=4	
T.XPIDLA	CPU-A Idle Program Exchange Package P=CP.MTR (If IP.ECSB≠0, FL=400000B ECS FL=17777777B B2=1 B5=L.ECSTK B7=0 A6=T.EPAGE)				
	Status of Idle Ex. Pkg.	Accumulated Slice Time		Exchange Package Links	
W.CPTIME	CPU-A Idle Time				
W.XPIR	I D L E 1				
T.XPIDLB	(If IP.MCPU=2) CPU-B Idle Program Exchange Package P=CP.MTR (If IP.ECSB≠0, FL=400000B ECS FL=17777777B B2=1 B5=L.ECSTK B7=1 A6=T.EPAGE)				
	Status of Idle Ex. Pkg.	Accumulated Slice Time		Exchange Package Links	
W.CPTIME	CPU-B Idle Time				
W.XPIR	I D L E 2				

PP COMMUNICATION AREA

The PP Communications Area contains up to twenty 8-word areas, one for each PP, through which the PPs communicate with each other.

T.PPCX {where X = 1, 2,..., 20}	First word address of each area.
W.PPIR	Word 0 - relative location of the PP input register within a PP communication area.
W.PPOR	Word 1 - relative location of the PP output register within PP communications area. For $PP_2 - PP_n$ Byte 0 contains a MTR function code.
W.PPMESx {where X = 1,2,...,6}	W.PPMESx are the relative locations of the six words of the PP message buffer within a PP communication area.

Each peripheral processor contains pointers to its Input-Register, and Message Buffer in peripheral processor memory locations 74 and 75, respectively. The communication areas are used to provide a means of communication between MTR and peripheral processor programs. When a peripheral processor is idle, its resident program continuously scans its Input Register. When MTR has a task for that processor, it sets the name of the appropriate routine in the Input Register of the idle processor, which when it recognizes the request, loads the routine and executes it.

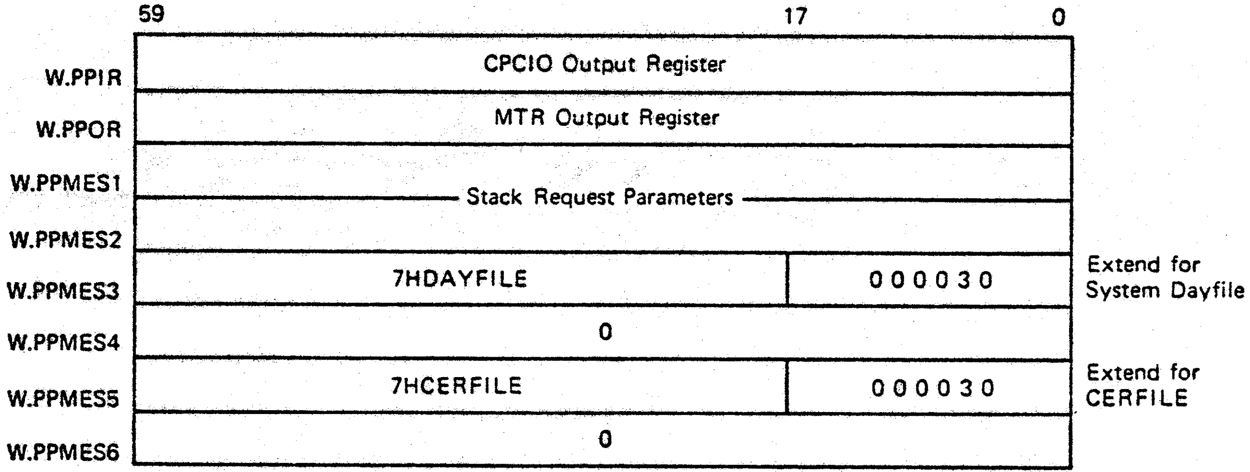
MTR regularly scans the Output Register of each active peripheral processor. When a peripheral processor requires MTR assistance {such as, for example, reserving a data channel}, it places a code in its Output Register. MTR detects the request during its scan of the output registers and processes it. When the request has been processed, MTR clears the requesting processor's Output Register; this informs the requesting processor that the request has been processed.

The six-word Message Buffer is used to pass parameters and messages between MTR and the peripheral processor resident programs.

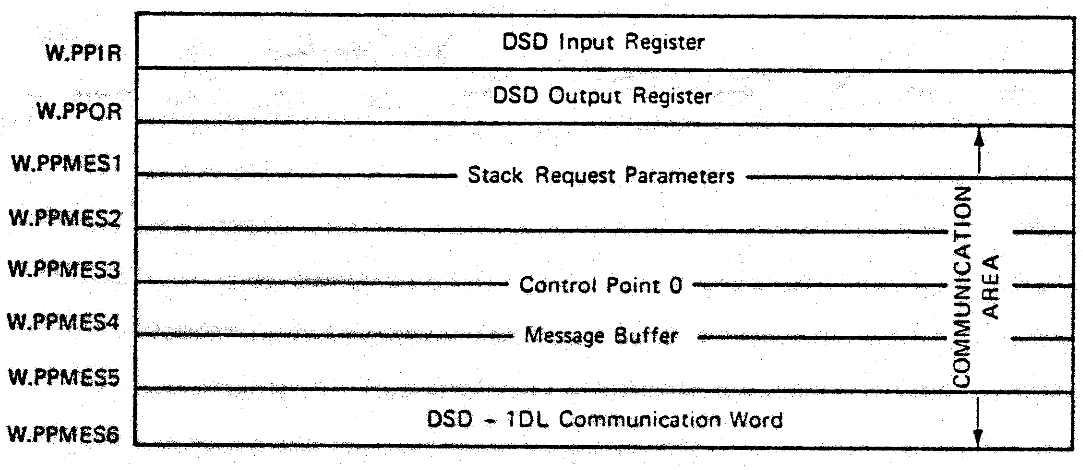
SCOPE 3.4

PP COMMUNICATION AREA

FOR PP0

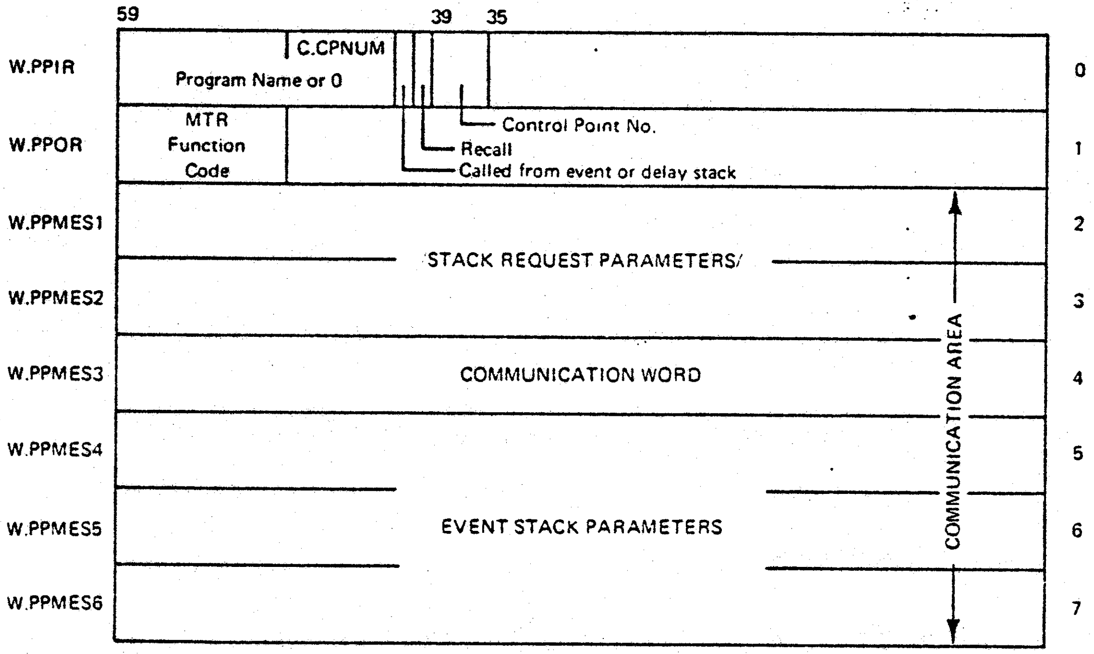


FOR PP1

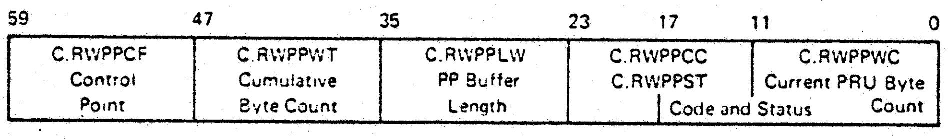


[illegible]

PP COMMUNICATION AREA
FOR PP2 THROUGH PPn



COMMUNICATION WORD



Scope 3.4

PP PROGRAM NAME RESERVATIONS

Routine Name	Description
ACE	Advance control card
CCP	6000 station routine
CEM	Central error manager for ECS
CIO	Preliminary I/O request processor
CKP	Saves information necessary to restart a checkpoint job
CLO	Dummy program used to call CIO
CON	INTERCOM-connect file to remote terminal
CP1	C.E.-415 card punch test
CR1	C.E.-405 card reader test
CY1	Resets FNT of file being processed by restart
DF4	C.E.-3234 test
DF7	C.E.-3553 test
DF8	C.E.-808 test
DIS	Console display program for a control point
DLE	C.E. Diagnostics
DMP	Dump CM
DPF	Dump permanent files to tape
DSP	Dispose function processor
EKG	Private pack closing-1EJ
EPF	Send audit information to CM
FAD	INTERCOM
FNT	INTERCOM-FNT alter routine
GBJ	INTERCOM-274 Graphics begin job
GEJ	INTERCOM-274 Graphics end job
IAP	INTERCOM-initiate another program
IEF	Routine for CEFAP
IUP	INTERCOM-initiate user program
JDP	Job dependancey count decrementor
LDL	Loader utility program
LDV	Loads CPU absolute overlays
LDW	Loads CPU absolute overlays in conjunction with LDV
LOC	Load octal corrections
LPF	In conjunction with LOADPF, reloads permanent files
LPT	C.E.-501 line printer test
LP1	C.E.-512 line printer test
MAC	INTERCOM
MDI	Used by EDITLIB to handle I/O involved in changing and moving directory
MEM	Process memory function
MES	INTERCOM-writes messages to remote terminal
MSD	INTERCOM
MSG	Issues dayfile messages
MTT	C.E.-60X tape test
MVJ	INTERCOM-Multi-User Job
OPE	Dummy program used to call CIO
PFA	Permanent file manager attach function
PFC	Permanent file managercatalog function

Routine Name	Description
PFD	Attaches permanent file directory to control point
PFE	Permanent file manager extend function
PFP	Permanent file manager purge function
PFR	Permanent file manager rename function
PFS	Permanent file manager position function
PPI	Reserved
PRM	Permission checking function
QAJ	Reserved
REQ	Makes non-allocatable device assignment and formats FNT entries for allocatable devices in response to request control card or a request macro call.
RMS	Routine for CERMS
RPV	Reprieve central program
RST	Restores control point area of restart job
RWE	INTERCOM-checks for INTERCOM job
SLT	Reserved
SRB	Used by EDITLIB to complete the disk address of a record
STS	Used by CP program to obtain certain status
TBL	INTERCOM-Get table
TDS	Terminate deadstart
TPF	Transfer permanent files and permanent file table
TPT	Transfer permanent file tables
T76	INTERCOM
VSM	STIMULATOR routine
VSN	Volume serial number card processor
XDQ	PP portion of dump queue
XRQ	PP portion of restore queue
0ZA-0Z9	PL10 drivers
1AJ	Advance job
1BR	INTERCOM-buffer manager
1BT	Blank label tape routine
1CI	INTERCOM-Queue Manager
1CL	Close function for all non-tape or non-permanent files
1CR	Tape read recovery - write CM for 9-track tapes
1CS	Tape read recovery - write CM for S tapes
1CT	Tape read recovery - write CM for SCOPE tapes
1C9	Write CM for tape read recovery
1DA	Process private packs
1DF	Dump dayfile
1DL	Overlay loader and dayfile message processor for DSD
1DM	Device queue manager
1DS	INTERCOM - H-display
1DU	Used in conjunction with DPF to clear dump flags and directory entries
1EJ	End of job processor
1EP	Twin stack processor for processing system double buffer stack requests for ECS buffering

SCOPE 3.4

Routine Name	Description
1FC	Creates an RBTC entry for PF catalog
1GJ	INTERCOM
1GM	Issues GOOD MORNING when time changes from 23.59 to 00.00
1GR	INTERCOM
1IB	Initiate batch job from input queue
1IS	INTERCOM-Send dayfile message to terminal
1IM	INTERCOM-Send message to terminal
1IQ	Initiate JANUS control point
1IR	Main JANUS routine; drives readers, punches, printers, etc.
1IS	Initialize overlay setup
1IU	Called by JANUS to backspace print file
1I1	INTERCOM-Initialization
1LT	Loads jobs from tapes
1LX	INTERCOM
1MF	Multifile positioning routine
1MH	Tape scheduling/prescheduling routine
1MT	Long record stranger tape driver
1NO	Tape read recovery noise record verifier
1NR	9 track tape read driver
1NW	9 track tape write driver
1N2	Tape read recovery noise record read forward 1
1N3	Tape read recovery noise record read forward 2
1OP	File open routine for non-tape files
1PC	Drop permanent file mass storage
1PD	Called by PFA to either enter event stack, call another PP routine or swap out
1PF	Permanent file queue manager
1PJ	INTERCOM - Process job card
1PK	Sequential disk pack close
1PL	Dummy plot program
1PS	6000 Station routine
1PT	INTERCOM-Low speed remote batch processor
1P1	Tape recovery to LGR positioning driver
1P2	Tape recovery write driver
1P3	Tape recovery verification driver
1P4	Tape recovery to LGNR positioning driver
1QM	INTERCOM-Check for MUJ swap-out completion
1QP	INTERCOM-Quantum calculator and MUJ serviceer
1RC	Restores field length of a checkpointed job
1RN	Ages queues, manages RBT chains and statuses tape drives
1RP	End of reel processor
1RQ	REQ overlay
1RS	Read stranger tape driver
1RT	Read SCOPE tape driver
1RV	Tape I/O read recovery driver initializer and terminator
1R2	Tape read recovery - tape parity error recovery 1

SCOPE 3.4

Routine Name	Description
1R3	Tape read recovery - tape parity error recovery 2
1R9	SCOPE tape 9 track {659} read tape driver
1SI	Routine to swap-in or roll-in a job
1SO	Swap-out or roll-out a job
1SP	Mass storage I/O processor {stack processor}
1SX	Error message and abort function for stack processors
1S5	Load and execute 1SP or 3DO at second entry
1TD	Dump output files to tape
1TF	Tape forward motion routine
1TO	Tape open routine
1TS	Tape sampler
1VG	STIMULATOR routine
1WB	INTERCOM-Wideband driver
1WI	SCOPE internal tape write driver
1WS	Stranger tape write driver
1W9	SCOPE tape 9 track {659} write tape driver
1XG	INTERCOM-1XP overlay used for graphics
1XP-6XP	INTERCOM-High speed EXPORT processor
1ZA-1Z9	INTERCOM drivers
2GJ	INTERCOM
21S	Reservoir of routines for 1IS
2LP	3256/3659 driver for an on-line print file
2ME	INTERCOM-Message sending routine
2PC	3446 card punch driver
2RC	3447 card reader driver
2RP	Overlay to 1RP-End-of-reel processor
2TA	Tape assignment overlay
2TB	All backward tape motion
2TC	Extended trailer label group processor
2TJ	Translate job card
3CF	INTERCOM - Overlay to 1CI
3CI	INTERCOM - Overlay to 1CI
3CT	INTERCOM - Overlay to 1CI
3CU	INTERCOM - Overlay to 1CI
3CX	INTERCOM - Overlay to 1CI
3DO	Initialize allocatable device file
3EP	ECS version of 6603-I disk driver
3EQ	ECS version of 6638 disk driver
3ER	ECS version of 865 drum driver
3ES	ECS version of 854 disk packs driver
3ET	ECS version of 6603-1I driver
3EU	ECS version of 814 disk driver
3EV	ECS version of 821 disk driver
3EW	ECS version of 841 MDD driver
3LX	INTERCOM-Overlay to 1LX
3ME	INTERCOM-Overlay to 2ME
3PK	User-pack initialization
3PM	Segment of 1P1 used for holding code for future use
3PO	Segment of 1P3 that processes uncorrectable parity error GO or RECHECK code

Routine Name	Description
3PS	Segment of 1P4 used for holding code for future use
3RQ	REQ overlay containing 2TACOM
3SP	Driver for 6603-I disk
3SQ	Driver for 6638 disk
3SR	Driver for 865 drum
3SS	Driver for 854 disk packs
3ST	Driver for 6603-II disk
3SU	Driver for 814 disk
3SV	Driver for 821 disk
3SW	Driver for 841 MDD
3SY	844 driver
3TT	INTERCOM-Transmit data from CPU to terminal
3T1-3T2	INTERCOM-Overlays to 3TT
4ES	Enter stack request
4LB	ANSI standard label processor
4LC	3000 label processor
4LX	INTERCOM-Overlay to 1LX
5DA	Initiate or destroy file on private pack
6BR	ANSI label processor read function code overlay
6BW	4LB overlay
6CR	3000 label processor read function code overlay
6CW	3000 label processor write function code overlay
6LC	Segment of 4LB or 4LC to load conversion table into MMTC
6LM	Segment of 4LB used to construct tape label messages
6L1	4LB inlay to convert PRU count
6L2	BCD conversion table inlay for 4LB
6L3	4LB inlay to check that proper conversion table is in the MMTC
6L4	4LB inlay for debug message writer
6L5	4LB inlay to format the lavel information
6L7	4LB inlay to pack and write label to tapes table
6MD	Dummy EDITLIB overlay
6NO	Tape error recovery debug segment assembled to give more detail about segment being read by 1NO
6PA	Prints system bulletin before header
6SI	Process Swap-in parity errors
6WM	Outputs dayfile error messages for I/O requests
7EC	Generate ECS buffers
7T1	ASCII/Displav code conversion table
7T2	EBCDIC/Display code conversion table
7W1-7W2	Overlay for 6WM
8AA-8A9	Reserved
8BA-8B9	Reserved
8CA-8C9	C.E.-Reserved names
8DA	A display overlay for DSD {dayfile buffers}
8DB	B display overlay for DSD {control point status}
8DC	C display overlay for DSD {central memory}
8DE	E display overlay for DSD {equipment status table}

SCOPE 3.4

Routine Name	Description
8DF	F display overlay for DSD {file name table}
8DH	H display overlay for DSD {I/O queues}
8DK	K display overlay for DSD {pointers and control point area}
8DL	L display overlay for DSD {central programmable}
8DM	M display overlay for DSD {PP communications area}
8DO	O display overlay for DSD {operator message}
8DP	P display overlay for DSD {tapes table and VSN previewing}
8DQ	Q display overlay for DSD {INTERCOM status}
8DR	R display overlay for DSD {JDT tables and queues}
8DS	S display overlay for DSD {job control area}
8DX	X display overlay for DSD {ECS memory}
8DY	Y display overlay for DSD {command format dictionary}
8DZ	Z display overlay for DSD {display dictionary}
8D1-8D9	DSD
8EA-8E9	DSD {7000 Station Displays}
8FA-8PS	Reserved
8GO	Loaded by 1R3 when GO or DROP operator dicision necessary during tape processing
8NO	Segment to 1N3 that writes debug messages to dayfile if IP.DBUG=1
8PT	INTERCOM - Overlay to 1PT
8PU-8W9	Reserved
8T3	Overlay to load MMTC memory
8XA	Channel commands overlay for DSD
8XB	Debugging commands overlay for DSD
8XC	PPU calling control points requests commands overlay for DSD
8XD	Equipment status commands overlay for DSD
8XE	Control point commands overlay for DSD
8XF	Deadstart commands overlay for DSD
8XG	Priority and tape staging job control commands overlay for DSD
8XH	INTERCOM commands overlay for DSD
8XI	Miscellaneous commands overlay for DSD
8XJ	Miscellaneous commands overlay for DSD
8XK	Tape scheduling commands overlay for DSD
8XL	Operator action manager commands overlay for DSD
8XM	Error flag commands overlay for DSD
8XN	CP-PP interlock commands overlay for DSD
8XO	Initiate system jobs command overlay for DSD
8XP	Tape assignment command overlay for DSD
8XQ	Bring up displays command overlay for DSD
8XR	Divert a file command overlay for DSD
8X1-8X9	DSD
8YA-8Y9	DSD {7000 Station Commands}
8ZA-8Z9	INTERCOM PP drivers
9AA-9A9	Customer Engineering
9YA-9YZ	
9ZA-9Z9	INTERCOM

SCOPE
3.4

SYSTEM TEXTS

System texts provide commonly used macro, micro, and symbol definitions for use in COMPASS source programs. SCOPE provides several text overlays which are loaded by COMPASS from the system libraries when specified by S parameters on the COMPASS control statement. S parameters can also be used on FTN control statements when FORTRAN source programs contain intermixed COMPASS subprograms. Up to seven system texts can be specified, each by a different S parameter, for a given assembler run. The system texts are made up of UPDATE common decks described below.

COMMON DECKS

System Action Request Macros: ACTCOM

IXi Xj*Xk
IXi Xj/Xk
IXi Xj/Xk,Bn
ABORT
CHECKPT
CLOCK
CONTRLC
DATE
DISPOSE
ENDRUN
FILESTAT
JDATE
LOADREQ
MEMORY
MESSAGE
RECALL
RECOVR
REQUEST
RTIME
SYSCOM
SYSTEM
TIME
TRANSR

Input/Output Macros using CPC: CPSYS

BKSP
BKSPRU
CLOSE
CLOSER
EVICT
FILEB
FILEC
LABEL
OPEN
POSMF
READ
READC
READIN
READN
READNS
READSKP
REWIND
REWRITE
REWRITEF
REWRITER
RFILEB
RFILEC
RPHR
SKIPB
SKIPF
UNLOAD
WPHR
WRITE
WRITEC
WRITEN
WRITEF
WRITER
WRITIN
WRITOUT

Record Manager Internal Text: RMCOM

Contains macro, micro, and symbol definitions used within Record Manager modules.

Installation Parameters: IPARAMS

Contains installation parameters as symbol and micro definitions.

Loader Request Macros: LMACOM

Contains two macros: LOADER and LDREQ.

Permanent File Macros: PFCOM

ALTER
ATTACH
CATALOG
EXTEND
FDB
PERM
PURGE
RENAME
SETP

Peripheral Processor System Definitions: PPSYS

Contains many system symbols and micros, and the following macros:

ADK
BIT
CEQU
CMICRO
CRI
ENM
JOBCARD
LDCA
LDK
PPENTRY
SBK
UJK

Integrated Scheduler Macros: SCHCOM

CISO
ENTRY34
LDW
SCHLOK
SCHSAVE
SCHSTOR
STREQ

Indexed Sequential Interface Macros: SISICOM

ACCESSK
ACCESSN
DELETE
FORCEW
INSERT
OPENNEW
OPENOLD
REPLACE
REPOS
SEEKL
SEEKS
SETBLKD
SETBLKI
SETCOLL
SETERR
SETFET
SETKEY
TERMNAT

Record Manager Definitions: bRMCOM

Contains macro, micro, and symbol definitions for user programs that use the Record Manager.

TEXT OVERLAYS

The SCOPE system text overlays contain various combinations of the common decks, as shown below:

CPCTEXT	System text for central processor programs using CPC. Common decks ACTCOM, CPSYS, and SISICOM.
IOTEXT	System text for central processor programs using Record Manager. Common decks ACTCOM and 6RMCOM.
IPTEXT	Installation parameter system text. Contains a single macro, IPARAMS, whose body is the IPARAMS common deck.
LDRTEXT	System text for central processor programs using Loader. Common deck LMACOM.
PFMTEXT	System text for central processor programs using permanent files. Common deck PFCOM.
PPTEXT	System text for peripheral processor programs. Common deck PPSYS.
SCHTEXT	System text for central and peripheral processor programs interfacing with the Integrated Scheduler. Common deck SCHCOM.
SCPTEXT	System text for central and peripheral processor programs in SCOPE. Common decks ACTCOM, CPSYS, and PPSYS.
SYSTEXT	System text for central processor programs. This is the default system text used by COMPASS when no S or G parameters are specified. It can be identical to either CPCTEXT or IOTEXT, at installation option. In the released system, SYSTEXT is equal to IOTEXT.
TXT6RM	System text for Record Manager modules. Common decks ACTCOM and RMCOM.

In addition to the above system texts provided by SCOPE, the following system texts are provided by product set members.

ALGTEXT	Contains COMPASS coded macros used to expand application areas of ALGOL-60.
FTNMAC	Contains macros used by COMPASS object programs produced by the FORTRAN Extended compiler {FTN}.
SMTEXT	Contains macros for central processor programs that call the SORT/MERGE system.

7-40

TEXT	ACTCOM	BRMCOM	CPSYS	IPARAMS	LDRCOM	PFCOM	PPSYS	SCHCOM	SMCOM	RMCOM	SISICOM	ALGTEXT
IOTEXT	X	X										
CPCTEXT	X		X								X	
SYSTEXT	X X	X	user selected SYSTEXT=IOTEXT or system default X	 user selected SYSTEXT=CPCTEXT							X	
IPTEXT				X								
LDRTEXT					X							
PFMTEXT						X						
PPTEXT							X					
SCHTEXT								X				
SCPTEXT	X		X				X					
SMTEXT									X			
TXTBRM	X									X		
ALGTEXT			X									X

SYSTEM TEXT

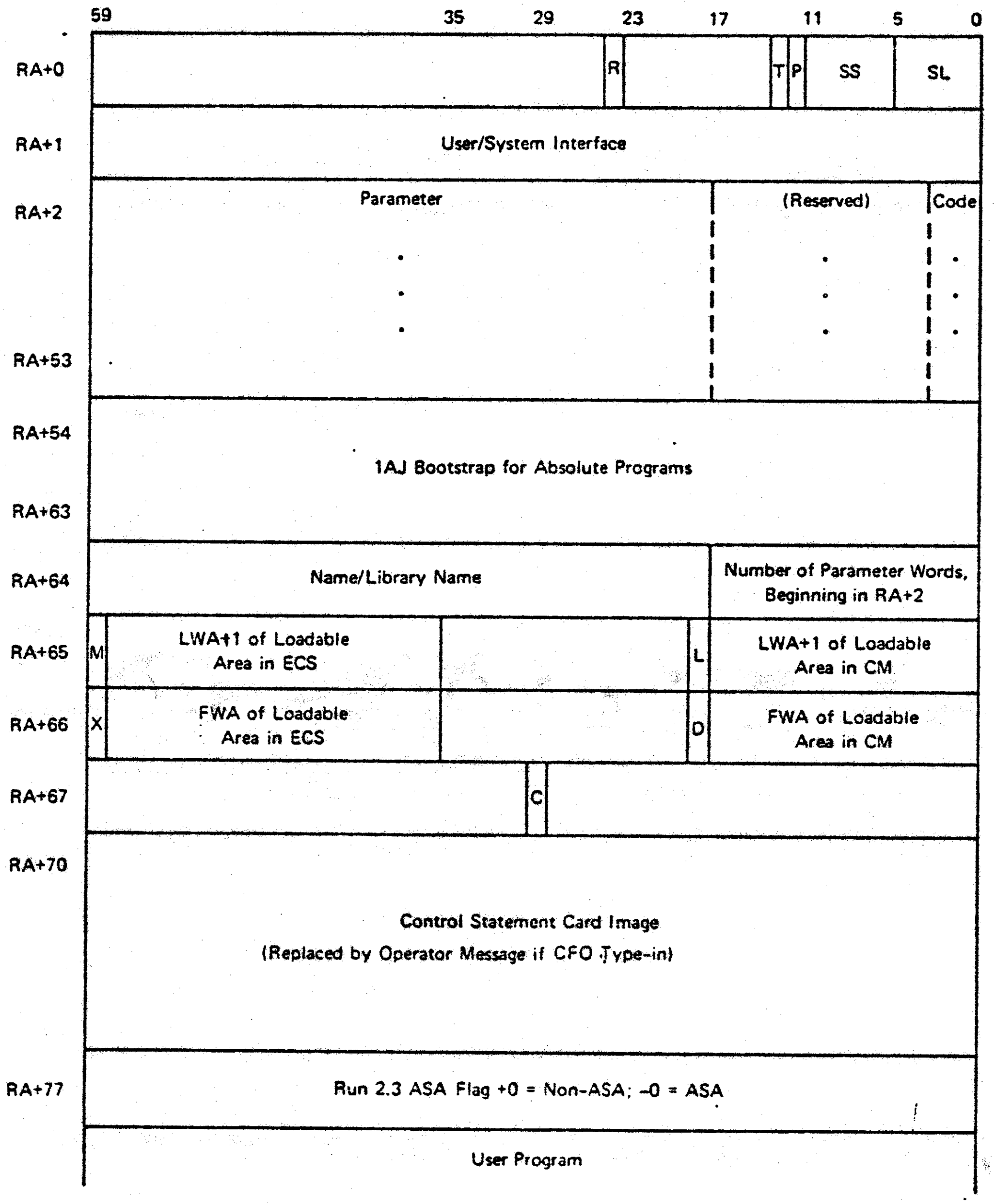
COMMON DECKS

Users View

RA COMMUNICATIONS AREA

The first 100 words of the user's field length is reserved for RA communications. This area is needed to allow the CP program to communicate with the system and the PP's.

RA COMMUNICATION AREA



NOTES ON RA COMMUNICATIONS AREA

RA — Reserved for use of hardware and software flags in the event of error.

R	Job dependency recheck bit
T	Storage move flag {1 = move being attempted}
P	Pause flag {1 = control point pausing}
SS	Sense switches
SL	Sense lights

RA+1 — If a user program wants to call a PP program, the call is placed in the RA+1 and then performs and XJ {CEJ - central exchange jump} to initiate CPMTR. CPMTR will execute certain RA+1 calls himself. If, however, the call should be assigned to a PP, the call will be passed to MTR. Should the XJ {CEJ/MEJ hardware} not exist on the machine, the CPMTR will be initiated by MTR, if he finds an RA+1 call in his normal scan.

Periodic Recall is accomplished by placing "RCL" left-justified into RA+1.

Automatic Recall for an RA+1 request is accomplished by setting bit 40 in RA+1. A CP program may put itself into auto-recall by putting "RCL" left-justified into RA+1 and setting bit 40 to one. The low order 18 bits will be, in any case, the address of the reply word.

RA+2-RA+63 — Contain control card parameters, if they exist. They are stored by 1AJ. As the control card is cracked, the following codes are used for special characters:

CODE
00 = Continuation
01 = Comma
02 = Equals sign
03 = Slash
04 = Left parenthesis
05 = Plus sign
06 = Minus sign
07 = Blank
10 = Semi-colon
11 =
12 =
13 = {reserved}
14 =
15 =
16 = Other
17 = Termination

RA+64-RA+67	1AJ records the total number of parameters in RA+64. This section is used by the first several Loader routines to record Loader information for modification of additional Loader routines.

L	Library/file flag {1 = name is library name}
X	XJ flag: If XJ = 1, an XJ can be issued
C	LDV completion flag {bit 29}
D	DIS RSS flag {bit 18}

SECTION EIGHT

PP RESIDENT

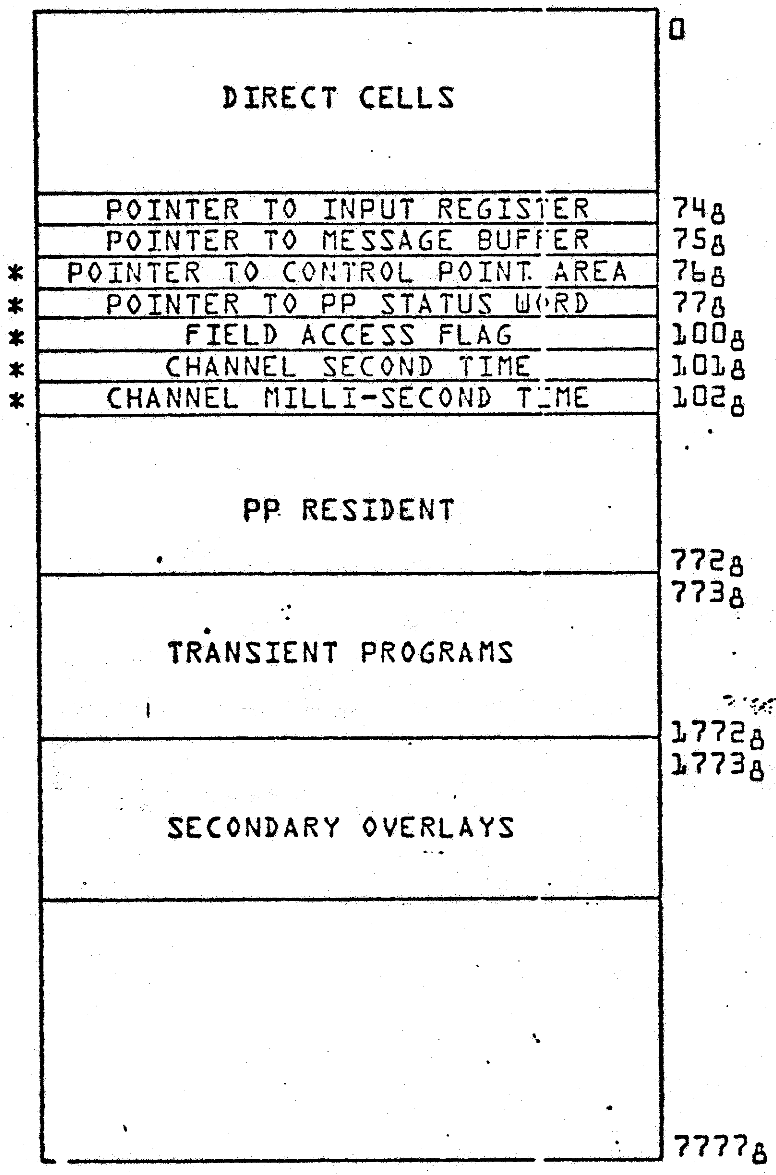
SECTION EIGHT - PP RESIDENT

CONTENTS

SECTION EIGHT - PP RESIDENT

CONTENTS

SCOPE
3.3



* Cells 76-102B constitute the five bytes of the PPU's central memory status word.

Fig. 1 Pool PP Layout

	DIRECT CELL			
*D.Za	00		SCRATCH AREA VOLATILE SCRATCH AREA !!	
	01			
	02			
	03			
	04			
	05			
	06			
	07			
*D.Tb	10			
	11			
	12			
	13			
	14			
	15			
	16			
	17			
*D.THc	20	D.FNT	WORDS 2 AND 3 OF FNT/FST ENTRY FOR CURRENT FILE	
	21	D.FNT+1		
	22	D.FNT+2		
	23	D.FNT+3		
	24	D.FNT+4		
	25	D.FNT+5		
	26	D.FNT+6		
	27	D.FNT+7		
*D.THd	30	D.FNT+8		
	31	D.FNT+9		
	32	D.EST	EST ENTRY FOR DEVICE BEING USED BY THIS PPU	
	33	D.EST+1		
	34	D.EST+2		
	35	D.EST+3		
	36	D.EST+4		
	37	D.DTS'	DEVICE TYPE AND ALLOCATION TYPE OR D.JFL JOB CM FL	
*D.FRe	40	D.BA	FET ADDRESS RELATIVE TO LOWER 18 BITS OF INPUT REGISTER	
	41	D.BA+1		
	42	D.BA+2		
	43	D.BA+3		
	44	D.BA+4		
	45	D.ECS	ECS FIELD LENGTH REQUIREMENT	
	46	D.CPR	COMPUTED PRIORITY	
	47	D.JTL	JOB TIME LIMIT/10B	
*D.FFf	50	D.PPIRB	CONTENTS OF PPU INPUT REGISTER	
	51	D.PPIRB+1		
	52	D.PPIRB+2		
	53	D.PPIRB+3		
	54	D.PPIRB+4		
	55	D.RA	REFERENCE ADDRESS/100B OF USER'S AREA	
	56	D.FL	FIELD LENGTH/100B OF USER'S AREA	
	57	D.FA	ADDRESS OF FILE FST ADDRESS	
*D.SXg	60	D.FIRST	CIRCULAR BUFFER	
	61	D.FIRST+1	FIRST PARAMETER FROM FET	
	62	D.IN	CIRCULAR BUFFER	
	63	D.IN+1	IN PARAMETER FROM FET	
	64	D.OUT	CIRCULAR BUFFER	
	65	D.OUT+1	OUT PARAMETER FROM FET	
	66	D.LIMIT	CIRCULAR BUFFER	
	67	D.LIMIT+1	LIMIT PARAMETER FROM FET	
*D.SVh	70	D.PPONE	THE CONSTANT "1"	
	71	D.HN	THE CONSTANT "100B"	
	72	D.TH	THE CONSTANT "1000B"	
	73	D.TR	THE CONSTANT "3"	
	74	D.PPIR	ADDRESS OF INPUT REGISTER	CM addresses
	75	D.PPMESi	ADDRESS OF PPU MESSAGE BUFFER	
	76	D.CPAD	ADDRESS OF CONTROL POINT	
	77	D.PPSTAT	ADDRESS OF PPU STATUS WORD	

*THE LOWER CASE LETTERS REPRESENT THE LEAST SIGNIFICANT DIGIT OF THE CELL NUMBER IN EACH CASE

Fig. 2 Direct Cells

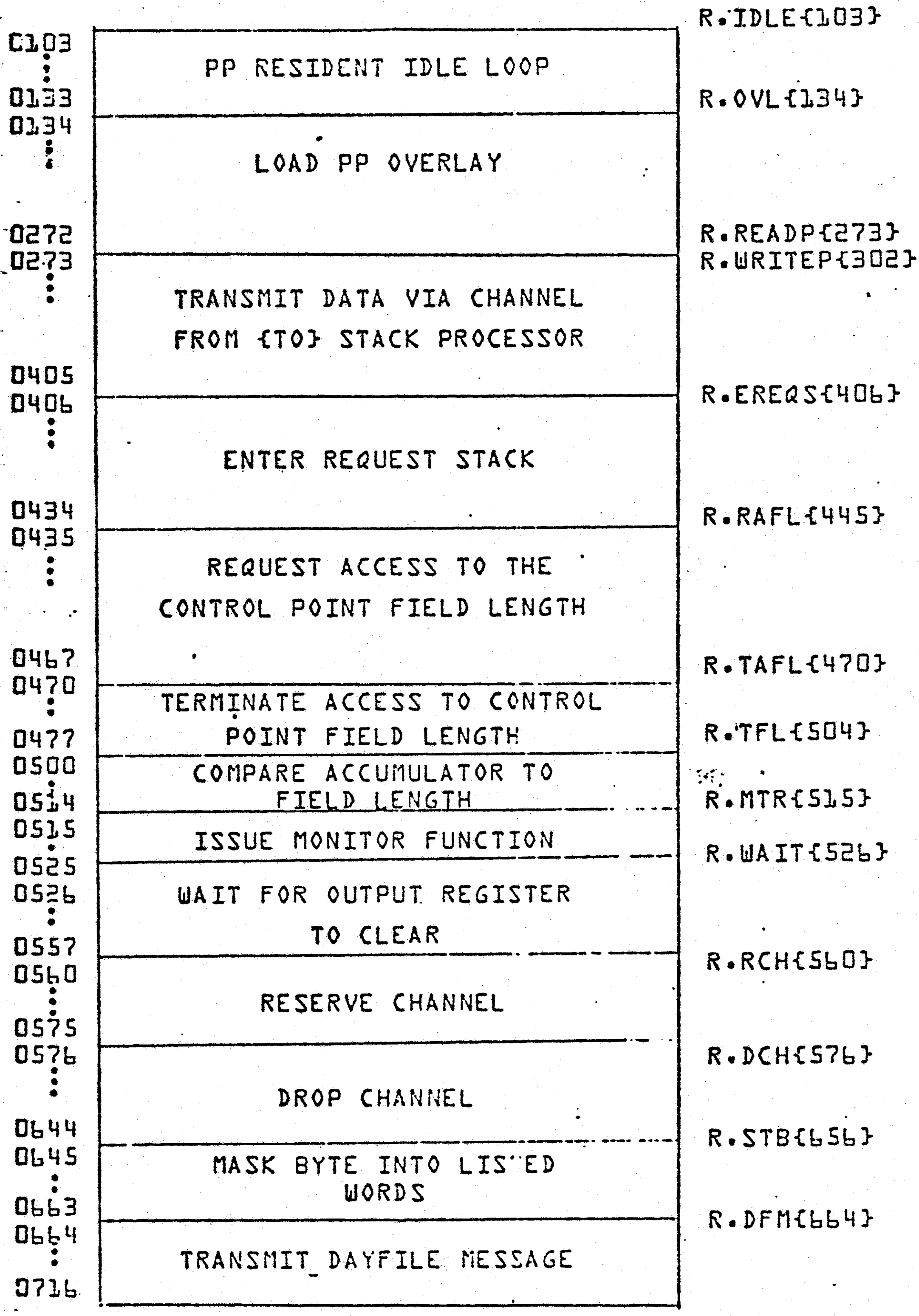


Fig. 3 PP RESIDENT ROUTINES

PERIPHERAL PROCESSOR RESIDENT

(COMMUNICATION CONVENTIONS) [illegible]

Introduction

In the SCOPE Operating System, the System Display program {DSD} and the Monitor program {MTR} permanently reside in two of the peripheral processors, 1 and 0 respectively. The remaining processors form a pool of processors to which MTR may assign tasks as required. These pool processors have no fixed assignments; any processor may be assigned to the execution of any system routine, and it is possible that more than one processor may be executing the same routine at the same time. All processors contain a small resident program which handles the communications between pool processor programs and the Monitor and initiates the execution of these programs as directed by MTR.

When SCOPE is Deadstarted a series of Deadstart PP programs are loaded into the PP's. The last Deadstart program to be loaded is named STL. It is loaded at location 100_8 in each of the pool PP's. The program STL contains PP Resident. STL starts executing at location 1000_8. When it is done it jumps to the PP Resident Idle loop, R.IDLE {see below}, and the PP is ready to load and run programs as directed by MTR.

Pool Processor Structure {See Fig. 1}

PP resident is contained in locations 0103_8 - 0772_8. When directed to do so by MTR, the resident loads a program into its memory and executes it; since that program remains in that processor only for the period of time required to perform its function, it is called a transient program. Transient programs occupy locations 0773 - 1772, although the first instruction is at location 1000. Transient programs generally load overlays to perform specific tasks. For example, CIO, which is a transient program, calls various overlays depending on the task {read, write, backspace} and the equipment {disk, tape, etc.} specified. Secondary overlays are loaded into memory beginning at location 1773, the first instruction falling at location 2000. Overlays are generally entered via a return jump. Transient programs have names beginning with a letter {CIO EXU} or the numeral 1 {1AJ, 1IO}; overlays have names beginning with a numeral 2 through 9 {2BP, 4LB, 9DM, etc.}.

Both transient and overlay programs, as well as the resident program, make extensive use of the low core locations 01-73. Figure 2 details these direct cell assignments.

The Resident

The peripheral processor resident program has two main functions to perform:

All communication between MTR and the transient or overlay programs is handled by the resident.

The resident, when directed by MTR, loads transient programs and initiates the execution of these programs.

Communication between MTR and the resident program is carried out through the use of PP communication areas in central memory, one for each processor. Each communication area consists of a one-word Input Register, a one-word Output Register, and a six-word Message Buffer.(See Fig. 7 of Section 2). Pool processors address these areas by means of pointers in locations D.PPIR, D.PPMES1 and D.PPSTAT (See Figure 2).

MTR assigns a task to a pool processor by placing the request in the processor's Input Register. The name of the program package which is to be loaded and executed appears in the high-order 18 bits of the Input Register. This name consists of three display code characters, such as 1AJ, CIO, etc. The number of the control point to which this package is assigned appears in the low-order three bits of byte 1 of the Input Register. Package parameters, such as the address of arguments required by the package, appear in the low-order 36 bits of the Input Register. The PP is given control to execute the code just loaded. The request itself remains in the Input Register until the task is completed. On completion of a task, the transient program requests MTR to release the processor; MTR then clears the processor's Input Register. The Input Register of a pool processor is thus clear only when the processor is idle.

All communication between the Monitor and the transient and overlay programs is handled by the resident program. MTR performs a variety of functions, each of which is identified by a function code of one or two octal digits. (See Section 7 for a detailed description of MTR's activity).

To transmit a request to MTR, the resident places the request in its Output Register. Byte 0 of the Output Register contains the function code in the low-order bit positions. Bytes 1 - 4 are used for arguments; the number of argument bytes depends on the particular function. Thus, for a Request Channel function (R.RCH=2), the channel number is placed in byte 1. For some functions, the function arguments are placed in the Message Buffer and only the function code appears in the Output Register. MTR regularly scans the Output Register of each processor to determine if a request is present. When the request has been detected, analyzed, and processed, MTR clears the Output Register. The resident, after placing the request in the Output Register, waits for the Output Register to be cleared before proceeding.

The resident contains a routine called R.MTR which handles the transmission of function requests to MTR. The R.MTR request routine uses locations D.T0-D.T4 in peripheral processor memory as temporary storage for the request to be written in the Output Register. A peripheral processor program may utilize the routine by placing the arguments for the function in bytes D.T1 through D.T4, setting the A register with the function number, and executing a return jump to R.MTR. Resident routine will enter the function number in location D.T0 and write the contents of locations D.T0-D.T4 in the Output Register. Control will be returned to the requesting program upon MTR's clearing the Output Register.

When a pool processor program completes execution, it exits to location R.IDLE, which is the address of the resident idle loop. The entry point to this idle loop is R.IDLE {103_8}. When referring to a PP Resident routine, the name of its entry point is used as the name of the routine. Thus the name of the idle loop is R.IDLE. In this idle loop, the processor's Input Register is scanned at intervals until a request is found in the Input Register. A delay between successive scans avoids unnecessary memory and read pyramid conflicts. Normally the PP Input Register of an idle PP contains zero. If the PP Input Register becomes non-zero, it means MTR wants PP Resident to load a PP transient program into the PP. When a request is detected, the resident stores the routine name and the control point. It then sends function R.TAFL, terminates access to the control point field length, to MTR and waits for MTR to clear the Output Register before continuing. When the Output Register is cleared R.IDLE calls the PP Resident subroutine R.OVL. R.OVL will then search the library directory for the requested routine; if found, the package is read from the resident library into the processor's memory beginning at location 773_8 {C.PPFWA-L.PPHDR}. If the routine is not found in the directory, the resident enters the message "XXX NOT IN PPLIB" in the dayfile, and requests MTR to abort the job which called the routine. The resident then returns to its idle loop. If the program is located, it is loaded by R.OVL which then returns control to R.IDLE which executes the instruction

```
LJM  C.PPFWA
```

This transfers control to the first instruction of the transient program. When a transient program terminates, the last instruction it must execute is

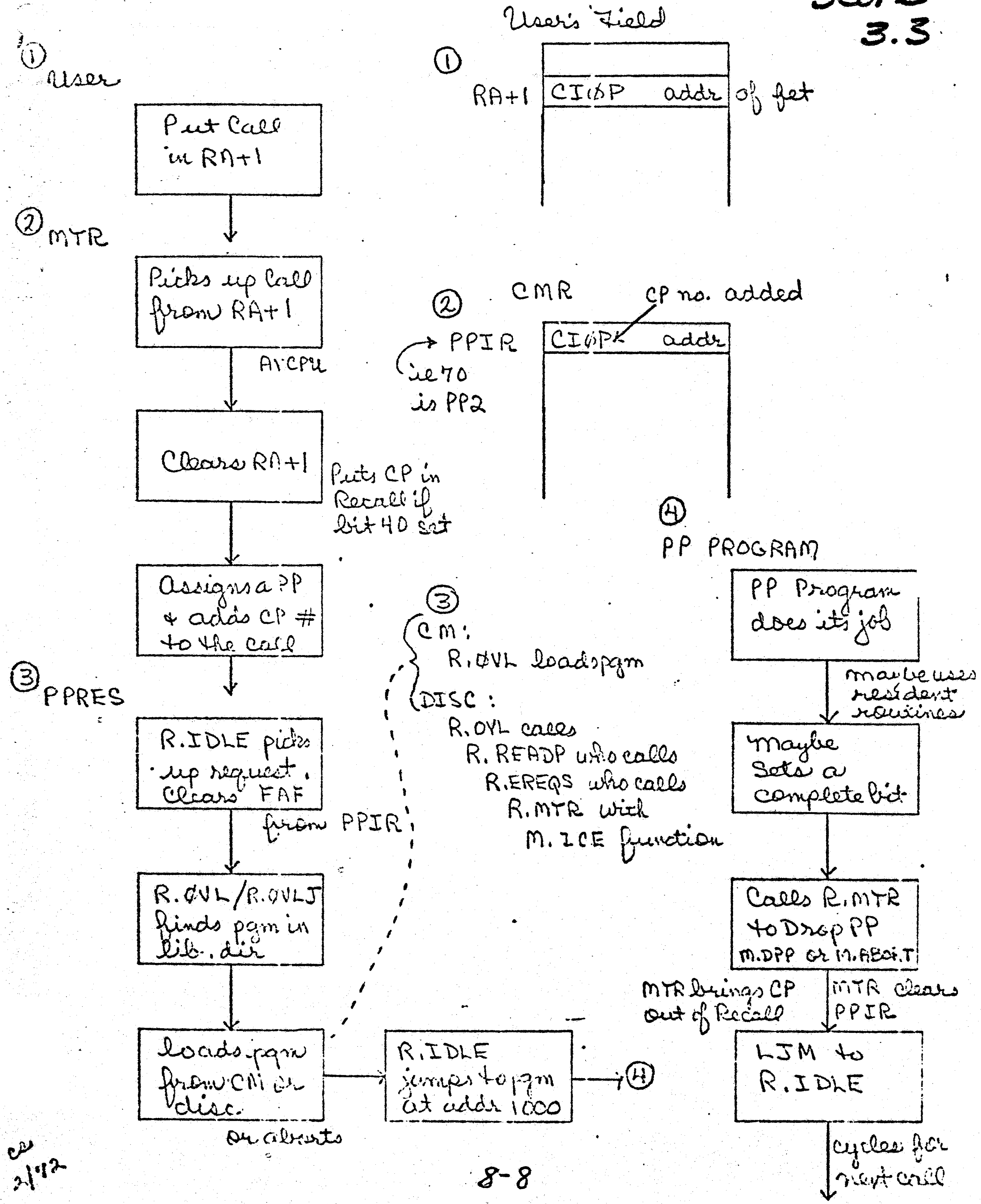
```
LJM  R.IDLE
```

<u>Resident Routines</u> {See Fig. 3}

Several resident routines and words are used by transient and overlay programs. These routines are described below. Location values are always subject to change.

COMMUNICATION : CP → MTR → PP

SCOPE 3.3



ce 2/72

R.IDLE

PSI SCOPE 3.3 Handbook

SCOPE
3.3

entry point name ↘ ↙ location in PP Resident

R.IDLE {103}

Calling Sequence: LJM R.IDLE

R.IDLE is the idle loop in which PP resident continually scans its input register for something to do.

R.IDLE reads the PP Input Register from Central memory into PP locations D.T6 - D.TW2. This is so that if a name is in the PPIR it will be positioned correctly for calling R.ØVLJ to load the PP program.

ie .

I A	J cp in			
D.T6	D.T7	D.TW0	D.TW1	D.TW2

Note that R.IDLE clears the field access flag (calls R.TAFL) — since R.IDLE is the last routine entered by a PP overlay it must be sure the flag is clear in case the overlay forgot to clear it. This also means a PP overlay called in by R.IDLE is entered with the flag *not* set.

Scope v. 3.3 PP Resident Listed under 3.3 System

```
                                   *     R.IDLE     PP RESIDENT IDLE LOOP
                                   *
                                   *     CALLING SEQUENCE
                                   *
                                   *     LJM        R.IDLE
                                   *
                                   *     ACTIONS
                                   *
                                   *     CYCLE WAITING FOR NON-ZERO INPUT REGISTER
                                   *     IF AN OVERLAY IS SPECIFIED, WRITE PP STATUS WORD IN CM,
                                   *        SET D.CPAD, LOAD THE OVERLAY AND TRANSFER CONTROL TO IT
                                   *


      0103     3074                R.IDLE   LDD    D.PPIR         READ INPUT REGISTER
      0104     6016                         CRD    D.T6
      0105     3017                         LDD    D.T7
      0106     0512                         NJN    IDL2           NO REQUEST, CLEAR PP STATUS WORD
      0107     3077                         LDD    D.PPSTAT
      0110     6216                         CWD    D.T6           CLEAR PP STATUS WORD
      0111     1470                IDL1     LDN    ILDELAY/2-6
      0112     1701                         SBN    1
      0113     0576                         NJN    *-1            DELAY ILDELAY MICROSECONDS

      0114     3074                         LDD    D.PPIR
      0115     6016                         CRD    D.T6           READ INPUT REGISTER
      0116     3017                         LDD    D.T7
      0117     0471                         ZJN    IDL1           NO REQUEST, LOOP
      0120     1207                IDL2     LPN    L.CPNUM
      0121     1007                         SHN    7
      0122     3476                         STD    D.CPAD
      0123     0200 0471                    RJM  R.TAFL           UPDATE PP STATUS WORD

      0125     2000 0773           R.OVLJ   LDC    C.PPFWA-L.PPHOR
      0127     0200 0135                    RJM  R.OVL            LOAD OVERLAY
      0131     0100 1000                    LJM    C.PPFWA        ENTER OVERLAY
      0133                                  BSS    1
```

PSI Scope 3.3 Handbook

R.OVLJ

SCOPE 3.3

R.OVLJ (125)

Calling Sequence: Store name of overlay in D.T6, D.T7. ← note
LJM R.OVLJ

The R.IDLE routine contains an additional entry point named R.OVLJ. A PP program can load a transient program on top of itself without changing the Input Register by storing the name of the transient program left-justified in D.T6 and D.T7, and then executing a long jump to R.OVLJ. The program will be loaded at C.PPFWA - L.PPHDR and control will be transferred to location C.PPFWA.

* { R.IDLE destroys direct cells 20_8 through 22_8 and some of the temporaries. R.OVLJ and all other PP Resident routines destroy only temporary cells ($0 - 17_8$). } *

Note that when R.ØVLJ is called (as from R.IDLE) the pp program will be loaded at address 1000, with its header at 773-777 (an exact copy of the CM library word). It will then be entered at 1000.

ØVLJ means load the overlay *and* jump to it.

Example -

① R.IDLE calls R.ØVLJ to load all transient programs found in the PPIR (ie 1AJ)

② 1AJ calls LØD or LØQ into the same PP without changing the name in the PPIR. (This is a flag to the loader to know who called him).

get loader name LØD or LØQ

in 1AJ

```
4001                LDI    D.ZI                                  SELOADR
3416                STD    D.T6        STORE LOADER NAME         SELOADR
5001 0001           LDM    1,D.ZI                                SELOADR
3417                STD    D.T7                                  SELOADR
0100 0125           LJM    R.OVLJ                                SELOADR
```

call R.OVLJ to load LØD or LØQ at 1000

1RA calls 1RN on top of himself (once every second)

```
                                                                             1RA     1078
2034 2216                 LDC    OV.1RN            CALL 1RN - RELEASE CHAIN  1RA     1079
1014                      SHN    12                                          1RA     1080
3416                      STD    D.T6                                        1RA     1081
1071                      SHN    -6                                          1RA     1082
1377                      SCN    77B                                         1RA     1083
3417                      STD    D.T7                                        1RA     1084
0100 0125                 LJM  R.OVLJ                                        1RA     1085
```

1AJ calls REQ on top of himself to handle REQUEST card.

8-13

(H) = 6 ?

```
                 *                                                           1AJ      710
                 *                                                           1AJ      711
                 *        CALL REQ INTO THIS PP                              1AJ      712
                 *                                                           1AJ      713
       220521    OV.REQ   EQU    OV.REQ                                      1AJ      714
6010             REQEND   CRD    D.T0              PLACE REQ WITH INTERNAL CALL 1AJ   715
2000 2205                 LDC    2RRE              BIT SET INTO THE PP OUTPUT REGISTER 1AJ 716
3410                      STD    D.T0                                        1AJ      717
3076                      LDD    D.CPAD                                      1AJ      718
1070                      SHN    -7                                          1AJ      719
2100 2140                 ADC    1RQ*64+40B                                  1AJ      720
3411             RPQV     STD    D.T1                                        1AJ      721
3074                      LDD    D.PPIR                                      1AJ      722
6210                      CWD    D.T0                                        1AJ      723
0100 0103                 LJM  R.IDLE              GO CALL IN REQ            1AJ      724
```

flag

puts REQ in PPIR

R.OVL {134}

```
                        STORE NAME OF OVERLAY IN D.T6, D.T7
    Calling Sequence:   Load A register  Load Address
                        RJM  R.OVL
```

The subroutine R.OVL is called by R.IDLE to load transient programs. It can also be called by a transient program to load an overlay, or by an overlay to load another overlay. To load an overlay, the name of the overlay must be stored left-justified in direct cells D.T6 and D.T7. The address in the PP at which the overlay is to be loaded is loaded in the A Register and a return jump is executed to R.OVL. When the overlay is loaded R.OVL returns control to the calling program which can transfer control to the overlay immediately or after some additional processing. For example the overlay 4LB can be loaded and executed as follows:

```
OV.4LB   EQU   OV.4LB                    GENERATE CROSS REFERENCE
         LDC   3RB4L                     LOAD "B4L"
         STD   D.T6                      STORE "4L" IN D.T6
?        SHN   -6
         SCN   77B
         STD   D.T7                      STORE "B" IN D.T7
         LDD   C.PPFWA-L.PPHDR           ADDR TO LOAD OVERLAY
         RJM   R.OVL
         RJM   C.PPFWA+1                 JUMP TO EXECUTE OVERLAY
```

A PP program can load another transient program on top of itself by modifying the Input Register to contain the name of the new program and then executing a long jump to R.IDLE. For example, when 1AJ finds there are no more control cards, it calls the program 1EJ into the same PP to terminate the job. This is done as follows:

```
OV.1EJ   EQU   OV.1EJ                    GENERATE CROSS-REFERENCE
         LDD   D.PPIR                    LOAD ADDR OF INPUT REGISTER
         CRD   D.T0                      READ INPUT REGISTER
         LDN   1RE-1RA
         RAD   D.T0                      CHANGE FIRST BYTE FROM "1A"
                                         TO "1E"
         LDD   D.PPIR
         CWD   D.T0                      WRITE NEW INPUT REGISTER
         LJM   R.IDLE                    JUMP TO LOAD AND EXECUTE 1EJ
```

R.OVL is used both by PP overlays to load higher level overlays and by PP resident to load the overlay named in the input register. PP resident does not reference the disk directly to load disk resident overlays but makes a call to the stack processor by calling R.READP.

R.OVL is called to load all overlays. It loads the overlay but does not jump to it

8-14

Examples:

① R.ØVLJ calls R.ØVL to load any transient overlay, passing the address 773 in A.

② 1AJ calls 1EJ into the same pp by changing the PPIR and calling R.IDLE (who calls R.ØVLJ who calls R.ØVL)

```
         1A+4=1E
1404 <-             CAL1EJ1   LDN     1RE-1RA                                   1AJ     439
                    *                                                           1AJ     440
3350                CAL1EJ2   RAD     D.PPIRB                                   1AJ     441
3374                          LDD     D.PPIR                                    1AJ     442
6250                          CWD     D.PPIRB  <--- change PPIR in CM           1AJ     443
0103 0103                     LJM  R.IDLE                                       1AJ     444
```

③ Any transient program calls it's overlay by putting the name in D.T6 + D.T7, the address in A.

How To Call an Overlay into a PP, Execute it, and Return to the Caller

```
              LDC   ØV.2BP
              RJM   CRPL
               {
                
      CRX     LJM    **
      CRPL    EQU    *-1
              SHN    12         }
              STD    D.T6       }  set up name of
              SHN    -6         }     overlay
              SCN    77B        }
              ST.D   D.T7       }
              LDC   C.PPTWA-L.PPHDR
              RJM    R.ØVL
              RJM    C.PPTWA+1  <---- enter 2BP at
              UJN    CRX              address 2001
               {                      if it begins with
                                      ENM
```

```
                                *       R.OVL      LOAD PP OVERLAY
                                *
                                *       CALLING SEQUENCE
                                *       STORE      OVERLAY NAME IN D.T6,D.T7
                                *       LOAD       ADDRESS TO START LOADING OVERLAY
                                *       RJM        R.OVL
                                *
                                *       ACTIONS
                                *
                                *       SEARCH DIRECTORY FOR OVERLAY
                                *       IF NOT FOUND, ISSUE OVL ERROR MESSAGE,ABORT CONTROL POINT
                                *                          AND EXIT TO R.IDLE
                                *       IF FOUND, LOAD THE OVERLAY FROM CM, ECS OR DISK
                                *


     0134                       R.OVL   ENM        X
     0135       5400 0211               STM        READCMOV+1        SAVE INPUT ADDRESS
     0140       5400 0256               STM        READPREQ+5*W.STPFW+C.STPFW
                                        IFNE       ECSLIB,0,1
                                        STM        READEMOV+1
                                                                ← hang PP while Edit66 being [illegible]
     0142       1401            OVL     LDN        P.LIO
     0143       6010                    CRD        D.T0              READ LIBRARY POINTER
     0144       3010                    LDD        D.T0+C.DIRFWA
     0145       1014                    SHN        12
     0146       3111                    ADD        D.T1+C.DIRFWA
     0147       0472                    ZJN        OVL
     0150       6010                    CRD        D.T0              READ FWA-1 OF PROGRAM DIRECTORY
     0151       1402                    LDN        2
     0152       3415                    STD        D.T5
     0153       1005                    SHN        5
     0154       3513                    RAD        D.T3              FWA OF PROGRAM DIRECTORY
     0155       3013            OVLP1   LDD        D.T3
     0156       1014                    SHN        12
     0157       3114                    ADD        D.T4
     0160       6115 0001               CRM        D.Z1,D.T5         READ DIRECTORY ENTRY
     0162       3414                    STD        D.T4              SAVE NEXT ADDRESS
     0163       1063                    SHN        -12
     0164       3413                    STD        D.T3
     0165       3006                    LDD        D.Z6+C.DIRPTR
     0166                               TESTZ      SHN,S.DIRPT,(-)       POSITION PROGRAM TYPE
     0167                               TESTZ      LPN,(8-S.DIRPT),(17B    )
     0170       0524                    NJN        PPCALLER          SENSE NOT TYPE PP
     0171       3001                    LDD        D.Z1
     0172       3316                    LMD        D.T6
     0173       0561                    NJN        OVLP1             SENSE NO HIT
     0174       3017                    LDD        D.T7
     0175       1377                    SCN        77B
     0176       3302                    LMD        D.Z2
     0177       0555                    NJN        OVLP1             SENSE NO HIT
     0200       3006                    LDD        D.Z6+C.DIRPTR
     0201                               TESTZ      SHN,S.DIRPR,(-)       POSITION RESIDENCE BYTE
     0202       1203                    LPN        3
                                        IFEQ       ECSLIB,0
```

```
                                 ELSE
                                 ZJN        READCM         JUMP IF RESIDENCE = CM
                                 SBN        1
                                 ZJN        READISK        JUMP IF RESIDENCE = DS
                                 LJM        READECS        JUMP IF RESIDENCE = ECS
                                 ENDIF

 0204    3007         READCM     LDD        D.Z6+C.DIRCMA  FETCH FWA OF PROGRAM
 0205    1277                    LPN        77B
 0206    1014                    SHN        12
 0207    3113                    ADD        D.Z7+C.DIRCMA
 0210    6105 0000    READCMOV   CRM        **,U.45        READ OVERLAY
 0212                            UJK        R.OVLX

                                 IFEQ       ECSLIB,0
 0214    3075         PPCALLER   LDD        D.PPMES1
 0215    6216                    CWD        D.T6
 0216    1403                    LDN        M.OVLERR       ISSUE PP CALL ERROR MESSAGE
 0217    0200 0516               RJM        R.MTR
 0221    1413         GETOUT     LDN        M.ABORT        ABORT CONTROL POINT
 0222    0200 0516               RJM        R.MTR
 0224    0100 0103               LJM        R.IDLE         EXIT TO IDLE LOOP
                                 ELSE
                      PPCALLER   LDN        0              GO ISSUE PP CALL ERROR MESSAGE
                                 LJM        BLAH
                                 ENDIF

 0226    3010         READISK    LDD        D.Z6+C.DIRRBA  RBT WORD PAIR ADDRESS
 0227    5400 0247               STM        READPREQ+5*W.STPRBA+C.STPRBA
 0231    3011                    LDD        D.Z6+C.DIRRBN  RBT NUMBER
 0232    5400 0250               STM        READPREQ+5*W.STPRBN+C.STPRBN
 0234    3012                    LDD        D.Z6+C.DIRPRU  PRU WITHIN RB
 0235    5400 0251               STM        READPREQ+5*W.STPPRU+C.STPPRU
 0237    2000 0247               LDC        READPREQ
 0241    0200 0274               RJM        R.READP        READ OVERLAY
 0243    3017                    LDD        D.T4+C.RWPPST
 0244    1066                    SHN        -9
 0245    0444                    ZJN        READCMOV+2     RETURN TO CALLER IF NO ERROR
                                 IFEQ       ECSLIB,0
 0246    0352                    UJN        GETOUT         ELSE GO ABORT CONTROL POINT
                                 ELSE
                                 LJM        GETOUT         ELSE GO ABORT CONTROL POINT
                                 ENDIF

 0247                 READPREQ   BSS        10
 0252                            ORG        READPREQ+5*W.STO+C.STO
 0252    0011                    VFD        12/0,RDPNP
 0257                            ORG        READPREQ+5*W.STFB+C.STFB
 0257                 T          BIT        S.STF+S.STFETP
 0257                 S          BIT        S.STF+S.STFNTP
 0257    1400                    VFD        12/S+T         NO FNT. NO FET
 0260                            ORG        READPREQ+5*W.STPLW+C.STPLW
 0260    7777                    DATA       -0

 0261                            BSS        READPREQ+10-*
 0261                            BSS        10
```

R.012

extension of R.ØVL at end of PPRES

```
                  IFNE      ECSLIB,0

         *        R.OVL EXTENSION TO LOAD FROM ECS

         READECS  LDD       D.T3
                  STD       D.T2              MOVE PNT ENTRY FWA+1
                  LDD       D.T4              FROM (D.T3,D.T4)
                  STD       D.T3              TO (D.T2,D.T3)
                  LDN       X.ECOVL
                  STD       D.T4              INITIATE CENTRAL EXECUTIVE
                  LDN       M.ICE             TO READ FROM ECS TO T.ECLBUF
                  RJM     R.MTR
                  LDC       **   (T.ECLBUF+1)  SET BY STL
         READEMOV CRM       **,D.Z5           READ OVERLAY FROM T.ECLBUF
                  SBD       D.Z5
                  SBN       1
                  CWD       D.T0              CLEAR INTERLOCK
                  LDD       D.T4
                  NJN       BLAH              GO ISSUE MESSAGE IF ERROR
                  LJM       B.OVLX            ELSE RETURN TO CALLER

         BLAH     STD       D.T4              TELL MTR WHICH MESSAGE...   note
                  LDD       D.PPMES1          ZERO = XXX NOT IN PP LIB
                  CWD       D.T6              NON-ZERO = ERROR LOADING XXX FROM ECS
                  LDN       M.OVLERR          PUT PROGRAM NAME IN MESSAGE BUFFER
                  RJM     R.MTR               MTR ISSUES DAYFILE MESSAGE
         GETOUT   LDN       M.ABORT
                  RJM     R.MTR               ABORT CONTROL POINT
                  LJM       R.IDLE            EXIT TO IDLE LOOP

                  ENDIF

         *        END OF PP RESIDENT

                  IFGT    *,C.PPFWA-L.PPHDR,1
                  ERR     PP RESIDENT IS TOO LARGE

```

R.RAFL

SCOPE 3.2

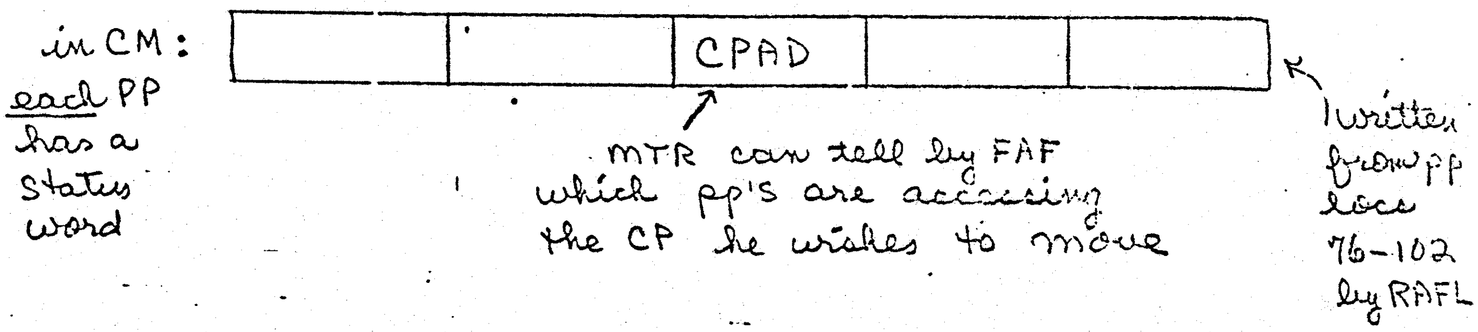
R.RAFL {435}

Calling Sequence: RJM R.RAFL

The subroutine is called to request access to the control point field length. A test is made on the storage move flag for the control point. If set, a call is made to R.TAFL to clear the field access flag in the PP status word, then pauses until the storage move is cleared. When it is cleared, set the field access flag in the PP status word and reset RA in D.RA, FL in D.FL.

Important notes:

When the FAF is set in the pp (loc 100) it contains the CPAD (from loc 76). When the FAF is set in CM it is the middle byte of the PP's status word:



R.RAFL is very tricky in that it sets the FAF, then checks the MOVE flag (in the CP area), unsets the FAF flag if MOVE was set & keeps on doing this until it is able to set the FAF & find MOVE not set. This is to keep MTR from accidentally moving the field by getting hung up with R.RAFL.

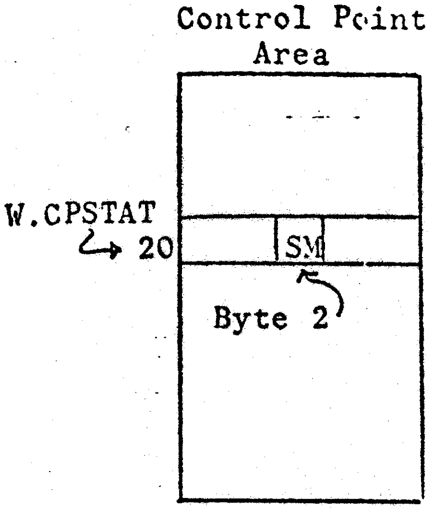
R.PAUSE (v 3.2) is EQU'd to R.RAFL

R. RAFL

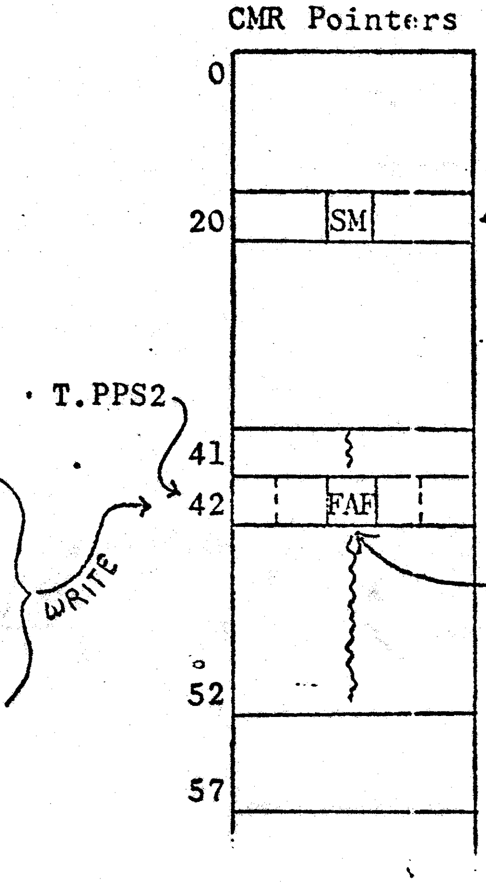
SCOPE 3.3

STORAGE MOVE PHILOSOPHY

FLAGS



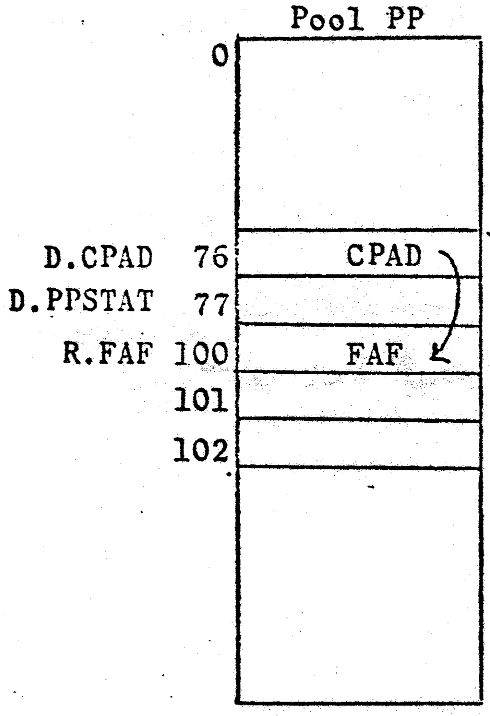
Word 20 of a CP Area is set
- when MTR is moving the associated user's field length, or
- when MTR desires to move the associated user's field length



Word 20 of CMR Pointers is set when MTR is moving <u>any</u> core

ie, word 42 is T.PPS2 - PP 2's status word

When set, FAF contains CPAD



When a PP desires to work on central memory, it sets the Field Access Flag:
- FAF in the PP
- byte 2 of T.PPSn in CMR Pointers (the PP writes his 76-102 to his T.PPSn)

The FAF being set in both the PP and CMR indicates the PP is working on the associated user's field length and prohibits MTR from moving that control point.

cs
11/72

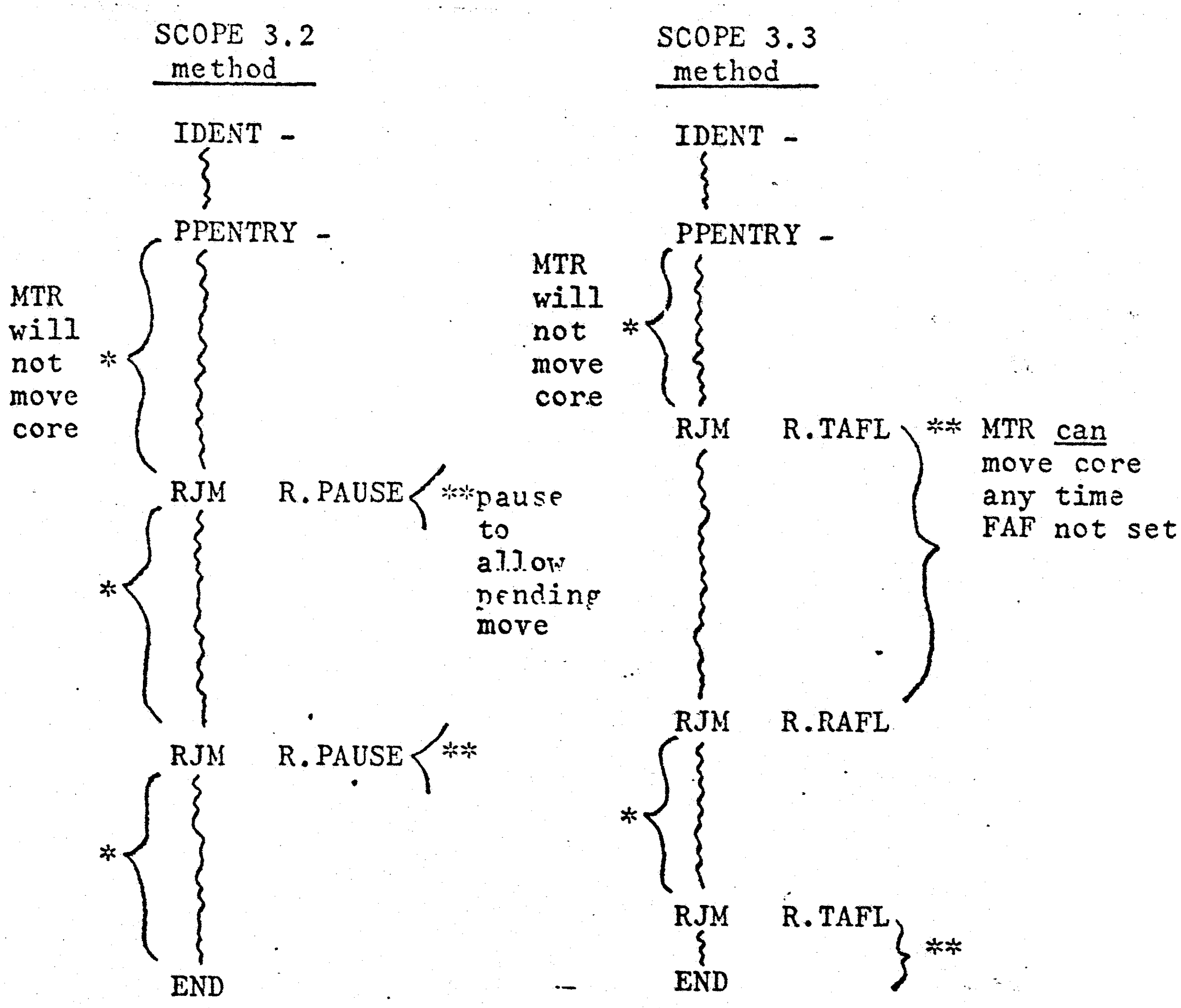
SM means Storage Move Flag; FAF means Field Access Flag

R.RAFL

Storage Move Philosophy

CODING

SCOPE
3.3



The Scope 3.2 philosophy was to <u>not</u> allow MTR to move core <u>until</u> a PAUSE was made.
The user periodically went to R.PAUSE to do this.

The Scope 3.3 and Scope 3.4 philosophy <u>is</u> to allow MTR to move core any time the FAF is <u>not</u> set.
The user must go to R.RAFL to set it -
and later to R.TAFL to clear it.

- next page -

R.RAFL

Protection:

- On entry, the flag is not set.
  The user must set it himself (by PPENTRY or calling R.RAFL)
- For upward compatibility from Scope 3.2, the PPENTRY macro calls R.RAFL - to set the flag.

  R.PAUSE is equated to R.RAFL -
  which steps and calls R.TAFL if a move is pending,
  then resets the flag.
- On exit, R.IDLE clears the flag.
  This is in case the user forgot to call R.TAFL before exit.

WARNING:

- Any Scope 3.2 program which did not use the PPENTRY macro may get its core moved before the first call to R.PAUSE.
- The PPENTRY macro must also now have two parameters: D.PPIRB and D.TO.

```
     ie,        PPENTRY      D.PPIRB,D.TO
```

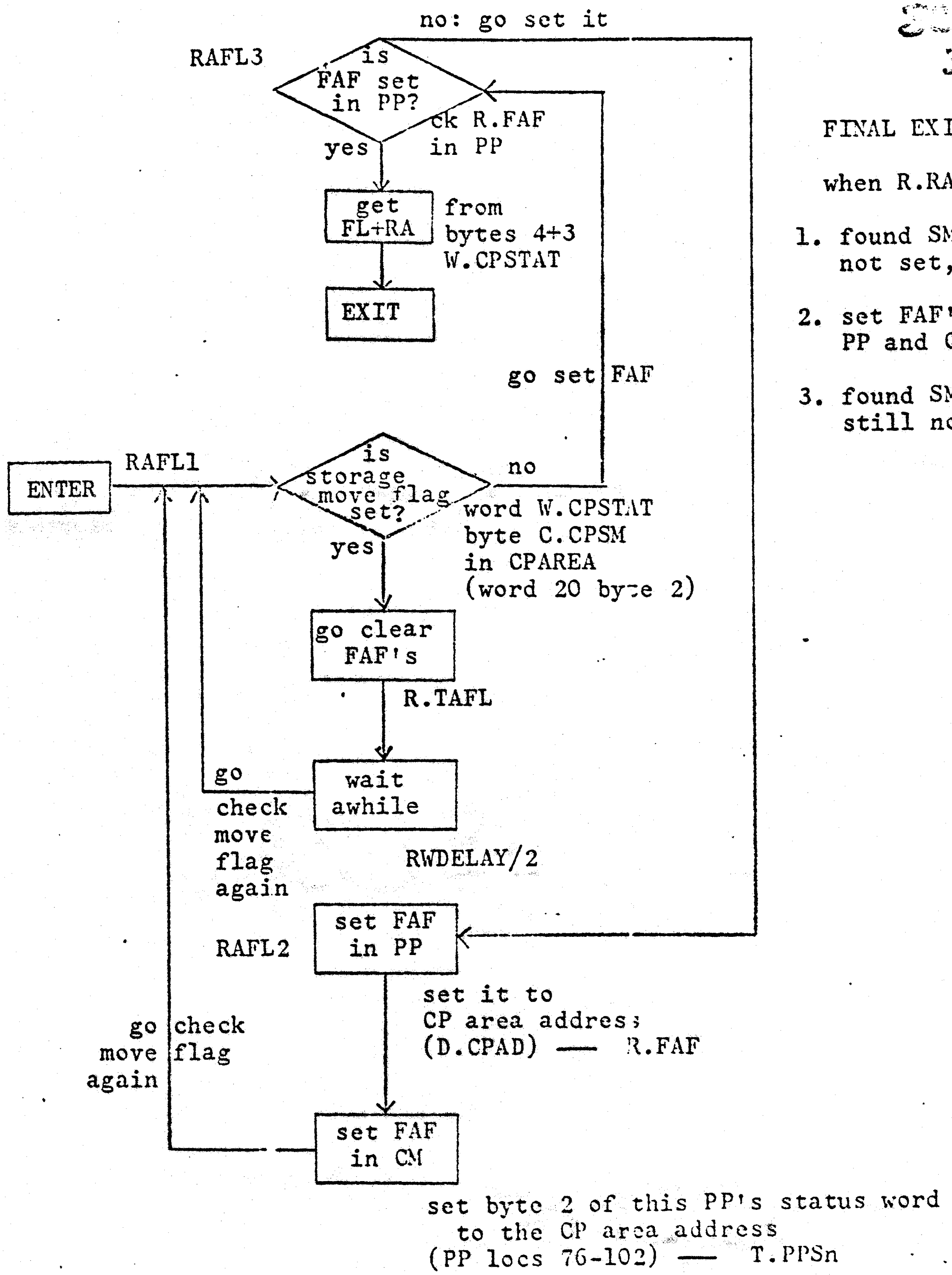
Central Memory Storage Move

SUMMARY

MTR MAY MOVE THE CENTRAL MEMORY FIELD LENGTH
OCCUPIED BY A CONTROL POINT
AT ANY OF THE FOLLOWING TIMES:

- UPON ENTRY TO THE PP PROGRAM
  - PRIOR TO A PPENTRY MACRO
    (PPENTRY CALLS R.RAFL)
  - PRIOR TO A PP PROGRAM CALL TO R.RAFL
    (R.RAFL SETS R.FAF, RESETS D.RA AND D.FL)
- AFTER (DURING) ANY CALL TO R.RAFL *
  - R.RAFL CALLS R.TAFL,
    THEN SETS R.FAF, RESETS D.RA AND D.FL
- AFTER (DURING) ANY MONITOR FUNCTION *
  - R.WAIT CALLS R.RAFL PERIODICALLY
- AFTER A CALL TO R.TAFL
  - R.TAFL CLEARS THE FAF's

**** * THE PP PROGRAM MUST RE-ABSOLUTIZE
ANY CENTRAL MEMORY ADDRESSES IT ALREADY HAS ****



FINAL EXIT:

when R.RAFL

1. found SM flag not set,

2. set FAF's in PP and CMR, and

3. found SM flag still not set

```
                                     *        R.RAFL    REQUEST ACCESS TO THE CONTROL POINT FIELD LENGTH      STL        326
                                     *                                                                         STL        327
                                     *        CALLING SEQUENCE                                                 STL        328
                                     *                                                                         STL        329
                                     *        RJM       R.RAFL                                                 STL        330
                                     *                                                                         STL        331
                                     *        ACTIONS                                                          STL        332
                                     *                                                                         STL        333
                                     *        TEST STORAGE MOVE FLAG FOR CONTROL POINT                         STL        334
                                     *        IF SET, CLEAR THE FIELD ACCESS FLAG IN THE PP STATUS WORD        STL        335
                                     *                  AND WAIT UNTIL THE STORAGE MOVE FLAG IS CLEARED        STL        336
                                     *        SET THE FIELD ACCESS FLAG IN THE PP STATUS WORD                  STL        337
                                     *        RESET RA IN D.RA, FL IN D.FL                                     STL        338
                                     *                                                                         STL        339


      0435      5000 0100           RAFL3     LDM       R.FAF                                                  STL        341
      0437      3376                          LMD       D.CPAD                                                 STL        342
      0440      0522                          NJN       RAFL2          FAF IS NOT SET, TRY AGAIN               STL        343
                                                                                                               STL        344
      0441      3014                          LDD       D.T0+C.CPFL                                            STL        345
      0442      3456                          STD       D.FL           RESET FL                                STL        346
      0443      3013                          LDD       D.T0+C.CPRA                                            STL        347
      0444      3455                          STD       D.RA           RESET RA                                STL        348
                                                                                                               STL        349
      0445                          R.RAFL    ENM                                                              STL        350
                                                                                                               STL        351
                         0000446    R.PAUSE   EQU       R.RAFL                                                 STL        352
                                                                                                               STL        353
      0447      3076                RAFL1     LDD       D.CPAD                                                 STL        354
      0450      1620                          ADN       W.CPSTAT                                               STL        355
      0451      6010                          CRD       D.T0           READ CONTROL POINT STATUS WORD          STL        356
      0452      3012                          LDD       D.T0+C.CPSM                                            STL        357
      0453      0461                          ZJN       RAFL3                                                  STL        358
                                                                                                               STL        359
      0454      0200 0471                     RJM       R.TAFL         CLEAR FIELD ACCESS FLAG                 STL        360
                                                                                                               STL        361
      0456      1462                          LDN       RWDELAY/2                                              STL        362
      0457      1701                          SBN       1                                                      STL        363
      0460      0576                          NJN       *-1            DELAY RWDELAY MICROSECONDS              STL        364
      0461      0365                          UJN       RAFL1                                                  STL        365
                                                                                                               STL        366
      0462      3076                RAFL2     LDD       D.CPAD                                                 STL        367
      0463      5400 0100                     STM       R.FAF          SET FIELD ACCESS FLAG                   STL        368
      0465      3077                          LDD       D.PPSTAT                                               STL        369
      0466      6276                          CWD       D.CPAD         UPDATE PP STATUS WORD                   STL        370
      0467      0357                          UJN       RAFL1                                                  STL        371
```

8-28

R.RAFL

R.TAFL {470}

Calling Sequence: RJM R.TAFL

R.TAFL is called to terminate access to the control point field length by clearing the field access flag in CM byte R.FAF.

```
                                        *         R.TAFL    TERMINATE ACCESS TO CONTROL POINT FIELD LENGTH   STL       373
                                        *                                                                    STL       374
                                        *         CALLING SEQUENCE                                           STL       375
                                        *                                                                    STL       376
                                        *         RJM       R.TAFL                                           STL       377
                                        *                                                                    STL       378
                                        *         ACTION                                                     STL       379
                                        *                                                                    STL       380
                                        *         CLEAR THE FIELD ACCESS FLAG IN THE PP STATUS WORD          STL       381
                                        *                                                                    STL       382


          0470                          R.TAFL    ENM       X                                                STL       384
          0472       1400                         LDN       0                                                STL       385
          0473       5400 0100                    STM       R.FAF               CLEAR FIELD ACCESS FLAG      STL       386
          0475       3077                         LDD       D.PPSTAT                                         STL       387
          0476       6276                         CWD       D.CPAD                                           STL       388
          0477       0370                         UJN       R.TAFLX                                          STL       389
```

8-28

R.TAFL

[illegible]

R.TFL

SCOPE
3.3

R.TFL {500} "Test Field Length"

```
Calling Sequence:   Load    relative address

                    RJM     R.TFL
```

R.TFL is used to insure that a relative address is within the field length. The 18-bit address is added to the control point reference address {RA} and compared with the field length. If the address is out of range, R.TFL will exit with a negative A register; if the address is legal, the A register will contain the absolute CM address {RA + relative address} upon exit. ~~The control point RA and FL are kept locally within PP resident at R.CPRA and R.CPFL, respectively. Since these locations are set by routine R.RAFL when a new transient program is initiated in a PP,~~ the transient program and its overlays cannot call R.TFL until R.RAFL has been called. Many PP programs do not call R.TFL but do their own checking of addresses.

not in 3.3

R.TFL should be used to check any addresses in a user's field to be sure they are within his limits

How To:

```
        }
        LDD     D.PPIRB+3
        SHN     12
        ADD     D.PPIRB+4
        RJM     R.TFL
        MJN     ABT
        }

ABT
```

```
                                  *        R.TFL     COMPARES ACCUMULATOR TO FIELD LENGTH              STL       391
                                  *                                                                    STL       392
                                  *        CALLING SEQUENCE                                            STL       393
                                  *                                                                    STL       394
                                  *        LOAD      VALUE                                             STL       395
                                  *        RJM       R.TFL                                             STL       396
                                  *                                                                    STL       397
                                  *        RETURNS                                                     STL       398
                                  *                                                                    STL       399
                                  *        ACCUMULATOR=VALUE+RA IF VALUE IS LESS THAN FIELD LENGTH     STL       400
                                  *        ACCUMULATOR=-0  OTHERWISE                                   STL       401
                                  *                                                                    STL       402


      0500      1014              TFL0     SHN       12                                                STL       404
      0501      3156                       ADD       D.FL              ADD FIELD LENGTH                STL       405
      0502      3155                       ADD       D.RA              ADD RA                          STL       406
      0503      1006                       SHN       6                 REPOSITION VALUE                STL       407
                                                                                                       STL       408
      0504                        R.TFL    ENM       X                                                 STL       409
      0506      0705                       MJN       TFL1              JUMP IF NEGATIVE                STL       410
      0507      1014                       SHN       12                POSITION TO HUNDREDS            STL       411
      0510      3256                       SBD       D.FL                                              STL       412
      0511      1006                       SHN       6                                                 STL       413
      0512      0765                       MJN       TFL0              SENSE IN RANGE                  STL       414
      0513      1500              TFL1     LCN       0                 ERROR RETURN                    STL       415
      0514      0367                       UJN       R.TFLX                                            STL       416
```

R.DFM

SCOPE 3.3

R.DFM {664}

Calling Sequence: LOAD L{message}+flag bits

RJM R.DFM

R.DFM will cause a message to be written from PP memory to the dayfile and/or the console. The flag bits are contained in the high-order 6-bits of the A register upon entry to R.DFM and are used to determine the destinations of the message. Possible values of the flag bits are described below; one or more bits may be on. All are optional.

clear these
1 = Dayfile only {B Display}
2 = Control Point 0 {System} message
4 = System Dayfile {No A Display}

set these
10B = Collation {Accounting} Flag. If set then a $ will be placed in the 20th character of messages that are sent to the system dayfile {not set by any CDC routine. Used by installations}.
20B = C.E. Error File.

---

on entry

(A) =	flag bits	msg buf addr

default 0 sends msg to dayfiles & B display

R.DFM moves the message to the PP Message Buffer in CM (PP Comm. Area) & then uses MTR function M.DFM to issue the message. I in first word of buffer makes message flash

```
                                              *        R.DFM      TRANSMIT DAYFILE MESSAGE                                          STL        580
                                              *                                                                                    STL        581
                                              *        CALLING SEQUENCE                                                            STL        582
                                              *                                                                                    STL        583
                                              *        LOAD       (FLAG)LOCATION OF MESSAGE                                        STL        584
                                              *        RJM        R.DFM                                                            STL        585
                                              *                                                                                    STL        586
                                              *        ACTIONS                                                                     STL        587
                                              *                                                                                    STL        588
                                              *        TRANSMIT MESSAGE TO PP MESSAGE AREA                                         STL        589
                                              *        CALL MONITOR FUNCTION M.DFM,(FLAG)                                          STL        590
                                              *                                                                                    STL        591

          0564                          R.DFM      ENM        X                                                                    STL        593
          0665        3410                         STD        D.T0            LOCATION OF MESSAGE                                 STL        594
          0667        1063                         SHN        -12                                                                  STL        595
          0670        3411                         STD        D.T1            STORE FLAG                                           STL        596
          0671        3075                         LDD        D.PPMES1                                                             STL        597
          0672        3412                         STD        D.T2            SET STORAGE (-1) ADDRESS                             STL        598
          0673        1413                  DFM2   LDN        D.T3                                                                 STL        599
          0674        3400                         STD        0               SET ASSEMBLY ADDRESS                                 STL        600
          0675        4010                  DFM1   LDI        D.T0            MOVE BYTE                                            STL        601
          0676        4400                         STI        0               TO ASSEMBLY AREA                                     STL        602
          0677        0402                         ZJN        *+2             SENSE END OF MESSAGE                                 STL        603
          0700        3610                         AOD        D.T0            ADVANCE IN MESSAGE                                   STL        604
          0701        3600                         AOD        0               AND ASSEMBLY AREA                                    STL        605
          0702        1125                         LMN        D.T3+5                                                               STL        606
          0703        0571                         NJN        DFM1            SENSE ASSEMBLY NOT FULL                              STL        607
          0704        3012                         LDD        D.T2                                                                 STL        608
          0705        6213                         CWD        D.T3            WRITE ASSEMBLY TO MESSAGE AREA                       STL        609
          0706        3612                         AOD        D.T2            ADVANCE STORAGE ADDRESS                              STL        610
          0707        1207                         LPN        7                                                                    STL        611
          0710        0403                         ZJN        DFM3            JUMP IF END OF MESSAGE AREA                          STL        612
          0711        3017                         LDD        D.T3+4                                                               STL        613
          0712        0560                         NJN        DFM2            LOOP IF NOT END OF MESSAGE AREA                      STL        614
                                                                                                                                   STL        615
          0713        1401                  DFM3   LDN        M.DFM                                                                STL        616
          0714        0200 0516                    RJM     R.MTR              SEND DAYFILE MESSAGE                                 STL        617
          0716        0345                         UJN        R.DFMX                                                               STL        618
```

8.32

R.DFM

[illegible]

R.READP/R.WRITEP

SCOPE
3.3

R.READP {R.WRITEP} {273} {302}

Calling Sequence: Load L{request}
RJM R.READP{R.WRITEP}

When a PP program wishes to issue a stack request for a transfer of data to or from its own memory, the PP program formats the stack request, loads the address of the request into the A Register and calls the PP Resident subroutine R.READP/R.WRITEP. There are two entry points to this subroutine. If the stack request is to read data the entry point R.READP is used. R.WRITEP is used when writing data.

R.READP {R.WRITEP} computes the PP word count from the first and last word addresses given in the already formatted request and adds the computed word count, the address of the PP message buffer, and the control point number to the request. The request is entered in the stack and data is transmitted via channel directly to {from} PP memory. Upon exit from R.READP {R.WRITEP}, the following information will be set:

{D.T3 + C.RWPPWT} = number of PP words transmitted
{D.T3 + C.RWPPLW} = LWA+1 of data transmitted
{D.T3 + C.RWPPST} = upper six bits of status in bits 0-5
{D.T4 + C.RWPPST} = lower twelve bits of status

The 18-bit status has the same format and meaning for PP I/O as the status in bits 0-17 of the first FET word for central memory I/O.

R.READP/R.WRITEP will call R.EREQS to issue the stack request. It then helps control the transfer of data by communicating with stack processor.

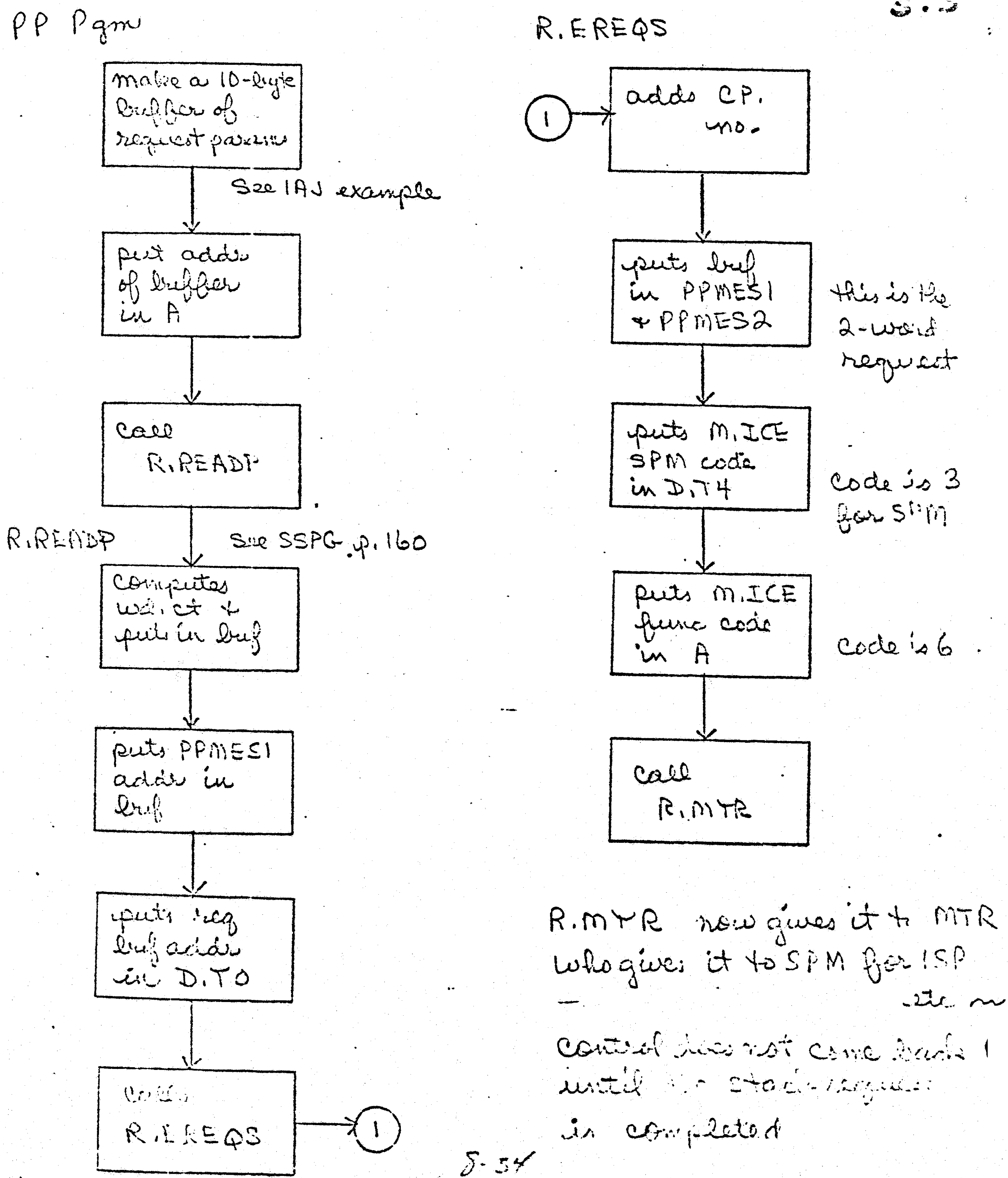
See the Scope Systems Programmers Guide page 136A for a good write up on how to format a request stack entry

How to Initiate a Stack Request

Example: a PP Program Inputting to a PP

R.READP/R.WRITEP

SC. 3
3.3



R.MTR now gives it to MTR who gives it to SPM for ISP

etc

control does not come back until the stack request is completed

Example: IAJ reads a pru of control cards

```
                *                                                                  1AJ     2119
                *          READ A PRU OF CONTROL CARDS                             1AJ     2120
                *                                                                  1AJ     2121
                           }
                *                                                                  1AJ     2138
2000 4704                  LDC     RQ                                              1AJ     2139
0200 0274                  RJM     R.READP             REQUEST PP READ             1AJ     2140
                *                                                                  1AJ     2141
                           }
```

IAJ puts FST here

order code 10 is Read

```
0000          RQ           DATA    0,0,0               FST ADDRESS WILL BE INSERTED                 1AJ     2194
0010                       VFD     12/0.RDP            PP READ ORDER                                1AJ     2195
0001                       VFD     12/1 <-- DST#1      UNIT                                         1AJ     2196
0000                       DATA    0,0                 SET BY R.READP (wd.ct., PPMESI addr)         1AJ     2197
0051                       VFD     12/BUFFER           FWA OF PP BUFFER                             1AJ     2198
0400                       VFD     12/NOFET            NO FET BIT                                   1AJ     2199
0551                       VFD     12/BUFFER+64*5      LWA OF READ AREA                             1AJ     2200
                           }
```

set by caller (IAJ)

	FST address		code 10 for Read	DST #1
word count	PPMESI address	FWA of PP buffer	no fet bit	LWA of PP buffer

← 64 CM words *5 for PP words

set by R.READP

96-38

```
                                  *           R.READP (R.WRITEP)        TRANSMIT DATA VIA CHANNEL FROM (TO)   STL        205
                                  *                                     STACK PROCESSOR                        STL        206
                                  *                                                                            STL        207
                                  *           CALLING SEQUENCE                                                 STL        208
                                  *                                                                            STL        209
                                  *           LOAD      L(REQUEST)                                             STL        210
                                  *           RJM       R.READP (R.WRITEP)                                     STL        211
                                  *                                                                            STL        212
                                  *           ACTIONS                                                          STL        213
                                  *                                                                            STL        214
                                  *           ADD PP MESSAGE AREA ADDRESS TO REQUEST                           STL        215
                                  *           ISSUE STACK REQUEST                                              STL        216
                                  *           TRANSMIT DATA VIA CHANNEL TO (FROM) PP MEMORY                    STL        217
                                  *                                                                            STL        218
                                  *           RETURNS                                                          STL        219
                                  *                                                                            STL        220
                                  *           (D.T3+C.RWPPLW)=LWA+1 OF DATA TRANSMITTED                        STL        221
                                  *           (D.T3+C.RWPPST,D.T4+C.RWPPST)=TERMINAL STATUS OF FILE            STL        222
                                  *           (D.T3+C.RWPPWT)=NUMBER OF PP WORDS TRANSMITTED                   STL        223
                                  *                                                                            STL        224

 0273                    R.READP  ENM       X                                                                  STL        226
 0275     3410                    STD       D.T0            SAVE REQUEST LOCATION                              STL        227
 0276     1471                    LDN       71B             READ                                               STL        228
 0277     0307                    UJN       RCOM                                                               STL        229
 0300     1012           RCOMA    SHN       10                                                                 STL        230
 0301     0671                    PJN       R.READPX        EXIT IF IAM                                        STL        231
                                                                                                               STL        232
 0302                    R.WRITEP ENM                                                                          STL        233
 0304     3410                    STD       D.T0                                                               STL        234
 0305     1473                    LDN       73B             WRITE                                              STL        235
 0306     0200 0320      RCOM     RJM     R.RWP             HANDLES READ/WRITE LOGIC                           STL        236
 0310     3613                    AOD       D.T3+C.RWPPCF       RELEASE 1SP                                    STL        237
 0311     3075                    LDD       D.PPMES1                                                           STL        238
 0312     1602                    ADN       W.RWPPCW                                                           STL        239
 0313     6213                    CWD       D.T3                                                               STL        240
 0314     5000 0370               LDM       RWPIOT          IAM OR OAM                                         STL        241
 0315     0361                    UJN       RCOMA                                                              STL        242
                                                                                                               STL        243
                                                                                                               STL        244
                                                                                                               STL        245
 0317                    R.RWP    ENM       X                                                                  STL        246
 0321     1006                    SHN       6               POSITION IAM/OAM FUNCTION                          STL        246
 0322     5400 0370               STM       RWPIOT          SET IAM OR OAM                                     STL        247
 0324     5010 0007               LDM       5*W.STPFW+C.STPFW,D.T0                                             STL        248
 0326     5400 0371               STM       RWPIOA          SET FWA FOR TRANSMISSION                           STL        249
 0330     3075                    LDD       D.PPMES1                                                           STL        250
 0331     5410 0006               STM       5*W.STPMS+C.STPMS,D.T0       ADD MESSAGE AREA ADDRES               STL        251
 0333     0200 0407               RJM     R.FREQS           ENTER REQUEST IN STACK                             STL        252
            0000334      R.RWPP   EQU       *-1             CHANGED BY LDR                                     STL        253
                                                                                                               STL        254
 0335     5000 0100      RWPP     LDM       R.FAF           IF FIELD ACCESS FLAG IS SET                        STL        255
 0337     0403                    ZJN       RWPD                                                               STL        256
 0340     0200 0446               RJM       R.PAUSE         PAUSE FOR STORAGE RELOCATION                       STL        257
 0342     1442           RWPD     LDN       RWDELAY/2       DELAY RWDELAY MICROSECONDS                         STL        258
                                                                                                               STL
```

```
0344      0576                         NJN       *-1                                      STL       260
0345      3075           RWPL          LDD       D.PPMES1                                 STL       261
0346      1602                         ADN       W.RWPPCW                                 STL       262
0347      6013                         CRD       D.T3               READ CONTROL WORD     STL       263
0350      3013                         LDD       D.T3+C.RWPPCF                            STL       264
0351      1703                         SBN       3                                        STL       265
0352      0413                         ZJN       RWPIO              SENSE TRANSMISSION READY    STL       266
0353      0643                         PJN       R.RWPX             SENSE END OF TRANSMISSION   STL       267
0354      1602                         ADN       2                                        STL       268
0355      0757                         MJN       RWPP               SENSE STILL WAITING FOR CHANNEL   STL       269
0356      0566                         NJN       RWPL               SENSE WAITING FOR TRANSMISSION    STL       270
0357      2100 0402                    LDC       RWPSTBL                                  STL       271
0361      0200 0657                    RJM       R.STH              STORE CHANNEL NUMBER  STL       272
0363      3613                         AOD       D.T3+C.RWPPCF      RESET CONTROL FLAG = 2      STL       273
0364      0312                         UJN       RWPWF                                    STL       274
                                                                                          STL       275
0365      3017           RWPIO         LDD       D.T3+C.RWPPWC                            STL       276
0366      6500 0366      RWPIOW        IJM       *,**               WAIT FOR CHANNEL ACTIVE     STL       277
0370      2400           RWPIOT        PSN                          TRANSMIT              STL       278
0371      2400           RWPIOA        PSN                          STARTING AT THIS ADDRESS    STL       279
0372      3017                         LDD       D.T3+C.RWPPWC                            STL       280
0373      5500 0371                    RAM       RWPIOA             BUMP TRANSFER ADDRESS STL       281
0375      3713                         SOD       D.T3+C.RWPPCF      RESET CONTROL FLAG = 2      STL       282
0376      3075           RWPWF         LDD       D.PPMES1                                 STL       283
0377      1602                         ADN       W.RWPPCW                                 STL       284
0400      6213                         CWD       D.T3               RESTORE CONTROL WORD  STL       285
0401      0343                         UJN       RWPL                                     STL       286
                                                                                          STL       287
0402      0016           RWPSTBL       VFD       12/D.T3+C.RWPPCC,12/RWPIOW,12/RWPIOT     STL       288
0403      0366
0404      0370
0405      0000                         DATA      0                                        STL       289
```

R.EREQS

SCOPE 3.3

<u>R.EREQS</u> {406}

Calling Sequence: Store L{request} in D.T0

RJM R.EREQS

In order to place a request in the request stack {for the stack processor} this PP subroutine adds the control point number to the request, places a request in the message area, and issues an M.ICE function for SPM.

```
                                     *      R.EREQS   ENTER REQUEST STACK                              STL       291
                                     *                                                                 STL       292
                                     *      CALLING SEQUENCE                                           STL       293
                                     *                                                                 STL       294
                                     *      STORE     L(REQUEST) IN D.TO                               STL       295
                                     *      RJM       R.EREQS                                          STL       296
                                     *                                                                 STL       297
                                     *      ACTIONS                                                    STL       298
                                     *                                                                 STL       299
                                     *      ADD CONTROL POINT NUMBER TO REQUEST                        STL       300
                                     *      PLACE REQUEST IN MESSAGE AREA                              STL       301
                                     *      ISSUE A M.ICE FUNCTION FOR SPM                             STL       302
                                     *                                                                 STL       303

      0406                          R.EREQS  ENM       X                                                STL       305
      0410      3010                         LDD       D.TO                                             STL       306
      0411      5400 0425                    STM       ENTRSTKW+1      SAVE LOCATION OF REQUEST         STL       307
      0413      3076                         LDD       D.CPAD                                           STL       308
      0414      1001                         SHN       1                                                STL       309
      0415      5410 0004                    STM       5*W.STCPU+C.STCPU,D.TO                           STL       310
      0417      1402                         LDN       2                                                STL       311
      0420      3410                         STD       D.TO                                             STL       312
      0421      1400                         LDN       P.ZERO                                           STL       313
      0422      6011                         CRD       D.T1                                             STL       314
      0423      3075                         LDD       D.PPMES1                                         STL       315
      0424      6310 0000           ENTRSTKW CWM       **,D.TO         PLACE REQUEST IN MESSAGE AREA    STL       316
      0426                                   TESTZ     ADN,(W.RWPPCW-2)   POINT TO COMMUNICATIONS WORD  STL       317
      0426      6211                         CWD       D.T1            CLEAR IT (FOR READP/WRITEP)      STL       318
                                                                                                        STL       319
      0427      1403                         LDN       X.SPM           SPM EXECUTIVE CODE               STL       320
      0430      3414                         STD       D.T4            STORE CENTRAL EXECUTIVE CODE     STL       321
      0431      1406                         LDN       M.ICE                                            STL       322
      0432      0200 0516                    RJM      R.MTR            ISSUE STACK REQUEST              STL       323
      0434                                   UJK       R.EREQSX        EXIT                             STL       324
```

8-39

loc. of request

PERIPHERAL PROCESSOR RESIDENT

COMMUNICATION CONVENTIONS

SCOPE 3.4

INTRODUCTION

In the SCOPE operating System, the System Display program {DSD} and the Monitor program {MTR} permanently reside in two of the peripheral processors, 1 and 0 respectively. The remaining processors form a pool of processors to which MTR may assign tasks as required. These pool processors have no fixed assignments; any processor may be assigned to the execution of any system routine, and it is possible that more than one processor may be executing the same routine at the same time. All processors contain a small resident program which handles the communications between pool processor programs and the Monitor and initiates the execution of these programs as directed by MTR.

When SCOPE is deadstarted a series of deadstart PP programs are loaded into the PP's. The last deadstart program to be loaded is named STL. It is loaded at location 100_8 in each of the pool PP's. The program STL contains PP Resident. STL starts executing at location 1000_8. When it is done it jumps to the PP Resident Idle loop, R.IDLE {see below}, and the PP is ready to load and run programs as directed by MTR.

POOL PROCESSOR STRUCTURE

PP resident is contained in locations 0103_8 - 0772_8. When directed to do so by MTR, the resident loads a program into its memory and executes it; since that program remains in that processor only for the period of time required to perform its function, it is called a transient program. Transient programs occupy locations 0773B - 1772B, although the first instruction is at location 1000B. Transient programs generally load overlays to perform specific tasks. For example, CIO, which is a transient program, calls various overlays depending on the task {read, write, backspace} and the equipment {disk, tape, etc.} specified. Secondary overlays are loaded into memory beginning at location 1773B, the first instruction falling at location 2000_8. Overlays are generally entered via a return jump. Transient programs have names beginning with a letter {CIO, EXU} or the numeral 1 {1AJ, 1IQ}; overlays have names beginning with a numeral 2 through 9 {2BP, 4LB, 9DM, etc.}.

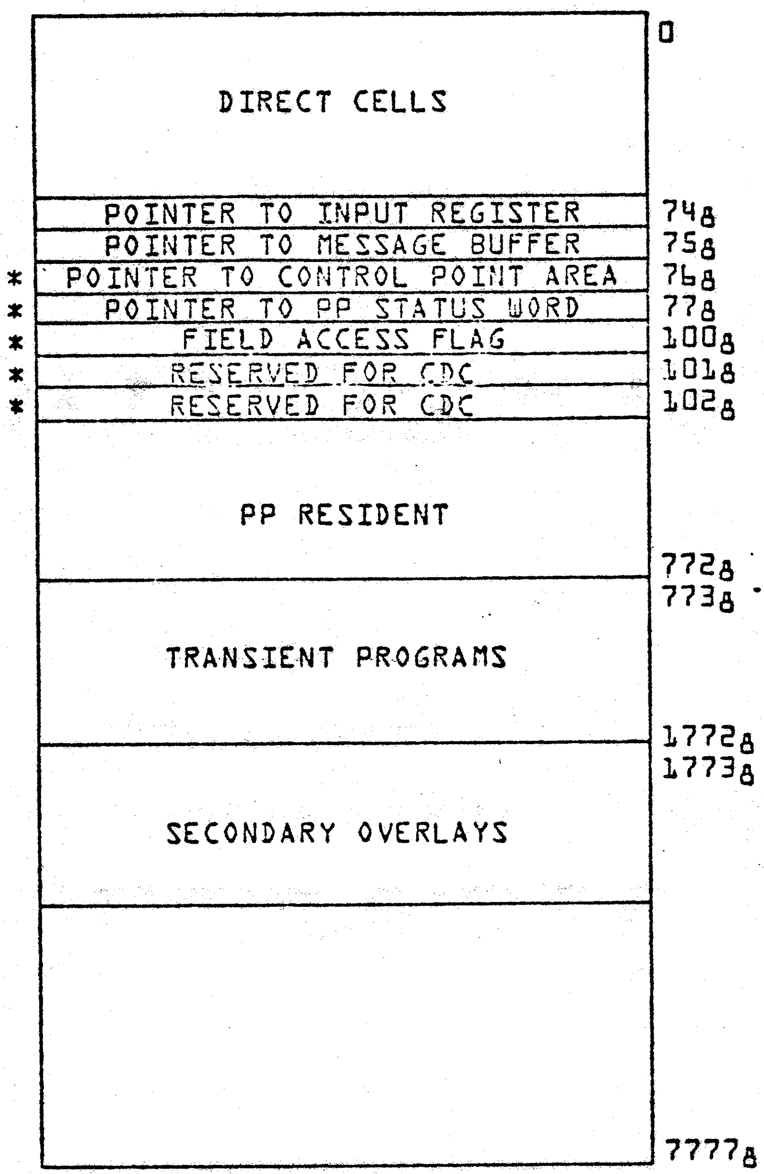
Both transient and overlay programs, as well as the resident program, make extensive use of the low core locations 01-73. Figure 2 details these direct cell assignments.

PP RESIDENT

The peripheral processor resident program has two main functions to perform:

1. All communication between MTR and the transient or overlay programs is handled by the resident.

SCOPE 3.4



* Cells 76-102$_8$ constitute the five bytes of the PPU's central memory status word.

Pool PP Layout

SCOPE
3.4

	DIRECT CELL		
*D.Za	00		SCRATCH AREA
	01		
	02		
	03		
	04		
	05		
	06		
	07		
*D.Tb	10		
	11		
	12		
	13		
	14		
	15		
	16		
	17		
*D.THc	20	D.FNT	WORDS 2 AND 3 OF FNT/FST ENTRY FOR CURRENT FILE
	21	D.FNT+1	
	22	D.FNT+2	
	23	D.FNT+3	
	24	D.FNT+4	
	25	D.FNT+5	
	26	D.FNT+6	
	27	D.FNT+7	
*D.THd	30	D.FNT+8	
	31	D.FNT+9	
	32	D.EST	EST ENTRY FOR DEVICE BEING USED BY THIS PPU
	33	D.EST+1	
	34	D.EST+2	
	35	D.EST+3	
	36	D.EST+4	
	37	D.DTS	DEVICE TYPE AND ALLOCATION TYPE OR D.JFL JOB CM FL
*D.FRe	40	D.BA	FET ADDRESS RELATIVE TO LOWER 18 BITS OF INPUT REGISTER
	41	D.BA+1	
	42	D.BA+2	
	43	D.BA+3	
	44	D.BA+4	
	45	D.JECS	ECS FIELD LENGTH REQUIREMENT
	46	D.JPR	COMPUTED PRIORITY
	47	D.JTL	JOB TIME LIMIT/10B
*D.FFf	50	D.PPIRB	CONTENTS OF PPU INPUT REGISTER
	51	D.PPIRB+1	
	52	D.PPIRB+2	
	53	D.PPIRB+3	
	54	D.PPIRB+4	
	55	D.RA	REFERENCE ADDRESS/100B OF USER'S AREA
	56	D.FL	FIELD LENGTH/100B OF USER'S AREA
	57	D.FA	ADDRESS OF FILE FST ADDRESS
*D.SXg	60	D.FIRST	CIRCULAR BUFFER
	61	D.FIRST+1	FIRST PARAMETER FROM FET
	62	D.IN	CIRCULAR BUFFER
	63	D.IN+1	IN PARAMETER FROM FET
	64	D.OUT	CIRCULAR BUFFER
	65	D.OUT+1	OUT PARAMETER FROM FET
	66	D.LIMIT	CIRCULAR BUFFER
	67	D.LIMIT+1	LIMIT PARAMETER FROM FET
*D.SVh	70	D.PPONE	THE CONSTANT "1"
	71	D.HN	THE CONSTANT "100B"
	72	D.TH	THE CONSTANT "1000B"
	73	D.TR	THE CONSTANT "3"
	74	D.PPIR	ADDRESS OF INPUT REGISTER
	75	D.PPMESI	ADDRESS OF PPU MESSAGE BUFFER
	76	D.CPAD	ADDRESS OF CONTROL POINT
	77	D.PPSTAT	ADDRESS OF PPU STATUS WORD

*THE LOWER CASE LETTERS REPRESENT THE LEAST SIGNIFICANT DIGIT OF THE CELL NUMBER IN EACH CASE

Direct Cells

SCOPE
3.4

2. The resident, when directed by MTR, loads transient programs and initiates the execution of these programs.

Communication between MTR and the resident program is carried out through the use of PP communication areas in central memory, one for each processor. Each communication area consists of a one-word Input Register, a one-word Output Register, and a six-word Message Buffer. Pool processors address these areas by means of pointers in locations D.PPIR, D.PPMES1 and D.PPSTAT.

MTR assigns a task to a pool processor by placing the request in the processor's Input Register. The name of the program package which is to be loaded and executed appears in the high-order 18 bits of the Input Register. This name consists of three display code characters, such as 1AJ, CIO, etc. The number of the control point to which this package is assigned appears in the low-order four bits of byte 1 of the Input Register. Package parameters, such as the address of arguments required by the package, appear in the low-order 36 bits of the Input Register. The PP is given control to execute the code just loaded. The request itself remains in the Input Register until the task is completed. On completion of a task, the transient program requests MTR to release the processor; MTR then clears the processor's Input Register. The Input Register of a pool processor is thus clear only when the processor is idle.

All communication between the Monitor and the transient and overlay programs is handled by the resident program. MTR performs a variety of functions, each of which is identified by a function code of one or two octal digits.

To transmit a monitor request, the resident routine R.MTR places the request in the PPs output register. R.MTR uses locations D.T0-D.T4 in peripheral processor memory as temporary storage for the request to be written into the output register. Byte 0 of the register contains the function code in the low-order bit positions. Bytes 1-4 are used for the arguments. Thus, for a Request Channel function {M.RCH=12}, the channel number is placed in bite 1. For some functions, the arguments are placed in the message buffer and only the function code appears in the output register. A peripheral processor program may utilize the routine by placing the arguments for the function in bytes D.T1 through D.T4, setting the A register function number and executing a return jump to R.MTR. The resident routine will enter the function number in location D.T0 and write the contents of locations D.T0-D.T4 into the output register. R.MTR will jump to R.WAIT.

If the system is using the XJ/MXN or MAN, R.WAIT will decide whether the monitor request is for CPMTR or MTR. If the request is for CPMTR, the PP input register address is written into word 47 of CMR,T.PPID. CPMTR requests are the first ten functions {12B} unless the ILR is used in which case CPMTR executes the first nine {11B} functions leaving M.RCH{12B} to be handled by MTR.

If the request in for MTR, the input register address is written into word 50 of CMR,T.PPIP. MTR executes all functions greater 128. The use of T.PPIP saves MTR having to search through all of the output registers. MTR need only check one word in CMR to know if a request is pending. PP resident will issue an MXN or MAN to initiate CPMTR if needed. Otherwise the request will be picked up by MTR in its loop. On an EXN system, only T.PPIP is used by PP resident. Regardless of mode of execution, the resident will wait until the output register is cleared by MTR before proceeding. Control will be returned to the requesting program upon MTR's clearing the output register.

When a pool processor program completes execution, it exits to location R.IDLE, which is the address of the resident idle loop. The entry point to this idle loop is R.IDLE {103_8}. When referring to a PP Resident routine, the name of its entry point is used as the name of the routine. Thus the name of the idle loop is R.IDLE. In this idle loop, the processor's Input Register is scanned at intervals until a request is found in the Input Register. A delay between successive scans avoids unnecessary memory and read pyramid conflicts. Normally the PP Input Register of an idle PP contains zero. If the PP Input Register becomes non-zero, it means MTR wants PP Resident to load a PP transient program into the PP. When a request is detected, the resident stores the routine name and the control point. It then sends function R.TAFL, terminates access to the control point field length, to MTR and waits for MTR to clear the Output Register before continuing. When the Output Register is cleared R.IDLE calls the PP Resident subroutine R.OVL. R.OVL will then search the library directory for the requested routine; if found, the package is read from the resident library into the processor's memory beginning at location 773_8 {C.PPFWA-L.PPHDR}. If the routine is not found in the directory, 1EJ enters the message "XXX NOT IN PPLIB" in the dayfile, and requests MTR to abort the job which called the routine. The resident then returns to its idle loop. If the program is located, it is loaded by R.OVL which then returns control to R.IDLE which executes the instruction

```
LJM C.PPFWA
```

This transfers control to the first instruction of the transient program. When a transient program terminates, the instruction it must execute is

```
LJM R.IDLE
```

At location 100B of PP resident is the field access flag. There is evidently some need for a better description of the use of the field access flag. The basic principles are:

1. The field access flag must always be set whenever any data is read or written within a user field length.

2. The execution of the R.MTR subroutine while the field access flag is set may cause R.PAUSE to be executed. If an absolute address has been computed and saved, it is invalidated by the execution of R.MTR, because D.RA may change.

3. When a PP program is looping waiting for any external event to occur, the loop must either be performed while the field access flag is not set or must include an execution of R.PAUSE.

4. When no field length access is required for a major process (i.e. searching the FNT) it is best to clear the field access flag while processing. This allows a storage move to be initiated without any delay.

R.RAFL (synonymous to R.PAUSE) will set the field access flag. If a storage move is in progress, the field access flag is not set until the storage move flag has been cleared. If the field access flag is already set, R.RAFL checks the storage move flag and temporarily clears the field access flag to allow the storage move.

R.TAFL is used to terminate access to the field length. It unconditionally clears the field access flag.

RESIDENT ROUTINES

Several resident routines and words are used by transient and overlay programs. These routines are described below. The order of the routines has been changed but essentially the function of each routine remains the same with the exception of R.OVL, R.RCH and R.DCH. These three routines have been conditionally modified to accommodate the Distributive Data Path (DDP) and Interlock Register (ILR) in CYBER 70 hardware. Should an installation not use these two hardware features, PPRES is functionally unchanged.

In the diagram of PPRES the labels MAIN, SEG-1 and MAIN2 refer to the segments described at the discussion of the DDP/ILR at the end of this section.

SC[illegible]
S.[illegible]

PP RESIDENT ROUTINES

MAIN			103
	R.IDLE	PP RESIDENT IDLE LOOP	
	R.OVLJ	LOAD PRIMARY OVERLAY INTERNALLY	125
	R.RAFL	REQUEST ACCESS TO CONTROL POINT FIELD LENGTH	133
	R.TAFL	TERMINATE ACCESS TO CONTROL POINT FIELD LENGTH	172
	R.TFL	COMPARE ACCUMULATOR TO FIELD LENGTH	202
	R.MTR	ISSUE MONITOR FUNCTION	217
	R.WAIT	WAIT FOR OUTPUT REGISTER TO CLEAR	230
	R.RCH	RESERVE CHANNEL	271
	R.DCH	DROP CHANNEL	322
	R.STB	MASK BYTE INTO LISTED WORDS	343
SEG-1	R.OVL	LOAD PP OVERLAY	362
	R.READP	TRANSMITS DATA FROM STACK PROCESSOR	542
	R.WRITEP	TRANSMITS DATA TO STACK PROCESSOR	551
	R.RWP	SUPPLIES DISK READ/WRITE LOGIC	566
	R.EREQS	ENTER REQUEST STACK	655
	R.DFM	TRANSMIT DAYFILE MESSAGE	704
MAIN2		CONDITIONAL CODE	737

.. R.IDLE

Calling Sequence: LJM R.IDLE

R.IDLE is the idle loop in which PP resident continually scans its input register for something to do.

.. R.OVLJ

Calling Sequence: Store name of overlay in D.T6, D.T7.
LJM R.OVLJ

The R.IDLE routine contains an additional entry point named R.OVLJ. A PP program can load a transient program on top of itself without changing the Input Register by storing the name of the transient program left-justified in D.T6 and D.T7, and then executing a long jump to R.OVLJ. The program will be loaded at C.PPFWA - L.PPHDR and control will be transferred to location C.PPFWA.

R.IDLE destroys direct cells 20_8 through 22_8 and some of the temporaries. R.OVLJ and all other PP Resident routines destroy only temporary cells {0 - 17_8}.

.. R.RAFL

Calling Sequence: RJM R.RAFL

The subroutine is called to request access to the control point field length. A test is made on the storage move flag for the control point. If set, a call is made to R.TAFL to clear the field access flag in the PP status word, then pauses until the storage move is cleared. When it is cleared, set the field access flag in the PP status word and reset RA in D.RA, FL in D.FL.

.. R.TAFL

Calling Sequence: RJM R.TAFL

R.TAFL is called to terminate access to the control point field length by clearing the field access flag in CM byte R.FAF.

.. R.TFL

Calling Sequence: Load relative address

RJM R.TFL

R.TFL is used to insure that a relative address is within the field length. The 18-bit address is added to the control point reference address {RA} and compared with the field length. If the address is out of range, R.TFL will exit with a negative A register; if the address is legal, the A register will contain the

SCOPE
3.4

absolute CM address {RA + relative address} upon exit. The control point RA and FL are kept locally within PP resident at D.RA and D.FL, respectively. Since these locations are set by routine R.RAFL, the transient program and its overlays cannot call R.TFL until R.RAFL has been called. Many PP programs do not call R.TFL but do their own checking of addresses.

.. R.MTR

Calling Sequence: Store function parameters in D.T1 to D.T4

Load function code

```
RJM  R.MTR
```

The PP resident subroutine R.MTR is called by PP transient programs and overlays to transmit requests to MTR. The requesting PP program sets direct cells D.T1 - D.T4 with the values it wants to be put into the four right-most bytes of the PP Output Register. The requesting program then loads the MTR function code into the A Register and executes a return jump to R.MTR. R.MTR stores the function code value from the A Register into cell D.T0 and then writes D.T0 - D.T4 to the Output Register. R.MTR then executes a return jump to the R.WAIT subroutine. R.WAIT checks the leftmost byte of the Output Register at a fixed interval. When the byte becomes zero {meaning that MTR has processed the request}, R.WAIT returns to R.MTR which returns to the calling routine.

In order to check byte zero of the PP Output Register, R.WAIT reads the Output Register into direct cells D.T0 - D.T4. When control is returned to the PP routine which called R.MTR, these direct cells are intact {i.e., they contain the value of the Output Register read by R.WAIT}. For certain MTR requests, MTR will return parameters to the requesting PP in bytes one through four of the PPs Output Register. The requesting PP routine can pick up these parameters from cells D.T1 through D.T4.

When a PP transient program has completed its function, it must inform MTR, so that MTR can assign a new task to the PP. The program tells MTR it has finished by issuing an M.DPP function. MTR will zero the input register of the PP and record the fact that the PP is available. The last few lines of code of each PP transient program therefore are:

```
LDN  M.DPP                     DROP THE PP ASSIGNMENT

RJM  R.MTR

LJM  R.IDLE                    EXIT TO IDLE LOOP
```

Note that R.IDLE is not a subroutine and it is entered with a long jump and not a return jump.

.. R.WAIT

Calling Sequence: RJM R.WAIT

R.WAIT has been modified for the use of two monitors and the MXN. R.WAIT is responsible for determining whether a PP request is for MTR or CPMTR. If the request is for CPMTR, the PP input register address is written into T.PPID. T.MXNCTL is read up and executed. This word contains the exchange package address to which the MXN will be issued.

If the request is for MTR, the input register address is written to T.PPIP. In either case, R.WAIT will cause the PP to idle until byte 0 of the output registers clear.

.. R.RCH

Calling Sequence: Load channel number

```
RJM    R.RCH
```

The channel numbers contained in the A-register will be stored in byte D.T1, monitor function M.RCH inserted in D.T0, and D.T0 - D.T4 written to the output register for that PP. Channels will be assigned by MTR on the following priority basis.

If alternate channels are specified MTR will stop looking for alternate channels upon sensing 6 bits of zero. Thus, if one alternate channel is desired, the programmer must clear D.T2 before entering R.RCH so the search will be terminated at that point. The procedure for requesting channel 12 with alternate channel 13 would be:

```
LDN   0

STD   D.T2

LDC   1312B

RJM   R.RCH
```

Monitor will stop looking for alternate channels after four channels have been investigated or 6 bits of zero are detected.

When R.RCH is used, D.T4 is automatically set nonzero; in this case, the function is not considered complete {i.e , output register is not cleared} until a channel can be assigned When complete, byte 0 of the output register is cleared.

.. R.DCH

Calling Sequence: LOAD channel number

RJM R.DCH

Since more than one PP can request the same channel at the same time, it is necessary to use a MTR request to reserve a channel.

The only PP which can release a channel, however, is the PP which reserved it and there is no need for an interlock. To release a channel reservation, a PP program loads the number of the channel into the A-Register and executes a return jump to the PP Resident subroutine R.DCH. If the channel is assigned to the PP, R.DCH will modify the Channel Status Table entry for the channel to indicate that the channel is free. If a PP calls R.DCH to release a channel it has not reserved, R.DCH will issue an M.KILL.

.. R.STB

Calling Sequence: Load L{List}

RJM R.STB

where list has the form

L {byte}

L {word 1}

L {word 2}

.
.
.

L {word n}

zero

An entry point to R.STB called R.STBMSK is the address of the mask ▿anded▿ with each word in the list before the word is ▿exclusive ored▿ with the byte. This mask is initially 7700B and this value should be restored by any routine which substitutes an alternate mask. R.STB is used primarily to substitute channel numbers in driver overlays.

All the PP hardware instructions used for I/O contain a field which specifies the number of the channel over which the I/O is to take place. For example the instruction

IAM BUFF,5

would be used to read data from hardware channel five into the PP starting at location BUFF.

When a programmer is coding a PP program, he normally does not know what channel will be used for the I/O. The channel number is normally obtained by the PP program from an entry in the EST table. For this reason the above I/O instruction would be written as follows:

```
IAM   BUFF,**
```

The double asterisks indicate that the value will be filled in by the program itself when it is executed. COMPASS assembles double asterisks as a zero.

Since the channel number goes into the first (or only) byte of an instruction along with the OP code, the first byte of the instruction would contain 7100_8 (the OP code for an IAM is 71_8). The second byte of the instruction would contain the value of BUFF. When the PP program is called, and determines the channel number, it must modify all the I/O instructions in itself so that the first byte of each instruction contains the OP code followed by the correct channel number. Normally there would be a list somewhere in the program giving the addresses of all instuctions to be modified in this way.

The PP resident subroutine R.STB can be called to insert a channel number into one or more instructions, whether or not the fields to be altered previously contain zero. Before return-jumping to R.STB, the program loads the address of a list in the A-register. The first byte in this list contains the address of some other PP cell that contains the new channel number. The second and following bytes of the list contain the addresses of the instruction words in which the new channel number is to be inserted. The first zero byte in the list terminates it.

Although R.STB is most often called to insert channel numbers into I/O instructions, it can also be called to perform general masking operations.

.. R.OVL

Calling Sequence: Store name of overlay in D.T6, D.T7
Load A register Load Address
RJM R.OVL

This routine has been changed for SCOPE 3.4. It now performs a binary search upon the PP Program Name Table (PPNT), looking for the name of the overlay. If the name is found, the overlay is loaded from CM, disk, or ECS. If it is not found, an OVL error flag is set and the control point is aborted. Then an exit is made to R.IDLE.

Calls - R.READP {Disk resident overlay}
R.MTR {PP Call Error}

.. R.READP {R.WRITEP}

Calling Sequence: Load L {request}

RJM R.READP {R.WRITEP}

When a PP program wishes to issue a stack request for a transfer of data to or from its own memory, the PP program formats the stack request, loads the address of the request into the A-Register and calls the PP Resident subroutine R.READP/R.WRITEP. There are two entry points to this subroutine. If the stack request is to read data the entry point R.READP is used. R.WRITEP is used when writing data.

R.READP {R.WRITEP} computes the PP word count from the first and last word addresses given in the already formatted request and adds the computed word count, the address of the PP message buffer, and the control point number to the request. The request is entered in the stack and data is transmitted via channel directly to {from} PP memory. Upon exit from R.READP {R.WRITEP}, the following information will be set:

{D.T3 + C.RWPPWT} = number of PP words transmitted
{S.T3 + C.RWPPLW} = LWA+1 of data transmitted
{D.T3 + C.RWPPST} = upper six bits of status in bits 0-5
{D.T4 + C.RWPPST} = lower twelve bits of status

The 18-bit status has the same format and meaning for PP I/O as the status in bits 0-17 of the first FET word for central memory I/O.

R.READP/R.WRITEP will call R.EREQS to issue the stack request. It then helps control the transfer of data by communicating with stack processor.

.. R.RWP

Calling Sequence: Load IAM/OAM function
71B = IAM
73B = OAM
RJM R.RWP

This routine performs a number of functions in handling the reads and writes on disk. R.RWP will set the functions for the IAM or OAM; sets the FWA for transmission; stores the PP message area address in the stack request and then issues the stack request. If the field access flag was set, R.RWP will pause for storage relocation and then perform the disk I/O. Transmission is governed by control word W.RWPPCW of the PP message area.

0 = Request is in stack
1 = Sense waiting for channel
2 = Sense waiting for transmission
3 = Sense transmission ready
4 = Sense end of transmission

Calls = R.EREQS
R.PAUSE
R.STB

.. R.EREQS

Calling Sequence: Store L{request} in D.TO

RJM R.EREQS

In order to place a request in the request stack {for the stack processor} this PP subroutine adds the control point number to the request, places a request in the message area, and issues an M.ICE function for SPM.

.. R.DFM

Calling Sequence: LOAD L{message}+flag bits

RJM R.DFM

R.DFM will cause a message to be written from PP memory to the dayfile and/or the console. The flag bits are contained in the high-order 6-bits of the A-register upon entry to R.DFM and are used to determine the destinations of the message. Possible values of the flag bits are described below; one or more bits may be on. All are optional.

Bit 0 = Do not send to B display.

Bit 1 = Do not send to control point dayfile.

Bit 2 = Do not send to system dayfile {no A display}.

Bit 3 = Flag as an accounting message {a $ will be placed in the 20th character of messages that are sent to system dayfile}.

Bit 4 = Send to hardware error file.

Bit 5 = Do not insert job name in system dayfile.

.. READECS

This is an extension to R.OVL which is entered when a load from ECS is needed.

PERIPHERAL PROCESSOR RESIDENT WITH DDP AND ILR

INTRODUCTION

This is a discussion of PP resident for those who will be using the two new features of CYBER 70 hardware, namely the Distributive Data Path {DDP} and the Interlock Register {ILR}. Complete descriptions of the hardware are found in the section called CYBER 70 Hardware Features. Likewise, most PP resident functions will remain the same and they are described earlier in this section on PP resident. The concern here is with the segmenting of PP resident and how it accomodates the DDP and ILR.

CYBER SYMBOLS

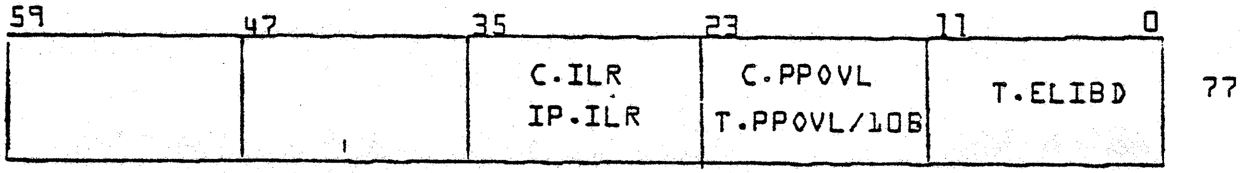
There are several parameters and symbols which are relevant to the discussion of the DDP and ILR. They are described here with their default values indicated.

IP.DPLIB = 0	If non-zero, this parameter indicates that the DDP is to be used for PP overlay loading from ECS. Certain code, such as SEG-2 and the PP resident segment loader in MAIN, is conditionally assembled with IP.DPLIB ≠ 0.
IP.ELIB = 0	Used with IP.DPLIB in the following manner: IP.ELIB = 0; no overlay loading from ECS IP.ELIB ≠ 0, IP.DPLIB = 0; perform overlay loading from ECS via CM buffer, ICEBOX. IP.ELIB ≠ 0, IP.DPLIB ≠ 0; perform overlay loading from ECS via DDP, if present. Otherwise use ICEBOX. STL makes this decision at initialization time by searching the EST. Certain code in MAIN is conditionally assembled depending upon IP.ELIB ≠ 0.
IP.ILR = 0	Indicates the presence of ILR. If no ILR exists everything is assembled with IP.ILR = 0. If ILR is present, all PP code is assembled with IP.ILR ≠ 0 but CMR is set zero or non-zero, to indicate whether or not the ILR is to be used. Word 77 of CMR {P.ILR/P.PPOVL} has byte 2 set to the value of IP.ILR. At initialization this byte is checked for the presence of the ILR. There is also a check made for a physical channel 15 If none exists PP resident will clear this byte.
CH.ILR = 15B	The ILR hangs on its own channel 15B. This channel is always active and if no ILR exists,

channel 15B is not used. The symbol CH.ILR should be used to refer to this pseudo channel.

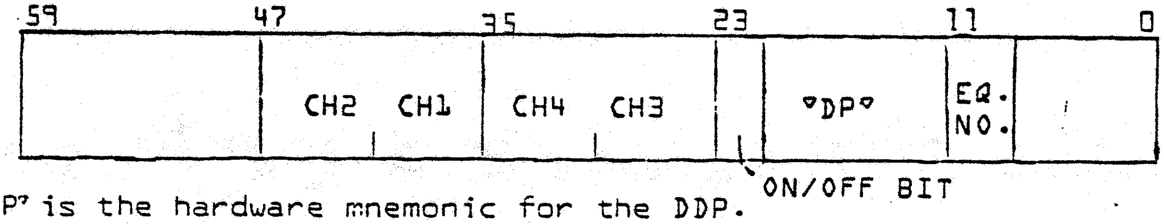
C.ILR = 2	The number of the byte in P.ILR (77) which contains the value of IP.ILR. This byte serves as the ILR on/off flag.
C.PPOVL = 3	Number of the byte in P.PPOVL (77) which contains the pointer to the CM buffer used to hold SEG-1 and SEG-2 of PPRES.
P.ILR = 77B	Word 77 of CMR which contains the ILR on/off flag.
P.PPOVL = 77B	Word 77 of CMR containing pointer to PPRES overlay buffer and address of the beginning FNT entry for the ECS library.
S.CHAN = 12D	The ILR has one bit for each channel and pseudo channel in the system, mapped from left to right. These channel numbers begin with channel zero at bit 12 (actual bit 13). Channel numbers are calculated by biasing the number by S.CHAN.

POINTER WORD - P.PPOVL/P.ILR



C.ILR/IP.ILR	- Contains value of IP.ILR. Read up by PP's to verify ILR exists.
C.PPOVL/T.PPOVL	- Table in CM holding PPRES overlays, SEG-1 and SEG-2.
T.ELIBD	- Beginning FNT entry for ECS library.

EST Entry for DDP



"DP" is the hardware mnemonic for the DDP.

SEGMENTATION OF PP RESIDENT

In order to support the DDP and ILR, considerable code had to be added to PP resident. Not desiring to expand the size of the resident area, it was decided that the most practical solution was to segment PPRES. This segmentation is transparent to the USER and, in fact, will occur only if IP.DPLIP is defined as non-zero.

The structure is as follows:

MAIN

103	R.IDLE
125	R.OVLJ
133	R.RAFL
172	R.TAFL
202	R.TFL
217	R.MTR
230	R.WAIT
271	R.RCH
322	R.DCH
342	R.STB

This segment permanently resides in PPRES beginning at location 103B. It is not overlayed.

SEG-1

362	R.OVL
542	R.READP
551	R.WRITEP
566	R.RWP
655	R.EREQS
704	R.DFM

SEG-2

364	DDP overlay loading
724	

- SEG-2 is assembled only if IP.DPLIB ≠ 0.
- R.OVL's entry point is part of MAIN. The real origin of SEG-1 is 364. This allows SEG-1 to be overlayed by SEG-2 and when SEG-2 has finished executing, return jump to R.OVL. SEG-2 is automatically overlayed by SEG-1 again, when execution is finished.

MAIN2 {conditionally assembled}

{IP.ELIB ≠ 0}

737	load from ECS via CP.ECOVL
766	

{IP.DPLIB ≠ 0}

737	PP res segment loader

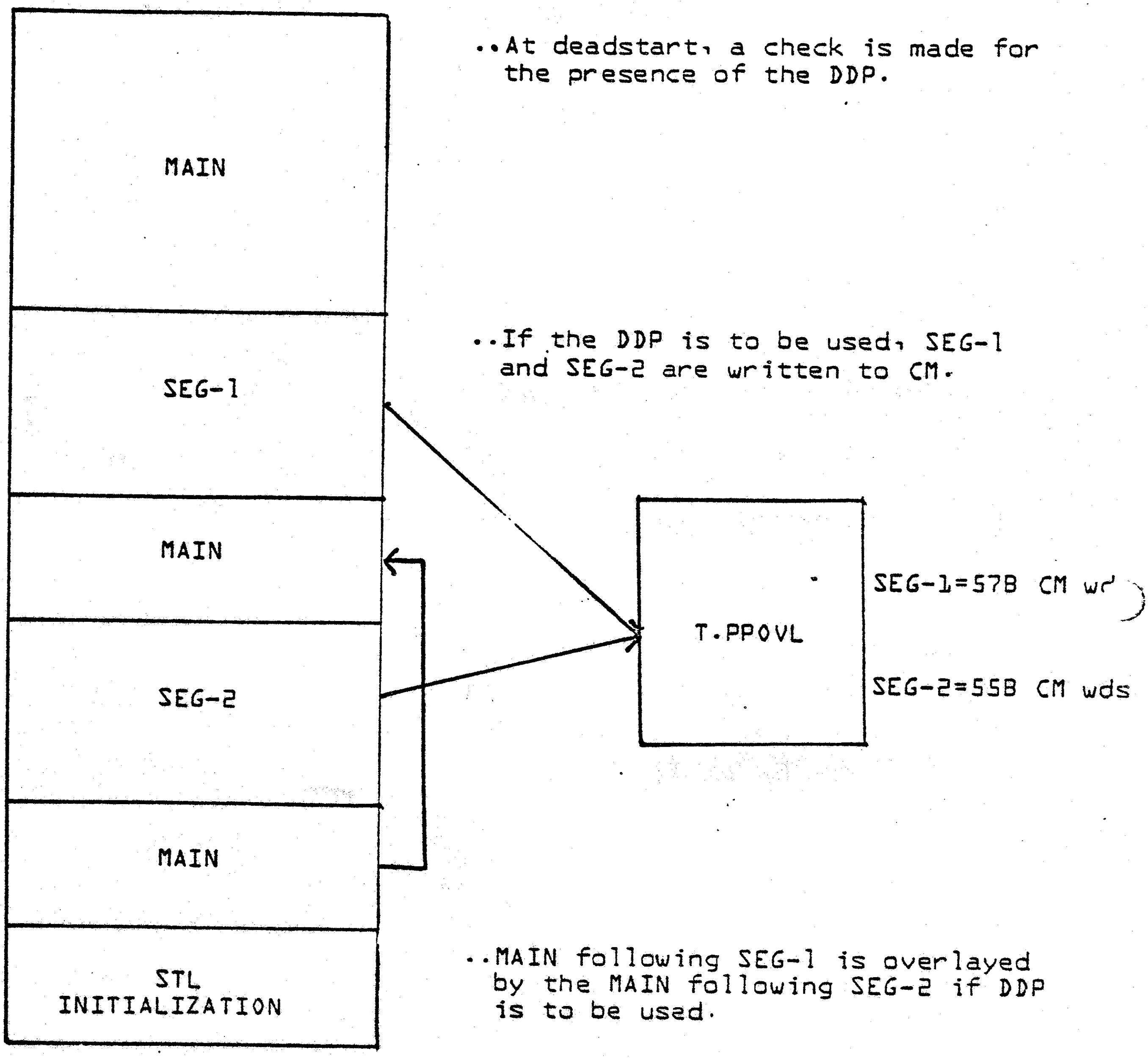
- This second section of MAIN is determined at deadstart time depending upon the setting of the IPARAMS. Once the decision is made, MAIN remains constant throughout the running of the system.

SCOPE
3.4

..At deadstart, a check is made for the presence of the DDP.

..If the DDP is to be used, SEG-1 and SEG-2 are written to CM.

..MAIN following SEG-1 is overlayed by the MAIN following SEG-2 if DDP is to be used.



PP Resident--Segmented

Sec. 3
3.4

PPRES - FUNCTIONING WITH THE DDP

If IP.DPLIB is set non-zero and the PP routines have been assembled with the DDP/ILR code, the procedure is as follows:

PP overlay loading from ECS--

- PPRES now contains MAIN-SEG-1-MAIN2.
- A request for a PP overlay is made.
- R.OVL searches the PPNT for the program name.
- When the program is found, a check for residency is made.
- If ECS resident, a jump is made to the portion of MAIN2, which contains the PPRES segment loader and load SEG-2 over SEG-1.
- SEG-2 check to see if the DDP is operational.
- If DDP available, the overlay is loaded through the DDP and then SEG-1 is reloaded.
- A jump is made to the entry point of R.OVL.
- If the DDP is not available, a jump is made to BACKUP where the residency of this overlay is changed from ECS to disk in the PP Program Name Table (PPNT).
- Then SEG-1 is reloaded, a jump is made to the entry point of R.OVL and the overlay loading process is begun again, this time from disk.

It should be noted here that the above procedure for backup is used in any case of DDP unavailability, i.e. the DDP is turned off; the hardware is dead, etc. It is also used in case of ECS parity error in which case the status word is read to check for parity or abort status. After the flaw table has been updated and an entry made in the C.E. Error File, the system will go to disk back up.

PPRES - Functioning with the ILR

The ILR is used by PPRES for channel reservations. There are several functions available in the ILR but the most commonly used functions are Test and Set, Test and Clear, and Clear.

R.RCH

If the ILR is present, R.RCH in PPRES will simply perform a Test and Set on the ILR bit corresponding to the channel to be reserved. In a channel request, PPRES will only test one time. If the R.RCH is unsuccessful (the channel bit is already set), the request must then wait and be processed by MTR in its loop. If, however, R.RCH was successful, the bit is set meaning the channel is now reserved. Then the Channel Status Table (CST) is updated to reflect the reservation.

Since, normally, channel reservation is performed by CPMTR, in the case of the ILR the number of CPMTR functions is dropped to eleven, eliminating M.RCH with code 12.

SCOPE
3.1

.. R.DCH

Dropping the channel is simply a matter of clearing the channel bit in the ILR. It is not necessary to update the CST, since channel reservation does not depend upon this information.

SECTION NINE

MONITOR FUNCTIONS

SECTION NINE - MONITOR FUNCTIONS

CONTENTS

SECTION NINE - MONITOR FUNCTIONS

CONTENTS

R.MTR {515}

R.MTR

Calling Sequence: Store function parameters in D.T1 to D.T4

Load function code

RJM R.MTR

The PP Resident subroutine R.MTR is called by PP transient programs and overlays to transmit requests to MTR. The requesting PP program sets direct cells D.T1 - D.T4 with the values it wants to be put into the four right-most bytes of the PP Output Register. The requesting program then loads the MTR function code into the A Register and executes a return jump to R.MTR. R.MTR stores the function code value from the A Register into cell D.T0 and then writes D.T0 - D.T4 to the Output Register. R.MTR then executes a return jump to the R.WAIT subroutine. R.WAIT checks the leftmost byte of the Output Register at a fixed interval. When the byte becomes zero {meaning that MTR has processed the request}, R.WAIT returns to R.MTR which returns to the calling routine.

In order to check byte zero of the PP Output Register, R.WAIT reads the Output Register into direct cells D.T0 - D.T4. When control is returned to the PP routine which called R.MTR, these direct cells are intact {i.e., they contain the value of the Output Register read by R.WAIT}. For certain MTR requests, MTR will return parameters to the requesting PP in bytes one through four of the PPs Output Register. The requesting PP routine can pick up these parameters from cells D.T1 through D.T4.

When a PP transient program has completed its function, it must inform MTR, so that MTR can assign a new task to the PP. The program tells MTR it has finished by issuing an M.DPP function. MTR will zero the input register of the PP and record the fact that the PP is available. The last few lines of code of each PP transient program therefore are:

```
        LDN  M.DPP                    DROP THE PP ASSIGNMENT

        RJM  R.MTR

        LJM  R.IDLE                   EXIT TO IDLE LOOP
```

Note that R.IDLE is <u>not</u> a subroutine and it is entered with a long jump and not a return jump.

R.WAIT {526}

R.WAIT

Calling Sequence: RJM R.WAIT

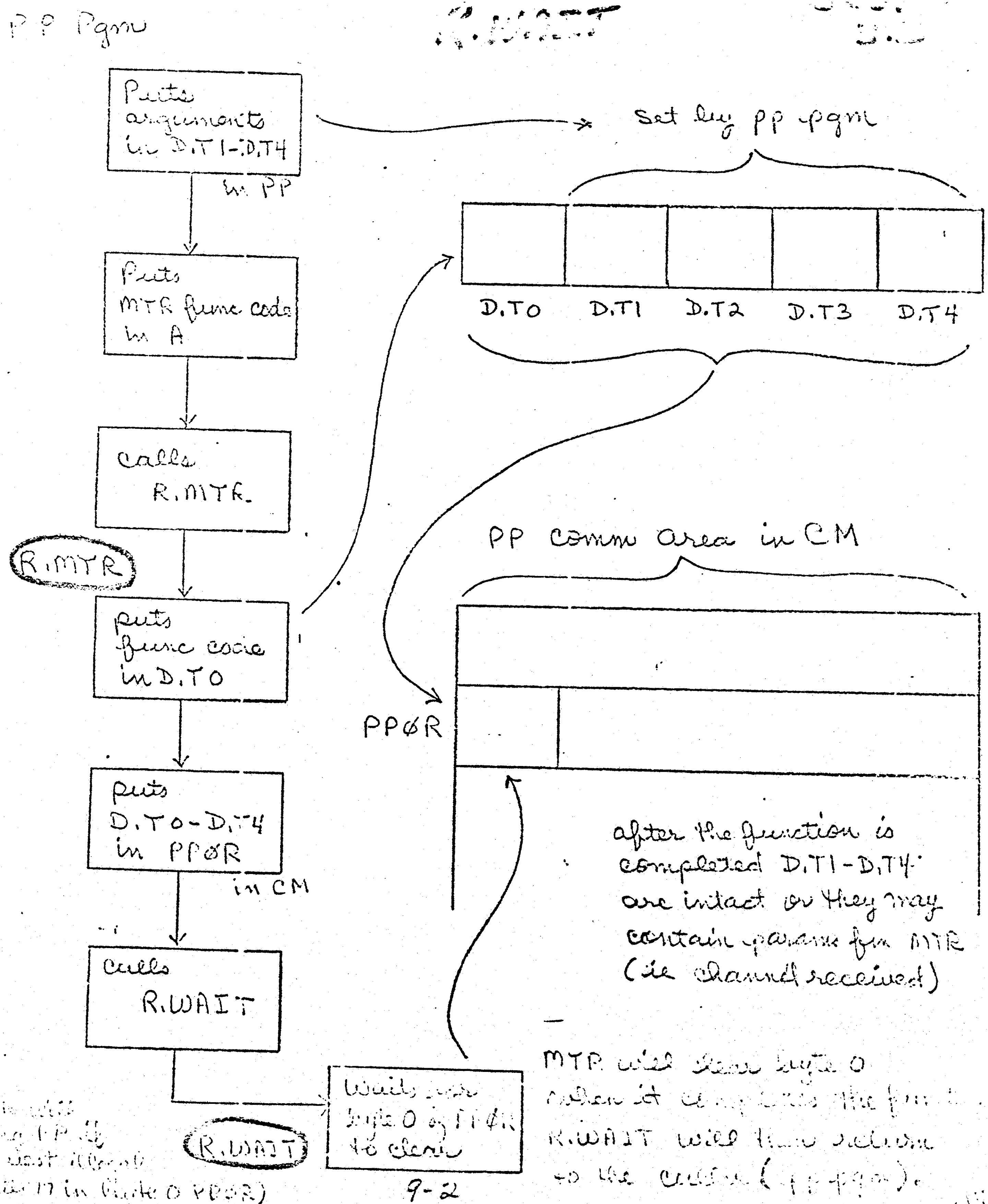
R.WAIT will cause the PP to idle until byte 0 of the output register is clear.

R.WAIT calls R.PAUSE every now & then to see if a storage move is pending while it is WAITing.

How a PP Makes a MTR Request

(See page 115a)

R.MTR &
R.WAIT



after the function is completed D.T1-D.T4 are intact or they may contain params from MTR (ie channel received)

MTR will clear byte 0 when it completes the function. R.WAIT will then return to the caller (PP pgm).

MTR will hang PP if request illegal (puts 17 in byte 0 PPØR)

9-3

```
                                    *       R.MTR         ISSUE MONITOR FUNCTION
                                    *
                                    *       CALLING SEQUENCE
                                    *
                                    *       (STORE     FUNCTION PARAMETERS IN D.T1-D.T4)
                                    *       LOAD       FUNCTION CODE
                                    *       RJM        R.MTR
                                    *
                                    *       ACTIONS
                                    *
                                    *       PLACE FUNCTION CODE IN D.T0
                                    *       WRITE FUNCTION (D.T0-D.T4) TO PP OUTPUT REGISTER
                                    *       WAIT FOR OUTPUT REGISTER TO CLEAR (FUNCTION ACCEPTED)
                                    *
                                    *       RETURNS
                                    *
                                    *       CONTENTS OF OUTPUT REGISTER IN D.T0-D.T4
                                    *


     0515                           R.MTR     ENM       X
                         0000516    R.PROCES  EQU       R.MTR            NAME OF ENTRY POINT
     0517       3410                          STD       D.T0
     0520       3074                          LDD       D.PPIR
     0521       1601                          ADN       W.PPOR-W.PPIR
     0522       6210                          CWD       D.T0             WRITE OUTPUT REGISTER
     0523       0200 0527                     RJM     R.WAIT             WAIT FOR FUNCTION ACCEPTED
     0525       0367                          UJN       R.MTRX           EXIT
```

```
                                        *    R.WAIT     WAIT FOR OUTPUT REGISTER TO CLEAR                                STL      448
                                        *                                                                                STL      449
                                        *    CALLING SEQUENCE                                                            STL      450
                                        *                                                                                STL      451
                                        *    RJM        R.WAIT                                                           STL      452
                                        *                                                                                STL      453
                                        *    ACTION                                                                      STL      454
                                        *                                                                                STL      455
                                        *    WAIT FOR OUTPUT REGISTER TO CLEAR AND PAUSE FOR CM                          STL      456
                                        *    RELOCATION IF THE STORAGE MOVE FLAG IS SET                                  STL      457
                                        *                                                                                STL      458


      0525                              R.WAIT   ENM      X                                                              STL      460
      0530       3074                   WAITOR   LDD      D.PPIR                                                         STL      461
      0531       1601                            ADN      W.PPOR-W.PPIR                                                  STL      462
      0532       6010                            CPD      D.TO                                                           STL      463
      0533       3010                            LDD      D.TO                                                           STL      464
      0534       0471                            ZJN      R.WAITX                                                        STL      465
                                                                                                                         STL      466
      0535       5000 0100              WAIT1    LDM      R.FAF                                                          STL      467
      0537       0403                            ZJN      WAIT2              JUMP IF FIELD ACCESS FLAG NOT SET           STL      468
                                                                                                                         STL      469
      0540       0200 0446                       RJM    R.PAUSE              PAUSE FOR CM RELOCATION IF NECESSARY        STL      470
                                                                                                                         STL      471
      0542       1462                   WAIT2    LDN      RWDELAY/2                                                      STL      472
      0543       1701                            SBN      1                                                              STL      473
      0544       0576                            NJN      *-1                DELAY RWDELAY MICROSECONDS                  STL      474
      0545       0362                            UJN      WAITOR             LOOP                                        STL      475
      0546                                       BSS      5                  (FOR MXN)                                   STL      476
      0553                                       BSS      5                                                              STL      477
```

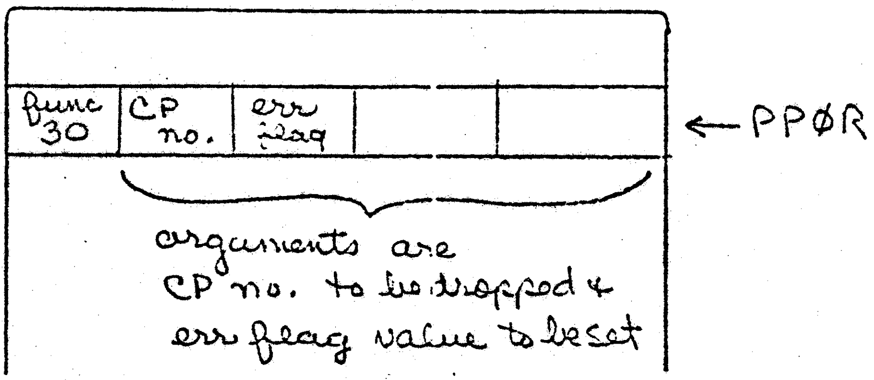
9-4

Example Formats the PP Comm Area May Have:

SCOPE 3.3

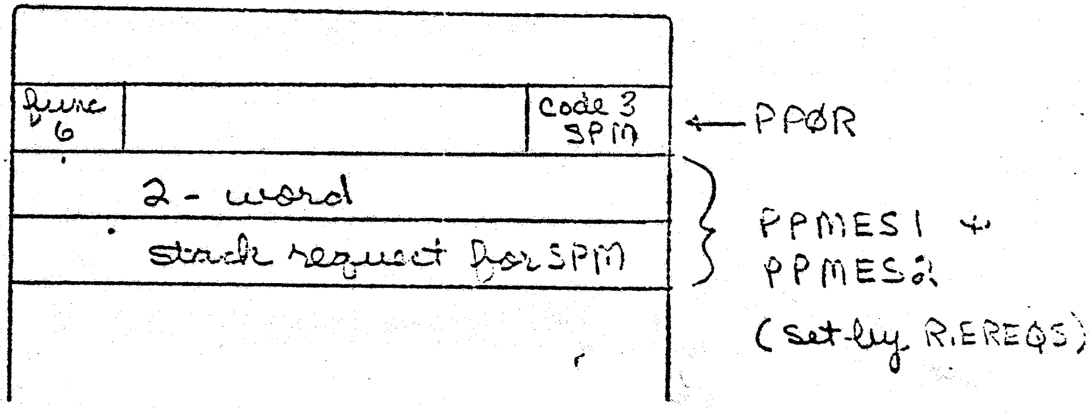
Normal MTR call (all params are in PPØR)

ie M.SEF

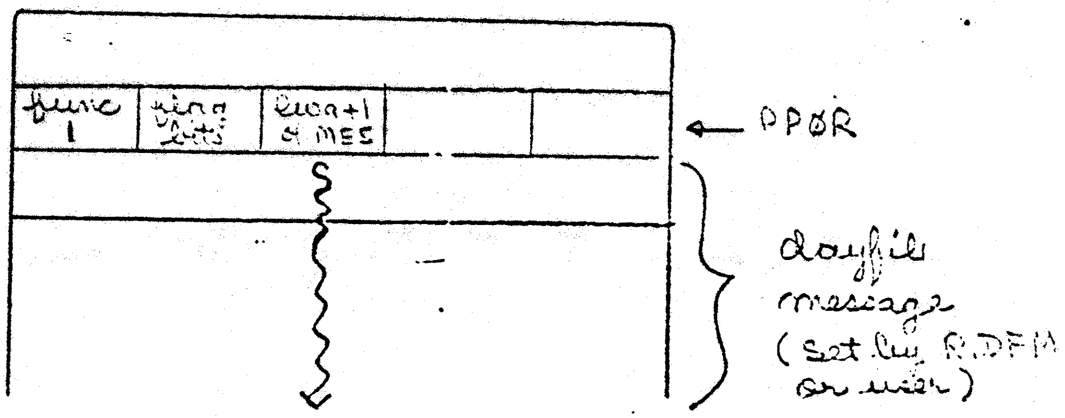


Some params are in PPMES area

ie M.ICE



ie M.DFM



any information required in the PPMES area must be set there by the PP program before calling R.MTR.

CC 8/71

PPM - [illegible] PP Monitor Request

All PPU requests are made by placing the function code of that request in byte zero of the PPU output register. Parameters for the various requests are supplied to PPM via bytes 1 - 4 of the output register and/or the PPU message buffer. Upon completion of the request, MTR replies by setting byte 0 of the output register to zero and gives the response back to the requesting PP in the remaining bytes of the output register or the message buffer.

If MTR detects that the format of the request is bad, it sets the high order bit of byte 0 of the appropriate PPU output register. This hangs the PPU, since PPM will ignore any request with this bit set. An appropriate message: 'PPn NAMcp BAD MTR REQUEST' is inserted in MTR's message buffer, and will be displayed at the bottom of the B display in flashing characters.

A complete description of the contents of the output register and function parameters for each of the above requests follows. Thosebits or bytes irrelevant to the function are denoted by *'s.

Following will be a description of each MTR function (from the PSI handbook) along with an example of how to use it.

Each description is formatted as follows:

ie M.ABØRT - abort Control Point

(0013, ****, ****, ****, ****) ← D.T0-D.T4

func code put in D.T0 by R.MTR

D.T1 - D.T4 params may be required for PP pgm. to set before calling R.MTR.

MONITOR FUNCTIONS

SCOPE 3.3

The table below lists all current MTR function codes and corresponding SCPTEXT mnemonics.

MONITOR FUNCTIONS

01	M.DFM	Process Dayfile Message
02	M.RCH	Request Channel
03	M.OVLERR	Issue OVL Error Message
04	M.PPTIME	Assign PPU Time
05	M.STEP	Monitor Step Control
06	M.ICE	Initiate Central Execution
07	M.RBTSTO	Request RBT Storage
10	M.RSTOR	Request Storage
11	M.AVTAPE	Update Available Tapes Count
12	M.DPP	Drop PPU
13	M.ABORT	Abort Control Point
14	M.NTIME	Enter New Time Limit
15	M.RCP	Request Central Processor
16	M.DCP	Drop Central Program
17		Reserved for CDC
20	M.RPP	Request PPU
21	M.RCLCP	Recall Central Program
22	M.REQP	Request Equipment
23	M.DEQP	Drop Equipment
24	M.RPRI	Request Priority Change
25	M.REM	Assign Error Exit Mode
26	M.SEQ	Assign Job Sequence Number
27	M.RACT	Request Control Point Activity
30	M.SEF	Set Error Flag
31	M.ISP	Initiate Stack Processor
32		Reserved for CDC
33		Reserved for CDC
34	M.SPRCL	Stack Processor Recall
35	M.CCPA	Change Control Point Assignment
36	M.CPUST	Change CPU Status
37	M.RPJ	Request Peripheral Job
40-47		Reserved for Installations
50-77		Reserved for CDC

pages 9-10 and 9-11

M.CCPA - Change Control Point Assignment

{0035,****,****,****,***N}

The requesting PPU is released from its current control point assignment in the same manner as if it had issued an M.DPP function, but its input register is not cleared. The PPU is then assigned to control point N with the new control point number inserted in its input register.

Example: IRA assigns himself to a "NEXT" control point. (He also uses M.CCPA to assign himself back to CP 0)

```
                   *                                                           1RA      548
                   *    ASSIGN THIS PP TO THE *NEXT* CONTROL POINT             1RA      549
                   *                                                           1RA      550
1400               SUJ      LDN    0                                           1RA      551
3413                        STD    D.T3                                        1RA      552
3055                        LDD    CPAD    }                                   1RA      553
1070                        SHN    -7      } put desired CP in D.T4            1RA      554
3414                        STD    D.T4    }                                   1RA      555
1435                        LDN    M.CCPA  } switch to that CP                 1RA      556
0200 0516                   RJM  R.MTR     }                                   1RA      557
3055                        LDD    CPAD    } Sets D.CPAD to CP he's going to work on  1RA  558
3475                        STD    D.CPAD  }                                   1RA      559
                            {
                            {

                   *                                                           1RA      647
1400       clears a PP    { LDN    P.ZERO        ASSIGN THIS PP BACK TO CP 0   1RA      648
6010       Clears 5 bytes { CRD    D.T0                                        1RA      649
1435                        LDN    M.CCPA  } switch back to CP0                1RA      650
0200 0516                   RJM  R.MTR     }                                   1RA      651
1400                        LDN    0       } now remembers CP 0 in D.CPAD      1RA      652
3476                        STD    D.CPAD  }                                   1RA      653
                            {
```

Any PP which services many control points (ie 1SP) will use M.CCPA

M.RSTOR - Request Storage

{0010,CCCC,XXXX,00TT,****}

TT = 00 CM request only
01 ECS request only
02 CM and ECS request

Assign CCCC hundred octal words of central memory and/or XXXX thousand octal words of extended core storage to the control point of the requesting PPU. Monitor replies to this request by setting CCCC and/or XXXX to the values actually assigned to the control point and by setting byte 0 to zero. These values should be compared with the original values requested to determine whether these requests have been honored or not. A request for more storage is rejected if not enough storage is available or if a storage move is already in progress. A request for less storage is always honored immediately. If TT = 02, MTR can honor a part of the request, without honoring the other.

9-10

Example: 1RA uses M.RSTOR to request storage for a new job.

```
         *
         *     REQUEST STORAGE  (INSTALLATION WITHOUT ECS)
         *
                         IFEQ   IP.MECS,0           ASSEMBLE IF NO ECS
         SUJ4            LDD    JOBFL
                         STD    D.T1 <-- amt core wanted
                         LDN    0
                         STD    D.T3 <-- cm only
                         LDN    M.RSTOR
                         RJM   R.MTR                REQUEST STORAGE
            reply  -->   LDD    D.T1                CHECK IF ASSIGNED
         from MTR        SBD    JOBFL
                         MJN   SUJ5                 JUMP IF STORAGE NOT ASSIGNED
                         LJM   SUJ7                 JUMP IF STORAGE ASSIGNED
                         ENDIF
```

M.AVTAPE - Update Available Tapes Count

{0011,000X,00MM,00NN,****}

Option 1: X is equal to 1
Then MM is the number of 7-track tapes and NN is the number of 9-track tapes requested by the PP program. If the requested number of both types of tapes cannot be satisfied, then the whole request is rejected and byte 1 of the PP output register is set non-zero. Byte 1 is set to 0 if the request is accepted.

Option 2: X is equal to 2
Then MM is the number of 7-track tapes and NN is the number of 9-track tapes to be returned to the system. This type of request will never be rejected; byte 1 will be set to 0.

MTR uses MM and NN to update bytes C.AVMT and C.AVNT in word P.AVTAPE in CMR.

Example: IRA uses M.AVTAPE to request the tapes from the system that the job needs.

```
                  *                                                              1RA
                  *        SETUP MTR REQUEST FOR NO OF 7 AND 9 TRACK REQUIRED    1RA
                  *                                                              1RA
1401                       LDN     1          requesting tapes                   1RA
3411                       STD     D.T1                  REQUEST OF TAPES        1RA
3035                       LDD     FST2+C.MT             NO OF SEVEN-TRACK REQUIRED  1RA
1071                       SHN     -6                                            1RA
3412                       STD     D.T2                                          1RA
3032                       LDD     FST2+C.NT             NO OF NINE-TRACK REQUIRED   1RA
1071                       SHN     -6                                            1RA
3413                       STD     D.T3                                          1RA
1411                       LDN     M.AVTAPE              ISSUE REQUEST TO DEDUCT TAPES  1RA
0200 0516                  RJM     R.MTR                 FROM SYSTEM             1RA
3011                       LDD     D.T1                                          1RA
0504                       NJN     SUJBA                 REQUEST REJECTED        1RA
                                       reply from MTR
```

M. REM - Assign Error Exit Mode

{0025, MMMM,****,****,****}

MTR assigns the value MMMM to the exit mode field in the control point exchange package area. {The control point cannot be in the waiting status.}

Examples:

① IRA uses M.REM to set the mode in the exchange package. note a constant "7" is used, so mode is not an IPARAM.

```
                          *                                                        1RA      832
                          *    SETUP THE CONTROL POINT AREA                        1RA      833
                          *                                                        1RA      834
1407                                LDN   7           EXIT MODE = 7B  (MTR REQUEST) 1RA     835
3411                                STD   D.T1                                     1RA      836
1425                                LDN   M.REM                                    1RA      837
0200 0516                           RJM   R.MTR                                    1RA      838
```

② IAJ uses M.REM to override the 7, for a MODE card.

```
                          *                                                        1AJ      912
                          *        MODE CARD PROCESSING                            1AJ      913
                          *                                                        1AJ      914
                                   {
                                   {
                                   {
                          *                                                        1AJ      928
                          *        REQUEST EXIT MODE USING MTR FUNCTION M.REM      1AJ      929
                          *                                                        1AJ      930
3411                               STD    D.T1        D.T1 HOLDS BINARY MODE       1AJ      931
1425                               LDN    M.REM       ISSUE MTR REQUEST.           1AJ      932
0200 0515                          RJM   R.MTR                                     1AJ      933
```

M.RPRI - REQUEST PRIORITY

{0024,PPPP,****,****,***N}

Assign priority PPPP to the control point of the requesting PPU if it is not control point zero. The parameter N gives the control point number to be considered if the requesting PPU is assigned to control point zero. In any other case, the value of N is irrelevant.

Example: IRA uses M.RPRI to set the priority in the CP area

```
                        {                                                    1RA     839
3024                    LDD   D.FNT+C.FPRI      PRIORITY (MTR REQUEST)       1RA     840
3411                    STD   D.T1                                           1RA     841
1424                    LDN   M.RPRI                                         1RA     842
0200 0516               RJM   R.MTR                                          1RA     843
                        {
```

priority from Job Card

M.ENPR from the console would also use M.RPRI

M.NTIME - Enter New Time Limit

{0014,TTTT,T***,****,***N}

A central processor job time limit of TTTTT seconds is entered at the control point. Any previous time limit is superseded. If the requesting PPU is assigned to control point zero, the parameter N will give the number of the control point to be considered; in any other case this parameter is irrelevant.

Example: M.NTIME is used to change the time limit of a job. For example, n.ENTL from the console will enter a new time limit. M.NTIME is not used by IRA when setting up the control point.

M.CPUST - Change CPU Status   (6500/6700 ONLY)
(0036,***X,****,****,***N)

Option 1: N is non-zero, X is non-zero.
The request is ignored if a CPU is off. If this is not the case, then the control point status is dedicated to the CPU X (X=1 or 2) and will only be able to use this CPU for the duration of the job.

Option 2: N is non-zero, X = 0.
Control point N is released from dedicated status.

Option 3: N = 0, X = 0.
If either CPU is off, it is returned to the on status. This does not affect a CPU that was locked off at deadstart load time.

Option 4: N = 0, X = 1 or 2.
CPU X is turned off. If any control point was dedicated to this CPU, the dedication is deactivated during period CPU is off; job returns to CPU when CPU is turned on again.

Example: IRA uses M.CPUST to switch cpu's

```
*                                                                 1RA        957
*          CDC6700 SUPPORT                                        1RA        958
*                                                                 1RA        959
CPP        IFNE   IP.MCPU,1                                       1RA        960
           LDN    P.ZERO                                          1RA        961
           CRD    D.T0                                            1RA        962
           LDD    FST1+C.FEQP                                     SCAK33X     43
           LPN    30B                  ISOLATE REQUESTED CPU      SCAK33X     44
           ZJN    CP6700                                          SCAK33X     45
           SHN    -3                                              SCAK33X     46
           STD    D.T1  —— X                                      1RA        967
           LDD    CPAD                                            SCAK33X     47
           SHN    -7                                              1RA        969
           STD    D.T4  —— N                                      1RA        970
           LDN    M.CPUST }                                       1RA        971
           RJM    R.MTR   }                                       1RA        972
CP6700     BSS    0                                               SCAK33X     48
CPP        ENDIF                                                  1RA        973
```

M.RPJ - Request Peripheral Job
(0037,DDDD,DDDD,****,****)

This function requests that another PPU program be initiated after a specified time delay. The first word of the requesting PPU message buffer contains the input register image of the new PPU program. The time delay is (DDDD DDDD)* 250_{10} microseconds. (This time is rounded upward to the next larger millisecond.) If the time delay is zero and no PPU is available, the request is entered in the PP job queue. If no space is available in the PP job queue buffer of MTR, the entire request remains pending until a queue entry becomes free.

M.RPJ is to make a pp request for the job queue or delay stack.

Example: IRA uses M.RPJ to recall himself back in every second. He then drops his pp. Note he must put the desired input register image (ie his own) in PPMES1 first.

```
                 *                                                              IRA       1060
                 *                                                              IRA       1061
                 *     RECALL - PUT IRA BACK INTO PP DEIAY STACK AND DROP PP    IRA       1062
                 *                                                              IRA       1063
3075             RECALL    LDD    D.PPMES1    WRITE INPUT REG TO MES BUF WORD 1 IRA       1064
6250                       CWD    D.PPIRB                                       IRA       1065
                 *                                                              IRA       1066
1400                       LDN    P.ZERO                                        IRA       1067
6010                       CRD    D.T0        time                              IRA       1068
                 *                                                              IRA       1069
2000 1750                  LDC    DELAY         NOTEO DELAY IS ASSUMED LESS THAN IRA      1070
3412                       STD    D.T2          77773 WHICH IS ABOUT 1 SECOND   IRA       1071
1437                       LDN    M.RPJ                                         IRA       1072
0200 C516                  RJM    R.MTR                                         IRA       1073
```

M.RPP - Request PPU

{0020,****,****,****,****}

This function requests immediate initiation of another PPU program. The first word of the requesting PPU message buffer contains the input register,image of the new PPU, including the control point number to which it should be assigned. The input register address of the assigned PPU is placed in the first byte of the requesting PPU message buffer {W.PPMES1, Byte 0}. A zero byte is returned and the request is rejected if no PPU is currently available.

M.RPP is for a program which wants an <u>immediate</u> pp, not simply an entry in the job queue or delay stack.

MTR responds with ~~D~~ byte 0 of PPMES1 = a PPIR if he was able to assign a PP

SCOPE 3.(1)

M.SEQ - Assign Job Sequence Number

{0026,****,****,****,****}

Monitor returns in byte 1 of the PPU output register a job sequence number {in display code}.

M.PPTIME - Assign PPU Time

{0004,****,****,****,****}

MTR adds the current time minus the PPU starting time to the accumulated PPU time in the control point area {word W.PPTIME}.

M.RBTSTO - Request RBT Storage

{0007,SSSS,****,****,****}

MTR sets SSSS*1008 as the new RBT starting address.

M.RACT - Request Control Point Activity

{0027,***N,IIII,****,****}

This equest allows a PPU to know the various activity counts of control point N at a given time {N cannot be zero}. If the parameter IIII is non-zero, the pseudo-activity count will be incremented or decremented by the constant IIII {after sign extension}. The reply of monitor is made via the PPU output register:

Byte	1	control point status {C.CPSTAT}
	2	Control Point Activity {General Activity} Count
	3	PP Delay Count
	4	Pseudo-activity Count

Example: 1AJ uses M.RACT to be sure he is the only pp assigned to a control point before he advances to the next control card. For example, DIS forces 1AJ to a control point, in which case 1AJ should drop out & let DIS continue.

```
3076           NEXT1    LDD    D.CPAD                                          1AJ
1070                    SHN    -7                                              1AJ
3411                    STD    D.T1                                            1AJ
1400                    LDN    0                                               1AJ
3412                    STD    D.T2                                            1AJ
1427                    LDN    M.RACT               REQUEST ACTIVITY COUNT     1AJ
0200 0516               RJM    R.MTR                                           1AJ
3712                    SOD    D.T2                                            1AJ
0603                    ZJN    NEXT2                JUMP IF PP                 1AJ
0100 4143               LJM    EXXIT                DROP PP IF MORE THAN 1     1AJ
```

CP no.

do not set pseudo count

see if 1 or 2 pp's are assigned

go drop out if more than 1

M.DCP - Drop Central Program
{0016,****,****,****,****}

Execution of the central processor job at the control point is stopped. The control point status is set to Z {zero status}; the secondary status is not altered.

The control point status bits prior to M.DCP are returned in byte 1 of the output register of the requesting PPU.

Example: The loader will M.DCP while he is loading in the user's field. (He might have been called via RA+1 without recall, & the user could do a memory macro & change his field!) He will reactivate the user's program using M.RCP.

M.RCP - Request Central Processor
{0015,****,****,****}

This request is ignored under the following conditions:

a. The requesting PPU is assigned to control point zero.
b. The error flag is set for the control point.
c. The job is already in the waiting status.

If none of the above conditions exist, MTR will set the job in

a. The automatic recall status {X+Y} if the auto-recall pointer still points to an incomplete status.
b. The waiting status {w} in any other case.

7/81

Example: 1AJ activates the loader (in CM) by using M.RCP after setting the P register in the ~~E~~ Exchange Package

set P to 55 in Exchange Package

```
1455                     LDN   55B           SET STARTUP ADDRESS =55B        1AJ
3401                     STD   D.T1                                          1AJ
3076                     LDD   D.CPAD                                        1AJ
6210                     CWD   D.T0                                          1AJ
```

Set-up RA+65, RA+66, etc

```
1415                     LDN   M.RCP         REQUEST CENTRAL PROCESSOR       1AJ
0200 0515                RJM   R.MTR                                         1AJ
```

M.DFM - Process Dayfile Message

{0001,FFFF,MMMM,****,****}

The dayfile flag bits FFFF determine {when set} the following message handling {bits 0 - 5 set by the calling PP, bits 6 - 8 by MTR}:

Bit 0	Do not sent to B display
1	Do not send to the control point dayfile
2	Do not send to system dayfile {No A display}
3	Flag as an accounting message
4	Send to hardware error file
5	Do not insert the job name in the system dayfile
6	FNT
7	Dayfile dump
8	Request to be handled by DSD

The parameter MMMM gives the LWA+1 of the message in the PP message buffer or gives a dump index when a dayfile dump is requested.

The possible value of the dayfile dump index is:

0	for a system dayfile dump
1 through N.CP	for a control point dayfile dump
N.CP+1	for a hardware error file dump

M.OVLERR - Issue OVL Error Message

(0003,****,****,****,NNNN)

NNNN = 0000 when the message "XXX NOT IN PP LIB" is to to be issued.

NNNN ≠ 0000 when there has been an ECS error in loading and the message "ERROR LOADING XXX FROM ECS LOC YYYYYY" is to be issued.

When this function is used, W.PPMES1 should contain the overlay name and W.PPMES1+2 should contain the ECS address in octal (if appropriate).

Example: M.ØVLERR is used by R.ØVL to issue error message before aborting the control point.

note

```
                                                                    STL      640
     PLAH      STD        D.T4          TELL MTR WHICH MESSAGE...   STL      641
               LDD        D.PPMES1      ZERO =(XXX NOT IN PP LIB)   STL      642
               CWD        D.T6          NON-ZERO = ERROR LOADING XXX FROM ECS STL 643
               LDN        M.OVLERR      PUT PROGRAM NAME IN MESSAGE BUFFER STL   644
               RJM      R.MTR           MTR ISSUES DAYFILE MESSAGE  STL      645
     GETOUT    LDN        M.ABORT                                   STL      646
               RJM      R.MTR           ABORT CONTROL POINT         STL      647
               LJM        R.IDLE        EXIT TO IDLE LOOP           STL      648
                                                                    STL      649
```

M.ICE - Initiate Central Execution

{0006,****,****,****,IIII}

The parameter IIII identifies a central memory resident program which will be started by MTR upon recognition of this request. These system programs run at control point N.CP+1 with a priority of 7777B, RA=0 and FL=377777B (ECS RA=0, ECS FL=10000000B). The M.ICE request is delayed if a system program is already active; MTR initiates only one system program at a time. Refer to page 7-8, paragraph 1, for more information on this.

IIII = 0 CM storage move
       1 ECS storage move
       2 ECS transfer (ICEBOX)
       3 Stack Processor Manager
       4 load PP program from ECS library

Example: 1AJ uses M.ICE to load from the ECS library

```
                 *                                                          1AJ
                 *        LOAD ABSOLUTE OVERLAY WHICH IS LOCATED IN ECS     1AJ
                 *                                                          1AJ
          2000   IP.ECLIB  EQU    IP.ECLIB                                  1AJ
                 ECS1      IFEQ   IP.ECLIB,0                                1AJ
                 ECS1      ELSE                                             1AJ
                                                                            1AJ
 2001            ECERRM    DIS    ,*PARITY ERROR LOADING FROM ECS  0000000* 1AJ
                                                                            1AJ
 3002            ECPROG    LDD    D.Z2             SETUP ICEBOX PARAMETERS... 1AJ
                           ~
                           ~
 1402                      LDN    X.ICE                                     1AJ
 3414                      STD    D.T4             INITIATE CENTRAL EXECUTIVE ICEBOX 1AJ
 1406                      LDN    M.ICE                                     1AJ
 0200 0516                 RJM    R.MTR            EXECUTE ICEBOX           1AJ
 3014                      LDD    D.T4             CHECK FOR PARITY ERROR   1AJ
 0603                      NJN    ECERR                                     1AJ
 0100 4077                 LJM    LOADED           NO ERROR, LOADING COMPLETED 1AJ
```

M.ISP - Initiate Stack Processor

{0031,***X,****,****,CCCC}

The CCCC parameter is the DST ordinal of the stack processor to be initiated. The request is delayed if the PP job queue is full. Otherwise, 1SP is either assigned to a PP or put at the top of the PP job queue if no PP is available.

The X parameter indicates whether MTR should activate a Stack Processor despite the fact that one with the same DST is active. This is used only for DST 1 and 3D0.

M.SPRCL - Stack Processor Recall

{0034,00AA,AAAA,000F,CCCC}

This function is called with F non-zero when a stack request has been completed to update the exit count of that control point {CPSR}, and to update the I/O channel time information {if IOTIME mods are assembled on} using AAAAAA which is the msec. count for PP time inserted by the stack processor. If F is zero, then only the I/O channel time information is updated. CCCC is the control point area address.

M.STEP - Monitor Step Control

{005,****,****,****,***N}

This control is initiated by a keyboard request. MTR sets an internal step control flag and at each subsequent request MTR pauses for console keyboard input. A space from the keyboard causes MTR to process the request. A period from the keyboard causes MTR to process the request and clear the step control flag to resume high speed operation. If N = 0, all PPU requests are stepped. If N is non-zero, control point N is the only one to be placed in step mode; only the requests issued by the PPU's assigned to control point N will be stepped.

M.RCLCP - Recall Central Program

{0021,****,****,****,****}

This request only has an effect if the central processor program associated with the requesting PPU is in the recall status and no error flag is set at the control point. In this case, the status of the control point is set to waiting {W}. In any other case, the status of the control point is not altered.

M.DPP - Drop PP
{0012,****,****,****,****}

MTR clears the PP control point assignment {the PP status word and the PP input register are cleared}.

Example: The normal exit for any pp is to drop himself and jump to R.IDLE.

```
                              }
1412              EXXIT      LDN   M.OPP          EXIT TO IDLE LOOP        1AJ
0200 0516                    RJM  R.MTR                                    1AJ
0100 0103                    LJM  R.IDLE                                   1AJ
                                                                           1AJ
                              }
```

M.ABORT - Abort Control Point

{0013,****,****,****,****,}

The job associated with the requesting PPU is terminated. The requesting processor is responsible for an explanatory message in the dayfile.

The operation of this function is identical with function M.DPP except that the error flag in the contol point area is set to F.ERPP{3} to note the abort function.

Example: M.ABORT is used by R.ØVL to abort a control point after issuing a dayfile message

" ______ NOT IN PP LIB" or

"ERROR LOADING ______ FROM ECS"

```
0214      3075          PPCALLER  LDD      D.PPMES1                                          STL      164
0215      6216                    CWD      D.T6                                              STL      165
0216      1403                    LDN      M.OVLERR     ISSUE PP CALL ERROR MESSAGE          STL      166
0217      0200 0516               RJM     R.MTR                                              STL      167
0221      1413          GETOUT    LDN      M.ABORT     ABORT CONTROL POINT                   STL      168
0222      0200 0516               RJM     R.MTR                                              STL      169
0224      0100 0103               LJM      R.IDLE       EXIT TO IDLE LOOP                    STL      170
```

M.SEF - Set Error Flag

{0030,***N,EEEE,****,****}

Monitor will drop the central program at control point N by putting the program in the zero status, and set the error flag to the value EEEE.

Example: 1AJ uses M.SEF to drop the central program which had a control card error. It first issues a dayfile message. Anny pp program can set the error flag (in word 20 CP area) to one of the desired values and then drop the PP.

```
                    *                                                                     1AJ        635
                    *         ERROR EXIT                                                  1AJ        636
                    *                                                                     1AJ        637
             entrance for c.card error
2000 1553    ->CCERR     LDC    MSG1   } ------- ISSUE DAYFILF ERROR MESSAGE              1AJ        639
0200 0665    ERROR       RJM  R.DFM    }                                                  1AJ        640
1400                     LDN    0                SET FLAG NOT TO RELOAD 1AJ               1AJ        641
3420                     STD    WORD1                                                     1AJ        642
3076         ERR2        LDD    D.CPAD }         SET ERROR FLAG                           1AJ        643
1070                     SHN    -7     } set N   CP no. in D.T1                           1AJ        644
3411                     STD    D.T1   }                                                  1AJ        645
1411                     LDN    F.EREX } set error flag value to 11                       1AJ        646
3412                     STD    D.T2   }                                                  1AJ        647
1430                     LDN    M.SEF  } go set the error flag in CM CP area              1AJ        648
0200 0516                RJM  R.MTR    }                                                  1AJ        649
3020                     LDD    WORD1                                                     1AJ        650
0403                     ZJN  ERR3           go M.DPP JUMP IF RELOAD NOT NECESSARY        1AJ        651
0100 4143                LJM  EXXIT <-       GO DROP PP SO AS TO FORCE RELOAD             1AJ        652
                                                                                          1AJ        653
0100 1052    ERR3        LJM  NEXT           NO RELOAD - LOOK FOR EXIT CARD               1AJ        654
                                                                                          1AJ        655
0317         MSG1      . DIS    ,*CONTROL CARD ERROR*                                     1AJ        656
                                             ^
                                              \ dayfile message
```

NWL's 1EJ
CS 11/72

9-30

ERROR NUMBE JUMP TABLE

1EJ issues the error message for any error code in his ERRTABLE. (note local mods). Otherwise the PP has to issue the message (ie 1AJ issuing CONTROL CARD ERROR message, p 9-29)

```
                    *                                                                   1EJ      3597
                    *                                                                   1EJ      3598
                    *                                                                   1EJ      3599
                    *     ERROR - IF ERROR FLAG IS SET ISSUE APPROPRIATE DFILE MESSAGE  1EJ      3600
                    *                                                                   1EJ      3601
                    *                                                                   1EJ      3602
0100 0000           ERROR     ENM    X                                                  1EJ      3603
                    *                                                                   1EJ      3604
5052 6443                     LDM    ERRTABLE,ERRFLAG  GET ADDR OF PROPER ERROR  ROUTINE 1EJ     3605
3410                          STD    0.T0                                               1EJ      3606
                    *                                                                   1EJ      3607
0110 0000                     LJM  0,0.T0              JUMP TO PROPER ROUTINE           1EJ      3608
                                                                                        1EJ      3609
                    *                                                                   1EJ      3610
6434                ERRTABLE  VFD    12/ERRORX         0 NO ERROR                       1EJ      3611
6502                          VFD    12/ERRTML         1 TIME LIMIT                     1EJ      3612
6502                          VFD    12/ERRARI         2 ARITH ERROR                    1EJ      3613
6533                          VFD    12/ERPOP3         3 PP ABORT (M.ABORT)             1EJ      3614
6573                          VFD    12/ERROP3         4 CP ABORT (ABT IN RA+1)         1EJ      3615
6505                          VFD    12/ERRPPC         5 PP CALL ERROR                  1EJ      3616
6513                          VFD    12/ERROPD         6 OPERATOR DROP                  1EJ      3617
6466                          VFD    12/ERRKILL        7 KILL                           1EJ      3618
6477                          VFD    12/ERRRRN         10 RERUN                         1EJ      3619
6533                          VFD    12/ERPOR3         11 CONTROL CARD ERROR            1EJ      3620
6510                          VFD    12/ERRECS         12 ECS PARITY ERROR              1EJ      3621
6575                          VFD    12/ERRJC          13 JOB CARD ERROR                1EJ      3622
6527                          VFD    12/ERRPA          14 JOB PRE-ABORT                 1EJ      3623
6521                          VFD    12/ERPCL          15 AUTO-RECALL ERROR             1EJ      3624
6646                          VFD    12/ERRHANG        16 JOB HUNG IN AUTO RECALL       1EJ      3625
5474                          VFD    12/ERMSL          17 MASS STORAGE LIMIT            1EJ      3626
6516                          VFD    12/ERRDOPD        20 DDS OPERATER DROP             DS33BJ6    29
6524                          VFD    12/ERRRT                                           TC1EJU      1
                              IFEQ   IP.RTMTR,1,1                                       TC1EJ     132
6524                          VFD    12/ERRRT          20  ILLEGAL RTM REQUEST          TC1EJ     133
```

see p. 9-31

at locn.6505

```
                  *                                                                   1EJ     3649
                  *      PP CALL ERROR                                                1EJ     3650
                  *                                                                   1EJ     3651
2000 6752         ERRPPC     LDC    PPCM                                              1EJ     3652
0322                         UJN    ERROR1                                            1EJ     3653
```

at locn.6576

```
                  *                                                                   1EJ     3703
                  *     JOB CARD ERROR                                                1EJ     3704
                  *                                                                   1EJ     3705
2000 7005         ERRJC      LDC    JCM                                               1EJ     3706
0100 6531                    UJK    ERROR1                                            1EJ     3707
```

9-31

```
                  *                                                                   1EJ     3768
                  *          ERROR MESSAGES                                           1EJ     3769
                  *                                                                   1EJ     3770
0317              DROPMSG    DIS    ,*CONTROL POINT DROPPED*                          1EJ     3771
1217              KILLM      DIS    ,*JOB KILLED*                                     1EJ     3772
1217              RRNM       DIS    ,*JOB RERUN*                                      1EJ     3773
2411              TMLM       DIS    ,*TIME LIMIT*                                     1EJ     3774
2020              PPCM       DIS    ,*PP CALL ERROR*                                  1EJ     3775
2404              DOPDM      DATA   H*DOS *                                           1EJ     3776
1720              OPDM       DIS    ,*OPERATOR DROP*                                0553B06    [illegible]
0503              ECSM       DIS    ,*ECS PARITY ERROR*                               1EJ     3777
1217              JCM        DIS    ,*JOB CARD ERROR*                                 1EJ     3778
1217              PAM        DIS    ,*JOB PRE-ABORTED*                                1EJ     3779
1217              HANGM      DIS    ,*JOB HUNG IN AUTO-RECALL*                        1EJ     3780
0125              RCLM       DIS    ,*AUTO-RECALL ERROR*                              1EJ     3781
0522              ARTA       DATA   12RERROR MODE =                                   1EJ     3782
5533              ARIB       DATA   4R 0.                                             1EJ     3783
5501              ARIF       DATA   10R ADDRESS =                                     1EJ     3784
0000              ARIC       DATA   0                                                 1EJ     3785
0000              ARID       DATA   0                                                 1EJ     3786
0000              ARIE       DATA   0                                                 1EJ     3787
0000                         DATA   0                                                 1EJ     3788
1501              ERMSLM     DIS    ,*MASS STORAGE LIMIT*                             1EJ     3789
                             IFEQ   IP.RTMTR,1,1                                      1EJ     3791
1114              RTERR      DIS    ,*ILLEGAL RTM REQUEST*                            TCLEJ   [illegible]
                                                                                      TCLEJ   [illegible]
```

SCOPE
3.4

MONITOR FUNCTIONS

01	M.SETST			Set CPU status bits
02	M.CLRST			Clear CPU status bits
03	M.RCP			Request central processor
04	M.DCP			Drop central processor
05	M.RCLCP			Recall central processor
06	M.ICE			Initiate central executive
		00	EX.CMSM	CM storage move
		01	EX.ECSM	ECS storage move
		02	EX.ECOVL	ECS overlay load
		03	EX.SPM	Call stack processor manager
		05	EX.SCH	Call scheduler
		06	EX.SCH1	Call scheduler
		07	EX.REQEB	Request ECS buffer
		10	EX.RELEB	Release ECS buffer
		11	EX.REQSB	Request system buffer
		12	EX.RELSB	Release system buffer
		13	EX.MVIN	Move data to ECS from system buffer
		14	EX.MVOUT	Move data from ECS to system buffer
		15	EX.FLHB	Flush buffer
		16	EX.CSWAP	Clean ECS after ECS RPE in swap file
		17	EX.AUTEB	Terminate automatic allocation
		20	EX.ECD	Display ECS
		21	EX.ECR	Release display
		22	EX.ECW	Modify ECS
		23	EX.CEM	Clear CEM-working flag
		24	EX.DDPER	Process DDP overlay loading error
		25	EX.ECLDV	Make successive partial reads of ECS record
07	M.CPUST			Change CPU status {IP.MCPU ≠ 1}
10	M.SLICE			MTR interrupts CPMTR at end of time slice for job
12	M.RCH			Reserve channel
13	M.DFM			Dayfile message
15	M.STEP			Enter step mode
16	M.RBTSTO			Request RBT storage
17	M.RSTOR			Request storage
20	M.TSR			Terminate storage request {IP.RTMTR ≠ 0}
21	M.DPP			Drop PP
22	M.ABORT			Abort control point and drop PP
25	M.SEQ			Assign job sequence number
26	M.SEF			Set error flag
27	M.ISP			Initiate stack processor
30	M.SPRCL			Stack processor recall
31	M.CCPA			Change control point assignment
32	M.RPJ			Request peripheral job
33	M.EES			Enter event stack
34	M.CPJ			Capture peripheral job
35	M.SCH			Initiate integrated scheduler
36	M.PASS			MTR ignores it - to be cleared by another routine
37	M.RACT			Request control point activity
41	M.NTIME			Enter new time limit
42	M.NOTE			Null function - cleared immeciately by PPMTR
43	M.RPCH			Request channel surveillance
44	M.BUFPTR			Buffer pointer address
45	M.PATCH			Enter a patch into MTR
46	M.TRACE			Turn on MTR trace.
47	M.SLPER			XJ other CPU
77	M.KILL			Issues bad monitor request

SCOPE 3.4

MONITOR

INTRODUCTION - CPMTR [illegible]

Monitor for SCOPE 3.4 is divided into two parts: The Central Processor Monitor, CPMTR, and the Peripheral Processor Monitor, MTR. CPMTR was designed to manage those functions which are directly executed by the CPU; i.e., CPU scheduling and monitor mode function. MTR maintains control over the system in general, operating in PP0. The following is a discussion of each of these monitors.

CPMTR - CENTRAL PROCESSOR MONITOR

As the name implies, CPMTR executes not in a PP but in the CPU. It is responsible for three main functions:

1. Processes those PP output register monitor functions with a value less than or equal to M.MTRCPU.

* 01	M.SETST	Set CPU status bit.
* 02	M.CLRST	Clear CPU status bits.
**03	M.RCP	Request central processor.
**04	M.DCP	Drop central processor.
**05	M.RCLCP	Recall central processor.
06	M.ICE	Initiate central executive.
**07	M.CPUST	Change CPU status (IP.MCPU≠1).
* 10	M.SLICE	Interrupts user job at end of time slice.
* 11	-	
**12	M.RCH	Reserve channel.

2. Processes user program RA+1 requests and system program output registers.

3. Schedules CPU(s).

CPMTR is designed to make use of the CEJ/MEJ feature, if it is available. There is a full description of this feature later in this chapter. Briefly, there are two modes of CPU execution: Monitor mode and user mode. Each time the registers are exchanged, the mode changes. When running in user mode, an exchange may be initiated by an XJ instruction from the CPU or an MXN or MAN instruction from a PP. When running in monitor mode the exchange is only initiated by the XJ instruction, therefore, monitor mode execution is not interrupted until it is finished.

CPMTR, SPM, and CPCIO execute in monitor mode. Integrated Scheduler, Storage Move, control point jobs and IDLE all run in user mode.

---

* New function for SCOPE 3.4.
** New function code for SCOPE 3.4.

If the CEJ/MEJ feature is not used all exchanges are initiated by MTR using the EXN instruction. CPMTR executes in a simulated monitor mode which it terminates by jumping to a loop that is recognized by MTR from the P address. MTR then EXNs to terminate CPMTR.

RA+1 PROCESSING

TIM is completely executed in CPMTR and control returned to the same control point. Auto recall is ignored.

END causes the CPU to be dropped and the CPU selection routine is called.

ABT is processed like END except that is also causes an M.SEF function to be passed to MTR through T.MTRRS.

RLC causes the recall status bit to be set, call the CP selection routine. An auto-recall bit also causes auto-recall status to be set. The X or Y status bits are set as is appropriate.

CIO is inspected to see if it is a PP call or a call to the central processor CIO and is processed accordingly.

Other RA+1 requests are passed to MTR for processing. If the recall bit is not set, control is returned to the same control point. If the auto-recall bit is set, the RCL procedure is followed.

Detailed CPMTR Functions

M.CLRST - Clear Status

{0002,BBBB,xxxx,xxxx,00NN}

Where:

BBBB = pattern of bits to be cleared.

NN = control point number {only if MTR output register}

Called to clear CPU status bits in control point areas. Will cause linkage or delinkage from chain of control points actively waiting for CPU.

M.CPUST - Change CPU Status

{0007,xxxX,xxxx,xxxx,xxxx}

Option 1	X = 0. If either CPU is off, it is returned to the on status. This does not affect a CPU that was locked off at deadstart load time.
Option 2	X = 1 or 2. CPU X is turned off. If any control point was dedicated to this CPU, it will not execute during the period the CPU is off; job returns to CPU when CPU is turned on again.

M.DCP - Drop Central Processor

{0004,****,****,****,00NN}

NN = Control point number {MTR only}.

Execution of the central processor job at the control point is stopped. The control point status bit C, D, W, X, and Y are cleared. The control point is removed from the active control point ring.

The control point status bits prior to M.DCP are returned to byte 1 of the output register of the requesting PPU.

M.ICE - Initiate Central Execution

{0006,PPPP,PPPP,PPPP,IIII}

The parameter IIII identifies a central memory program which will be started by CPMTR upon recognition of this request. Some of these programs run in user mode and some in monitor mode. Only one user mode program may be initiated at any time. A user mode program though, may be interrupted by the execution of a monitor mode function.

All programs initiated by an M.ICE operate with RA=0 and a large enough field length to allow access to the entire central memory and ECS.

The parameters passed in the center three bytes are interpreted differently by each of the central executive programs.

*	0	CM storage move
*	1	ECS storage move
	2	Load ECS resident overlay
	3	Stack Processor manager
	4	Unused
*	5	Scheduler
*	6	Scheduler {Storage Request Entry}
	7	Request ECS buffer
	10	Release ECS buffer
	11	Request system buffer
	12	Release system buffer
	13	RMS-ECS move
	14	ECS-RMS move
	15	Flush ECS buffer
	16	Clean ECS after ECS RPE in swap file
	17	Terminate automatic allocation
	20	Display ECS
	21	Release display
	22	Modify ECS
	23	Clear CEM working flag
	24	ECS transfer through DDP failure; try RMS
	25	Enable successful partial reads of ECS records

---

* Executes in User Mode

[illegible]

SCOPE
3.4

M.RCH - Request Channel
{0012,BBAA,DDCC,xxxx,RRRR}

AA = first choice channel number
BB = second choice channel number
CC = third choice channel number
DD = fourth choice channel number

RRRR = 0000 Request immediate reply
RRRR ≠ 0000 No reply until a requested channel has been reserved.

When channel zero is requested, it must be field AA. When BB, CC, or DD is zero it is assumed that this is not a channel request and that there are no alternate choices beyond it.

If none of the requested channels are available and an immediate reply is requested, MTR will set bytes 0 and 4 of the PPU output register to zero.

When a channel is granted, the number of that channel will be reutrned in the PPU output register byte 1 {location of AA}. Byte 4 will be set to a non-zero value.

On exit, if a channel has been reserved, the output register will look like: 0000 XXXX xxxx xxxx YYYY

Where:

XXXX = channel number
YYYY = PP input register address

If the CEJ/MEJ feature is not in use, this function is performed by MTR.

M.RCLCP - Recall Central Processor
{0005,FFFF,xxxx,xxxx,00NN}

Where:

FFFF = Control point address of CPU if pre-emption is to take place, or address plus 1 to flag I/O in process.

NN = Control point number {MTR only}.

This request has two forms:

1. If bit zero of FFFF is not set this function is used to remove a control point from recall status. The X and Y status bits are cleared. If the resulting status permits, the control point is linked into the ring of jobs waiting for the CPU. If FFFF contains the control point address, the CPU is assigned to the job immediately.

CPMTR

SCOPE
3.4

2. If bit zero of FFFF is set, this is a function issued by MTR when the value of a buffer pointer has changed while the job is in the active CPU ring but is not currently running. The effect is to schedule the job immediately.

M.RCP - Request Central Processor

{0003,xxxx,xxxx,xxxx,00NN}

Where:

NN = control point number {MTR only}

This request is synonymous with an M.SETST to set the W status, except that the pre-emption flag is set.

M.SETST - Set Status

{0001,BBBB,xxxx,xxxx,00NN}

Where:

BBBB = pattern of bits to be set

NN = control point number {only if MTR output register}

Called to set CPU status in control point area. Will cause linking or delinking when appropriate.

M.SLICE - Terminate Slice Period

{0010,xxxx,xxxx,xxxx,xxxx}

Only MTR can issue this function. It is issued to interrupt an executing user mode program so that CPMTR can reschedule.

MTR

MTR - SCOPE SYSTEM MONITOR - Overview

MTR is in general control of SCOPE. It is loaded into PP0 at deadstart time and remains there for the duration of system execution.

The primary function of MTR is to control and coordinate all system activities in order to avoid conflicts between the pool peripheral processors. It allocates the central and peripheral processors, central memory and ECS to the various control points. Additionally, MTR maintains the system clock.

MTR requests may be made in several ways. A PP may issue a MTR request by writing the request into the PPs own output register and going through R.MTR which in turn calls R.WAIT. R.WAIT writes the input address of the PP into T.PPIP. This word {T.PPID for CPMTR} is used to hold the latest request for MTR and save MTR the time of searching all the PP output registers.

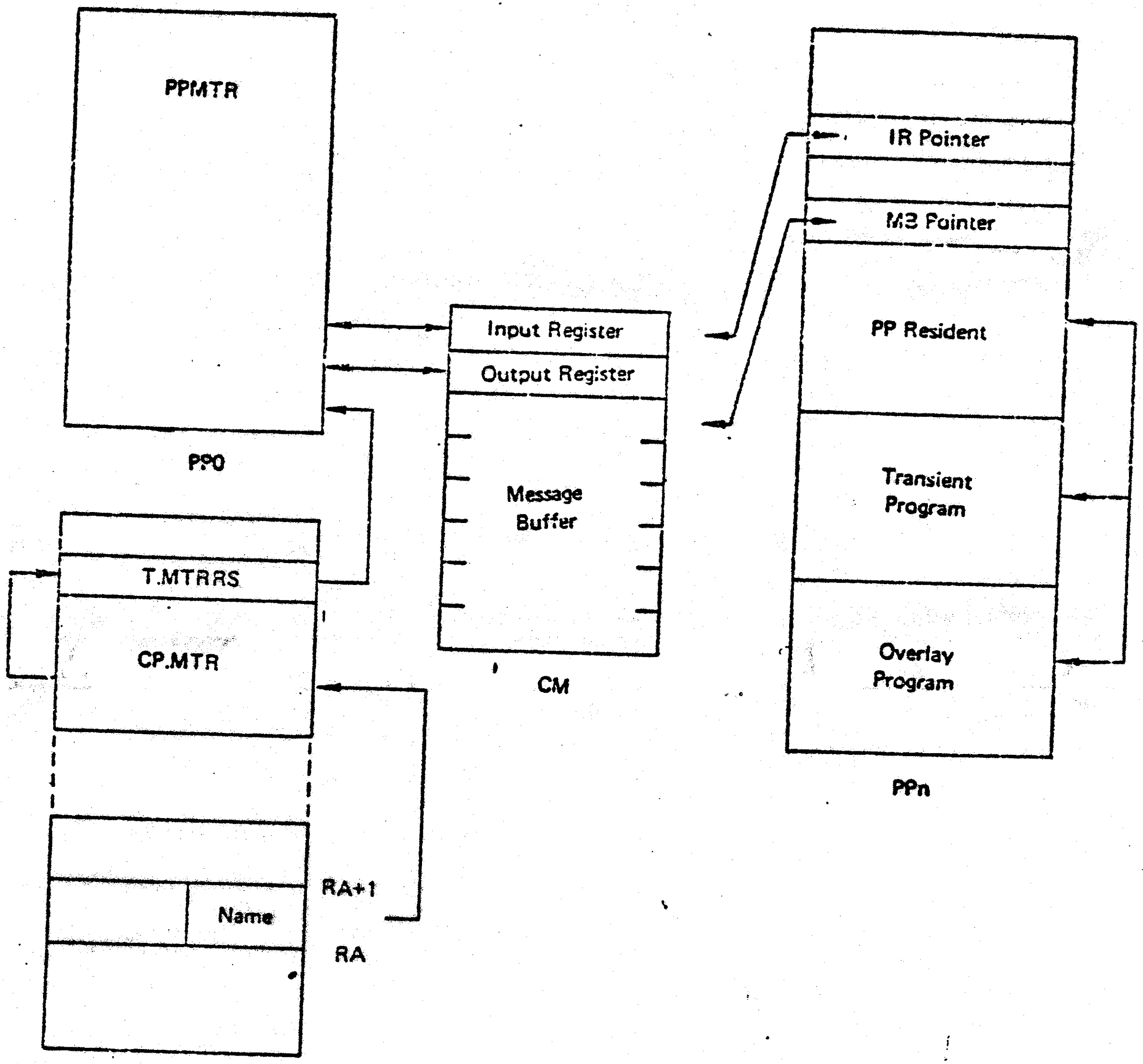
CPMTR itself can issue a MTR request by setting up the request and passing it through T.MTRRS, a four word circular buffer.

A user program may issue a request through RA+1 call. Both MTR and CPMTR scan for RA+1 requests. If there is an RA+1 request found by MTR, it will immediately initiate CPMTR. Here it is determined that the RA+1 call is to be executed by MTR and CPMTR will pass the request back to MTR through T.MTRRS.

✓

SCOPE
3.4

Monitor Request Processing



MTR

✓

Detailed MTR Functions

SCOPE 3.4

PP Monitor Functions

*	13	M.DFM	Issue dayfile message
*	15	M.STEP	Enter STEP mode
*	16	M.RBTSTO	Request RBT storage
*	17	M.RSTOR	Request storage
**	20	M.TSR	Terminate storage request
*	21	M.DPP	Drop PP
*	22	M.ABORT	Abort control point and drop PP
*	25	M.SEQ	Assign job sequence number
*	26	M.SEF	Set error flag
*	27	M.ISP	Initiate Stack Processor
*	30	M.SPRCL	Stack Processor Recall
*	31	M.CCPA	Change control point assignment
*	32	M.RPJ	Request peripheral job
**	33	M.EES	Enter event stack
**	34	M.CPJ	Capture peripheral job
**	35	M.SCH	Initiate Integrated Scheduler
**	36	M.PASS	To be cleared by another routine
*	37	M.RACT	Request control point activity
*	41	M.NTIME	Enter new time limit
**	42	M.NOTE	Null function, cleared immediately Used as break point.
**	43	M.PPCH	Request channel surveilance by PP MTR
**	44	M.BUFPTR	Buffer pointer address
**	45	M.PATCH	Enter a patch into MTR
**	46	M.TRACE	Turn on MTR TRACE
**	47	M.SLPER	XJ to other CPU
**	77	M.KILL	Bad MONITOR request made

---

* New function code for SCOPE 3.4

** New function for SCOPE 3.4

Function 23 {M.REQP} and 24 {M.DEQP} have been deleted because they were used so infrequently that their space {approximately 100 bytes in MTR} could not be justified. Any routines using these functions should be modified accordingly. This implies requesting CH.EST and searching the EST or the equipment needed. When the equipment is found, a check must be made for a control point number in the entry. If a control point number is present, the equipment is reserved; therefore drop the channel and try again later. If there is no control point number, the equipment is free and may be reserved by writing the requesting program's control point into the EST entry.

A complete description of the contents of the output register and function parameters for each of the above requests follows. Those bits or bytes irrelevant to the function are denoted by asterics {*}. Functions are in alphabetical order.

MTR

SCOPE 3.4

M.ABORT - Abort Control Point and Drop PP
{0022,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈⋈⋈}

The job associated with the requesting PP is terminated. The requesting processor is responsible for an explanation message in the dayfile. The operation of this function is identical with function M.DPP except that the error flag in the control point area is set to F.ERPP {3} to note the abort function.

M.BUFPTR - Address of Buffer Pointer Word
{0044,⋈⋈⋈⋈,⋈⋈⋈⋈,00AA,AAAA}

AAAAAA = Buffer Pointer Address

The address is absolute; the I/O driver has set the field access flag. The function is not cleared by MTR but by the I/O driver itself. When the low order 12 bits of the buffer pointer changes, MTR restarts the associated control point.

M.CCPA - Change Control Point Assignment
{0031,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈NN}

The requesting PPU is released from its current control point assignment in the same manner as if it had issued an M.DPP function, but its input register is not cleared. The PPU is then assigned to control point NN with the new control point number inserted in its input register The calling program must change the control point address at D.CPAD and rewrite the PP status word.

M.CPJ - Capture Peripheral Jobs
{0034,00XX,XXXX,⋈⋈⋈⋈,⋈⋈⋈⋈}

XXXXXX = Address relative to RA of the buffer where captured job data is to be placed.

Issued to find a job either in the event stack or in the PP delay stack for a control point. First the event stack is searched, then the delay stack. If a job is found, its data is written to a buffer specified in the call. When the end of the delay stack is reached, an exit is made.

M.DFM - Process Dayfile Message
{0013,FFFF,MMMM,⋈⋈⋈⋈,⋈⋈⋈⋈}

Where:

FFFF = Dayfile flag bits

MMMM = LWA+1 of message {MMMM PPOR}

= Dayfile dump index {MMMM PPOR}

The dayfile flag bits, when set, determine the following message handling {bits 0-5 set by the calling PP; bits 6-8 by MTR}:

MTR

SCOPE
3.4

Bit	0	Do not send to B display
	1	Do not send to the control point dayfile
	2	Do not send to system dayfile {no A display}
	3	Flag as an accounting message
	4	Send to hardware error file
	5	Do not insert the job name in system dayfile

The possible value of the dayfile dump index is:

0	For a system dayfile dump
1 thru N.CP	For a control point dayfile dump
N.CP+1	For a hardware error file dump

M.DDP - Drop PP

{0021,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈⋈⋈,⋈⋈⋈⋈}

MTR clears the PP control point assignment {the PP status word and PP input register are cleared.}

M.EES - Enter Event Stack

{0040,00AA,AAAA,⋈⋈⋈⋈,SYTT}

Where:

AAAAAA = Word address of Event Status

Y = Byte address in word

TT = Bit address in byte

S = F+B

F.ESOFF	0000	F=0	Assign when bit = 0
F.ESON	4000	F=4	Assign when bit = 1
F.ESABS	0000	B=0	AAAAAA is an absolute address
F.ESREL	1000	B=1	Relative to RA
F.ESCPA	2000	B=2	Control point address

The event stack is similar to the delay stack and is used by the Scheduler. This function writes to the peripheral job table, the PP input register and three parameter words. It then sets up the control point number and linkages in the event stack. The new entry is linked to the oldest prior entry. Now it clears the output register and exists. A separate queue is maintained for each control point.

MTR

SCOPE 3.4

M.ISP - Initiate Stack Processor

{0027,000X,xxxx,xxxx,CCCC}

Where:

X = 0 Initiate 1SS only if PP active flag = 0

≠ 0 Initiate 1SS whether PP active flag is set or not

CCCC = DST ordinal

A check is made to see if a PP is already assigned to this DST ordinal. If no, check for available PP and if one is available, proceed to either set the PP active flag or not and then store the DST ordinal and °1SS° in the input register. If a PP was assigned to the DST ordinal initially, check if 1SS should be initiated anyway. If no, clear the output register and exit. If yes, proceed as above if a PP is available or reserved. Exit if PP job queue is full. After the input register has been set up in the available PP the message buffers are cleared and the 1SP count updated. Check for a previously assigned 1SP. If no, get pointer to PP reserved when no 1SP is assigned. Push down the available PP chain. Whether another 1SP was assigned or not, if no PP is available, make an entry into the peripheral job table, identify it as a stack processor and push down the stack placing 1SP on top. Then, update the PPQ count and exit clearing the output register. If after 1SP assignment check there was a PP available assign 1SP to control point zero, set the 1SP flag for PP status and job queue and exit, clearing the output register.

M.NOTE - Null Function

{0042,xxxx,xxxx,xxxx,xxxx}

This function is for use in debugging PP programs. It may be used as a breakpoint. M.NOTE is cleared by MTR.

M.NTIME - Enter New Time Limit

{0041,TTTT,Txxx,xxxx,xxNN}

A central processor job time limit of TTTTT seconds is entered at the control point. Any previous time limit is superceded. If the requesting PPU is assigned to control point zero, the parameter NN will give the number of the control point to be considered; in any other case this parameter is irrelevant.

M.PASS - MTR PASS

{0036,xxxx,xxxx,xxxx,xxxx}

Indicates a no operation by MTR and it will be cleared by another routine.

MTR

SCOPE
3.4

M.PATCH - Enter Patch into MTR

{0045,AAAA,BBBB,CCCC,DDDD}

Where:

AAAA = address for BBBB

CCCC = address for DDDD

Simply inserts patch at address indicated. May be used by an operator during debugging of MTR.

M.PPCH - Request Channel Surveilance by PP MTR

{0043,BBAA,DDCC,****,RRRR}

AA = First choice channel number
BB = Second choice channel number
CC = Third choice channel number
DD = Fourth choice channel number
RRRR = 0 reply immediately ≠ 0 reply after reservation

This function is part of the RCH routine. M.RCH is a CP monitor function. If CP monitor requests a channel and the request is rejected, the M.RCH function is changed to M.PPCH, in order that MTR can keep a surveilance of the channels until the requested channel is free. MTR will update the channel reject history and when the requested channel is available, MTR will change the M.PPCH to M.RCH and initiate CP monitor.

M.RACT - Request Control Point Activity

{0037,**NN,IIII,****,****}

This request allows a PPU to know the various activity counts of control point NN at a given time {NN cannot be zero}. If the parameter IIII is non-zero, the pseudo-activity count will be incremented or decremented by the constant IIII {after sign extension}. The reply of monitor is made via the PPU output register:

Byte 1 control point status {C.CPSTAT}

2 control point activity {general activity} count

3 PP delay count

4 pseudo-activity count

M.RBTSTO - Request RBT Storage

{0016,SSSS,****,****,****}

MTR sets SSSS*100B as the new RBT starting address.

MTR

SCOPE
3.4

M.PPJ - Request Peripheral Job
{0032,DDDD,DDDD,HHHH,HHHH}

Where:

DDDDDDDD = time delay {unit = 0.25 msec.}

Requests initiation of a PP program. If the PP job queue is full, exit. If not, check if delay is requested. If no, process the PP job assignment. If delay is requested, enter request into the delay stack, write information out to the peripheral job table and rank it. Then update the control point activity count and the delayed PP count, clear the output register and exit.

M.RSTOR - Request Storage
{0017,CCCC,XXXX,00TT,HHHH}

Where:

CCCC = Requested CM/100B

XXXX = Requested ECS/1000B

TT = 00 CM request only

= 01 ECS request only

= 02 CM and ECS request

= 03 Request reserved CM

= 04 Request CM - will await request

= 06 Request CM+ECS will await request

= 07 IP.POSFL from swapper

= 20 Priority storage requested

MTR replies to request by setting CCCC and/or XXXX to values actually assigned to the control point and by setting byte 0 to zero. These values should be compared with the original values requested to determine whether these requests have been honored or not. A request for more storage is rejected if not enough storage is available or if a storage move is already in progress. A request for less storage is always honored as soon as possible. If TT = 02 or 06, MTR can honor a part of the request, without honoring the other.

M.SCH - Initiate Integrated Scheduler
{0035,000X,****,****,****}

Where:

X = 2 places {0000,0R1=0R4} in stack ELSE initiate Scheduler immediately.

If X≠2, call the Scheduler, clear the output register and exit. Otherwise, check if Request Stack is full and if yes, call the Scheduler to empty it before making stack entry. If stack is not full, make entry. In either case, advance the request stack pointer.

M.SEF - Set Error Flag
{0026,**NN,EEEE,****,****}

Monitor will drop the central program at control point NN and set the error flag to the value EEEE.

M.SEQ - Assign Job Sequence Number
{0025,****,****,****,****}

Monitor returns in byte 1 of the PPU output register a job sequence number {in display code}.

M.SLPER - Initiate CPMTR in Other CPU
{0047,****,****,****,****}

M.SLPER is issued to initiate CPMTR in the other CPU. CPMTR itself will check and issue the function if the other CPU should be executing.

M.SPRCL - Stack Processor Recall
{0030,00SS,AAAA,000F,CCCC}

This function is called with F non-zero when a stack request has been completed to update the exit count of that control point {CPSR}, and to update the I/O channel time information {if IOTIME mods are assembled on} using SS*AAAA which is the disk factor {SS=millisecond/PRV*4}*PRV count {AAAA}. If F is zero, then only the I/O channel time information is updated. CCCC is the control point area address.

M.STEP - Monitor Step Control
{0015,****,****,****,**NN}

This control is initiated by a keyboard request. MTR sets an internal step control flag and at each subsequent request MTR pauses for console keyboard input. A space from the keyboard causes MTR to process the request. A period from the keyboard causes MTR to process the request and clear the step control flag to resume high speed operation. If NN = 0, all PPU requests are stepped. If

NN is non-zero, control point NN is the only one to be placed in step mode; only the requests issued by the PPU's assigned to control point N will be stepped.

M.TRACE - Trace Output Registers
{0046,AAAA,FFFF,NNNN,****}

Where:

AAAA	=	Absolute address of buffer/100B {typically within job's field length}
FFFF	=	Field length of buffer/100B
NNNN	=	Number of next word pair in buffer.

This is a function reserved for CDC development. A buffer is defined for MTR to dump its trace of the monitor functions issued by other PPs. A trace record consists of a two word entry contain function and PP status information.

M.TSR - Terminate Storage Request
{0020,****,****,****,****}

DSD issues this function when the operator types DIRABT. This function clears the SMABT flag to zero.

This causes the rejection of an M.RSTOR function that is hung up because a control point will not allow itself to be moved.

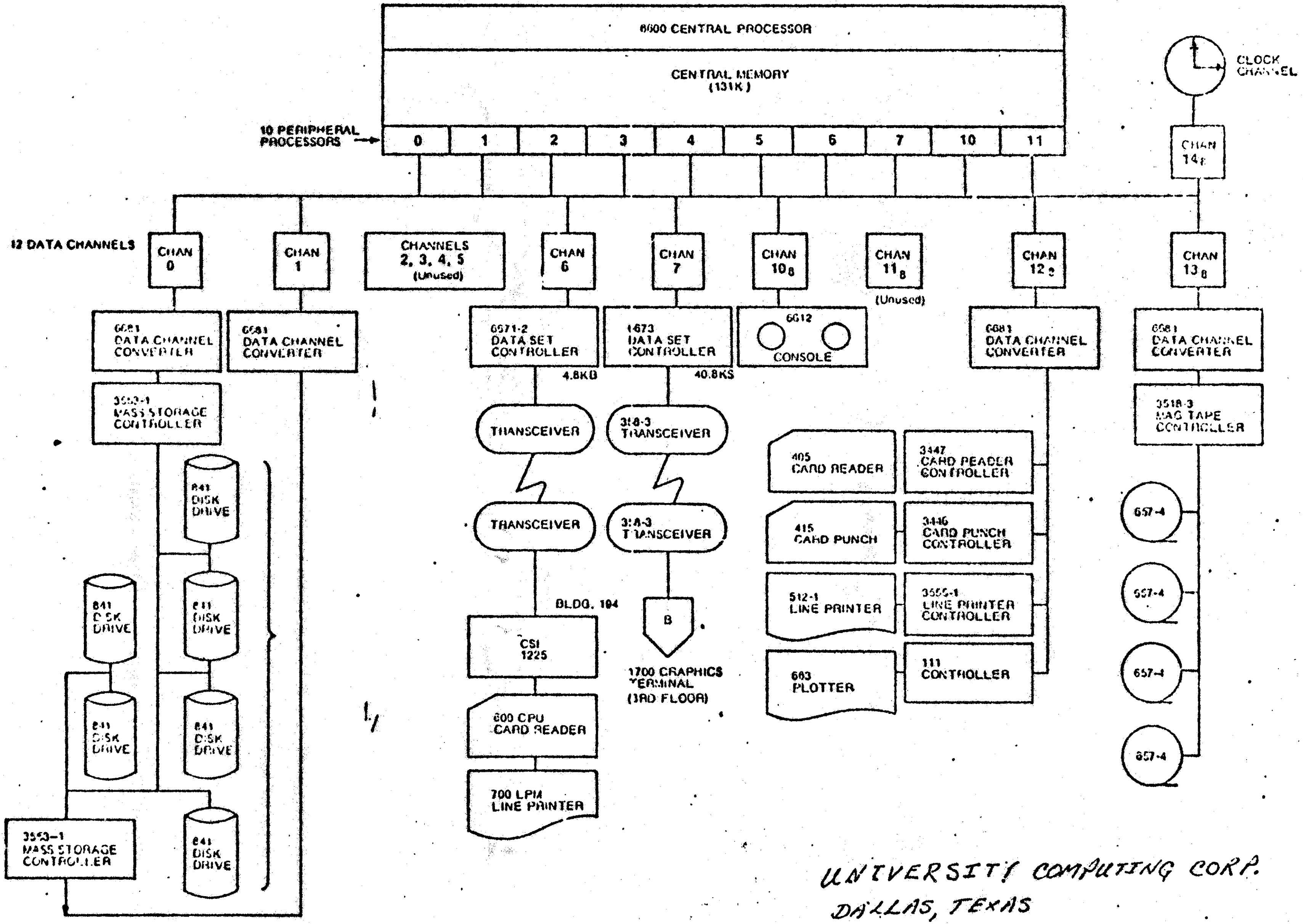
SECTION TEN

EXTERNAL INPUT / OUTPUT

SECTION TEN - EXTERNAL INPUT/OUTPUT

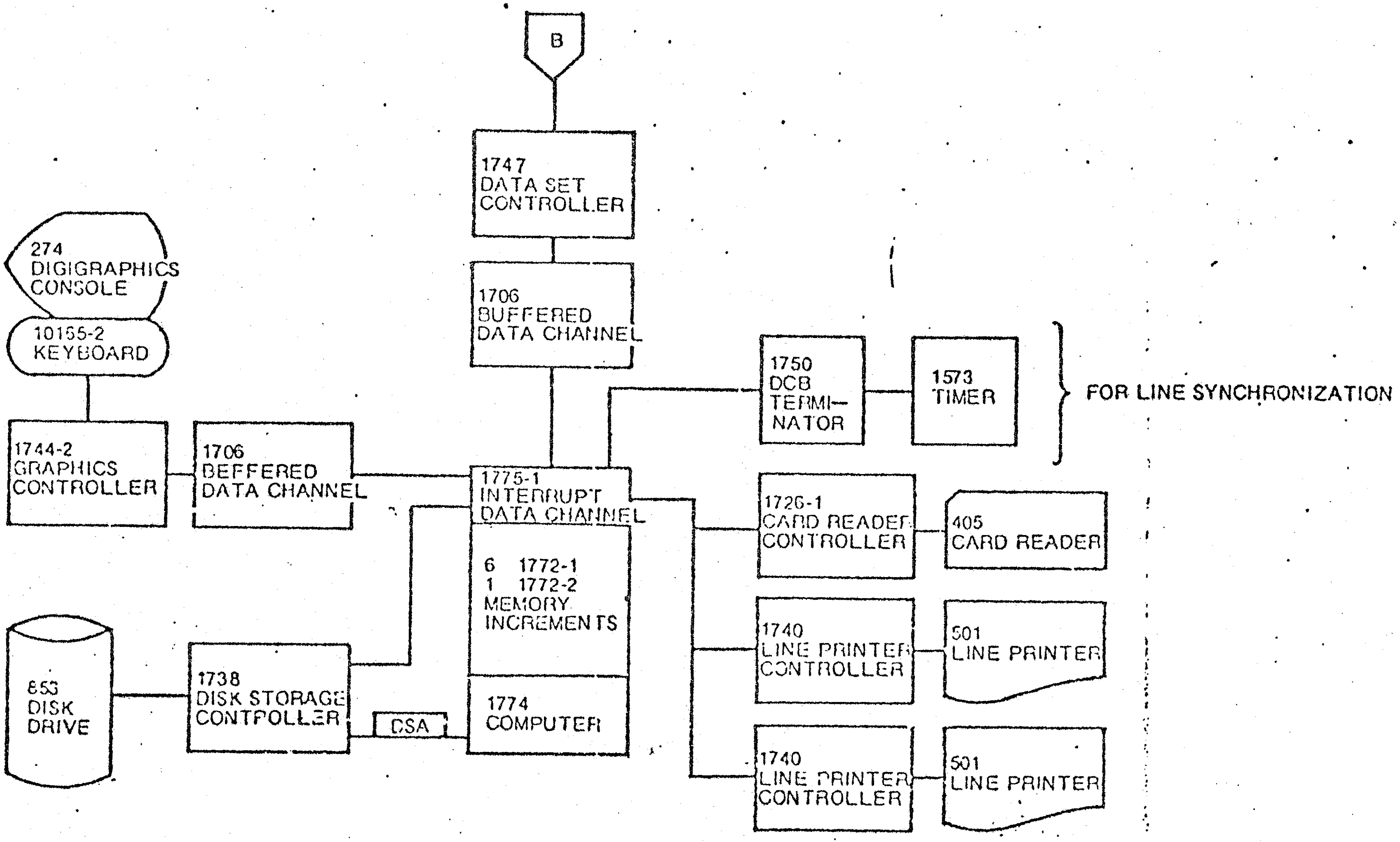
CONTENTS

6600 CONFIGURATION



UNIVERSITY COMPUTING CORP.
DALLAS, TEXAS
10/72

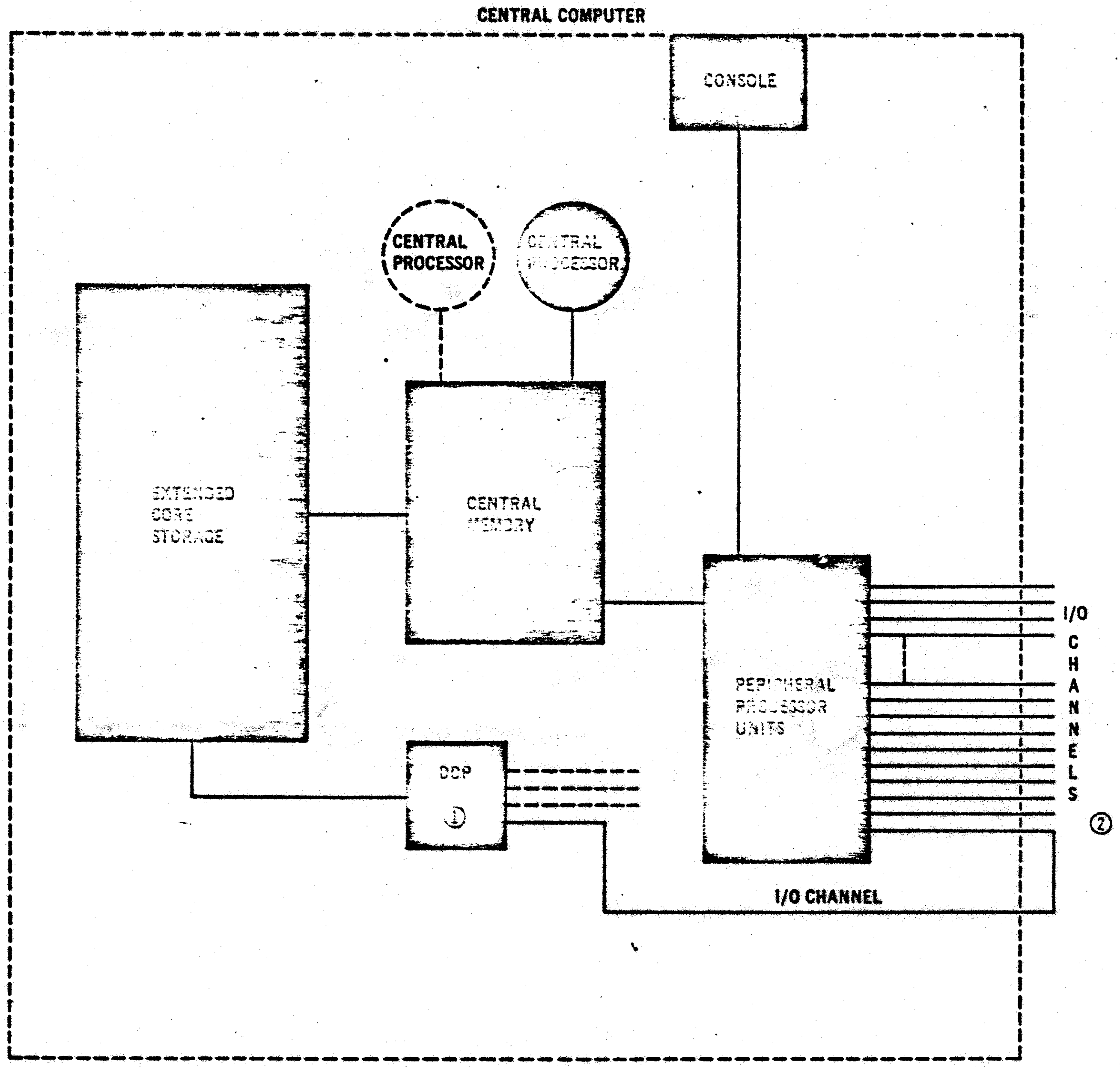
REMOTE GRAPHICS/ BATCH TERMINAL



C-11

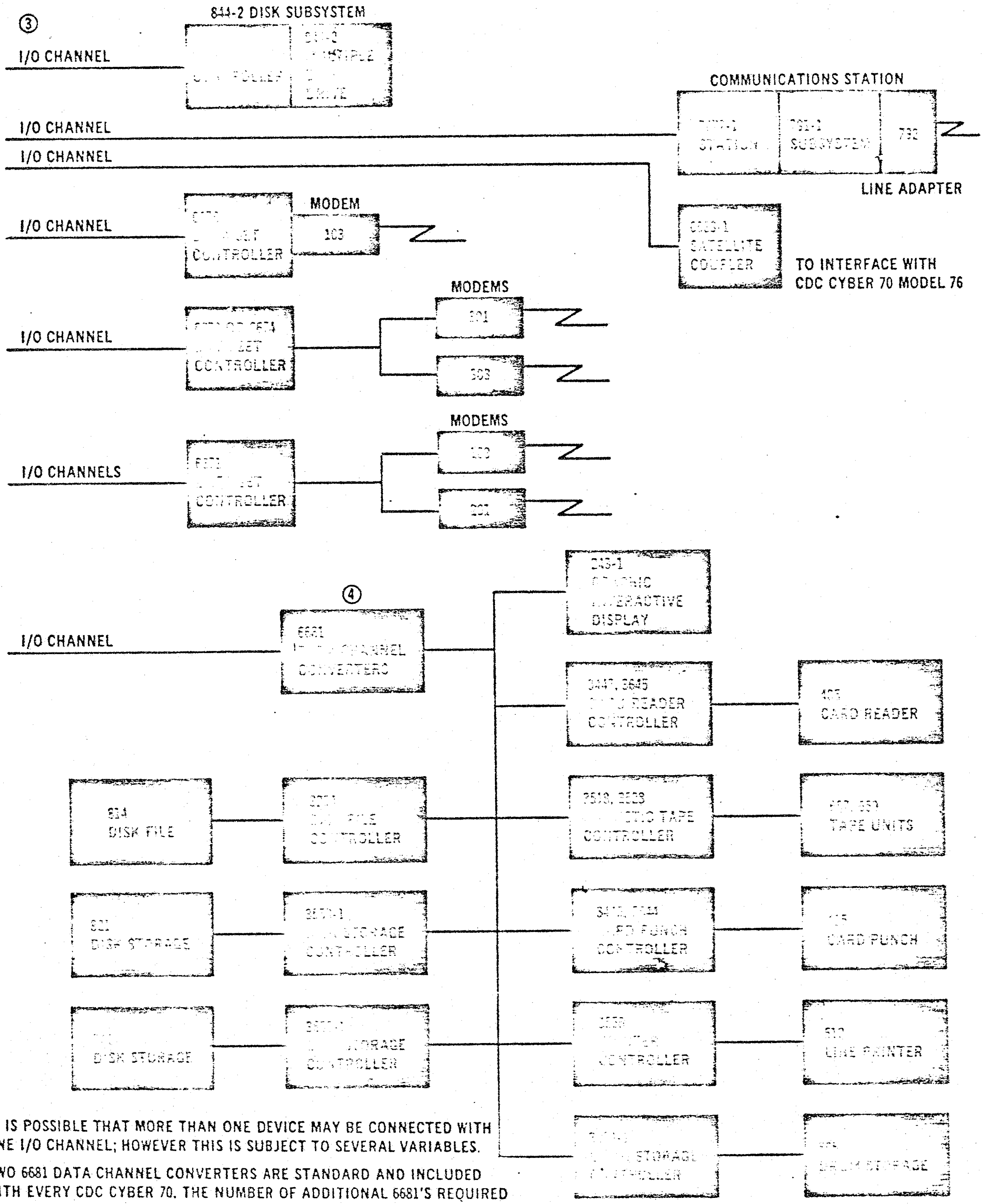
UCC, DA[illegible]AS, TEX.
10/7[illegible]

CDC CYBER 70 MODELS 72, 73, 74 GENERAL CONFIGURATION GUIDE



① DISTRIBUTIVE DATA PATH.

② TO PERIPHERALS AND CONTROLLERS.

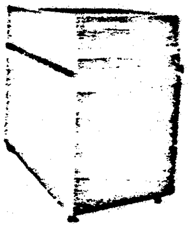


③ IT IS POSSIBLE THAT MORE THAN ONE DEVICE MAY BE CONNECTED WITH ONE I/O CHANNEL; HOWEVER THIS IS SUBJECT TO SEVERAL VARIABLES.

④ TWO 6681 DATA CHANNEL CONVERTERS ARE STANDARD AND INCLUDED WITH EVERY CDC CYBER 70. THE NUMBER OF ADDITIONAL 6681'S REQUIRED IS A FUNCTION OF SEVERAL VARIABLES, BUT NORMALLY VARIES BETWEEN 1 AND 3, WITH 2 BEING MOST COMMON.

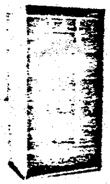
Peripheral Devices

DISK STORAGE DRIVE



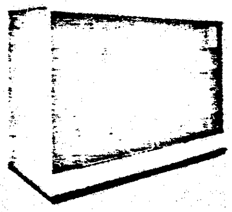
Summary Description	844-2
Controller:	7054-1
Capacity:	869 million bits
Positioning Time:	10 to 55 milliseconds
Average Access Time:	30 milliseconds
Transfer Rate:	6.8 million bps
Disk Pack:	872

MASS STORAGE CONTROLLER



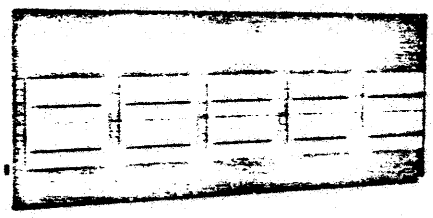
Summary Description	7054-1
Receives from:	Model 72, 73, or 74 Computer System
Sends to:	844-2 Disk Storage Drive
Controls:	Up to 8 Disk Storage Drives
Connects to:	One I/O channel

DISK FILE SYSTEM



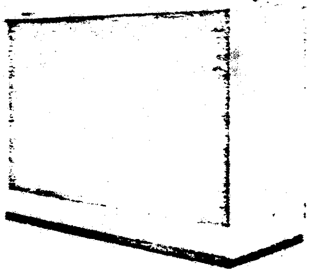
Summary Description	7638
Receives from:	Model 76 Computer System
Capacity:	800 million 6-bit characters
Positioning Time:	20-140 milliseconds
Transfer Rate:	6.67 million cps
Number of Disks:	72

MULTIPLE DISK DRIVE



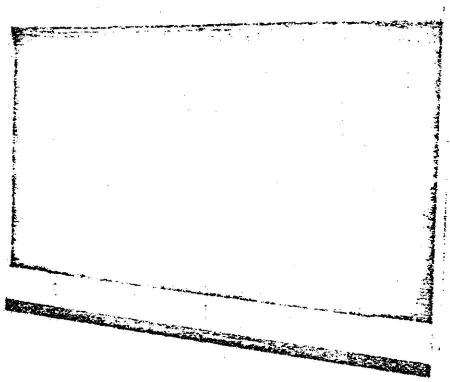
Summary Description	841
Controller:	3553-1 3553-2
Capacity:	107 to 286 million 6-bit characters
Positioning Time:	25 to 135 milliseconds
Average Access Time:	75 milliseconds
Transfer Rate:	420,000 cps
Disk Pack:	One 871

MASS STORAGE DISK



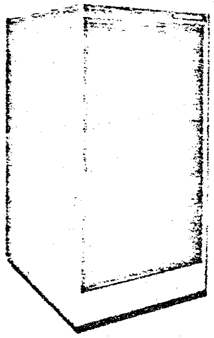
Summary Description	813	814
Controller:	3234	3234
No. of Disk Files per Controller:	8	8
Capacity (6-Bit Characters):	133,000,000	266,000,000
Positioning Time:	25-110 milliseconds	25-110 milliseconds
Average Latency:	25 milliseconds	25 milliseconds
Character Transfer Rate:	196,000 per second	196,000 per second

MASS STORAGE DISK



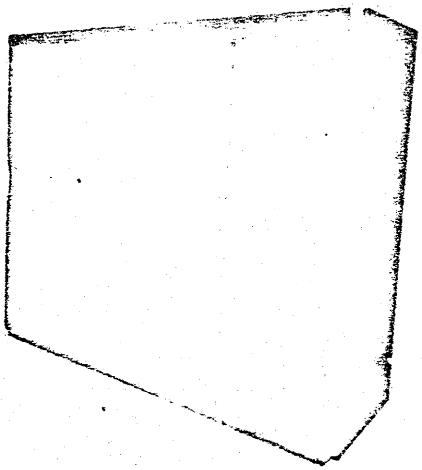
Summary Description	821-X
Controller:	3553
No. of File/Disks per Controller:	8
Capacity (Characters):	419-838,000
Positioning Time:	25-145 milliseconds
Average Latency:	19 milliseconds
Character Transfer Rate:	420,000 per second

MASS STORAGE DRUM



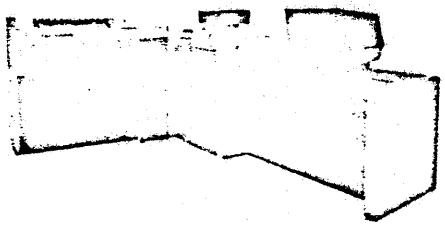
Summary Description	865
Controller:	3637
No. of Drums per Controller:	8
Capacity (Characters):	8,300,000
Average Access Time:	17 milliseconds
Storage Transfer Rate:	1,000,000 cps

EXTENDED CORE STORAGE



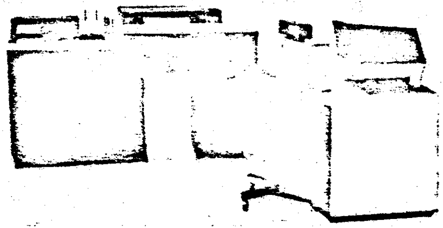
Summary Description	7030
Access Time:	3.2 microseconds per 488 bits
Maximum Storage:	20 million characters
Distributive Data Path:	480-bit buffer register (up to 4)
	7030-1
Core Storage:	1,259,520 characters
Transfer Rate:	25 million cps
	7030-2
Core Storage:	2,519,040 characters
Transfer Rate:	50 million cps
	7030-4
Core Storage:	5,038,080 characters
Transfer Rate:	100 million cps
	7030-8
Core Storage:	10,076,160 characters
Transfer Rate:	100 million cps
	7030-16
Core Storage:	20,152,320 characters
Transfer Rate:	100 million cps

LOW-SPEED BATCH TERMINAL



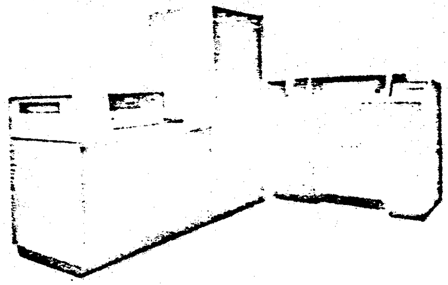
Summary Description	731
Receives from:	Communications Station/Subsystem
Memory:	8K; 8-bit bytes of 16-bit words
Card Reader:	300 cpm
Line Printer:	300 lpm
Options:	730-100 Memory Increment
	730-101 Display
	730-102 Eight-Channel Increment
	730-103 Cyclic Encoder
	730-104 Card Punch/Reader

MEDIUM-SPEED BATCH TERMINAL



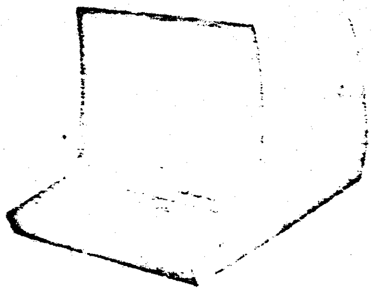
Summary Description	732
Receives from:	Communications Station/Subsystem
Memory:	8K; 8-bit bytes of 16-bit words
Card Reader:	500 cpm
Line Printer:	600 lpm
Options:	730-100 Memory Increment
	730-101 Display
	730-102 Eight-Channel Increment
	730-103 Cyclic Encoder
	730-104 Card Punch/Reader

HIGH-SPEED BATCH TERMINAL



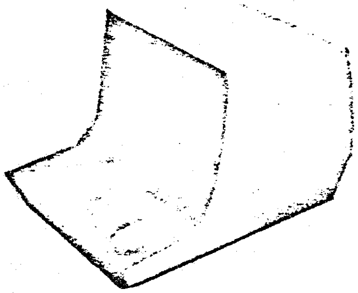
Summary Description	733
Receives from:	Communications Station/Subsystem
Memory (200 nanosecond):	8K; 8-bit bytes of 16-bit words
Card Reader:	1200 cpm
Line Printer:	1200 lpm (595-X print train)
Options:	733-101
	733-110
	733-120
	733-140 Memory Increment

REMOTE ALPHA/NUMERIC DISPLAY TERMINAL



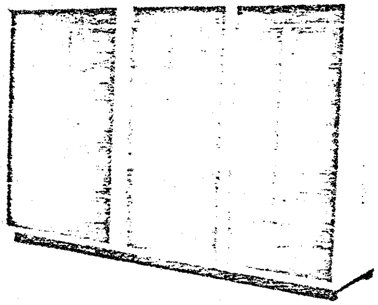
Summary Description	711
Controller:	Included in 711
Memory Capacity:	256 8-bit words
No. of Consoles per Controller:	1
CRT Entry/Display:	8 inches x 10 inches
Keyboard:	Electronic
No. of Characters per Display:	640

TELETYPEWRITER COMPATIBLE DISPLAY TERMINAL



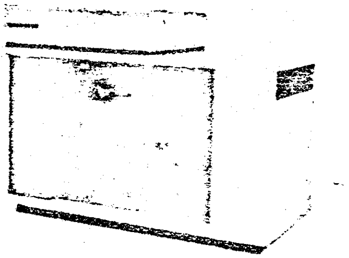
Summary Description	713
Controller:	Included in 713
CRT Entry/Display:	8 inches x 10 inches
No. of Consoles per Controller:	1
Memory Capacity:	256 8-bit words
Keyboard:	Electronic
No. of Characters per Display:	640 or 1280

COMMUNICATIONS STATION



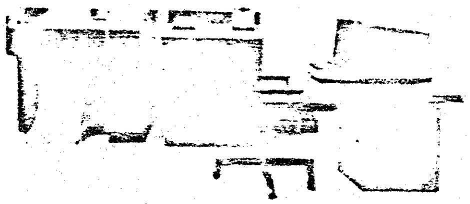
Summary Description	7077-1
Interfaces to:	CDC Cyber 70 Models 72, 73, 74 input/output channel
Controls:	Up to three 791-1 (144 channels) Communication Subsystems
Memory:	16,384 bytes
Cycle Time:	1.1 microsecond
Memory Expansion:	To 65,536 bytes CDC 10262 Memory Module
Buffer Storage:	8K words

COMMUNICATIONS SUBSYSTEM



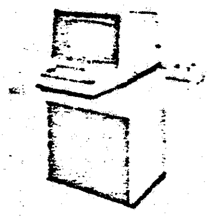
Summary Description	791-1
Interfaces to:	7077-1 Communications Station or 7611-10 Service Station
Communication Adapters:	48
Sends to:	(up to 16) 792 Communication Adapters
Core Memory:	4096 16-bit words
Cycle Time:	200 nanosecond
Line Speed:	75 to 50,000 bps
Expansion:	10274-1 Memory Module

REMOTE TERMINAL



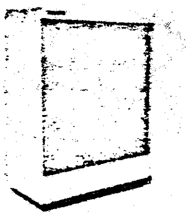
Summary Description	200 User Terminal
CRT Entry/Display:	6 inches x 8 inches
Card Reader:	333 cpm
Printer:	300 lpm
Batch Operations:	Yes
Interactive Operations:	Yes
No. of Printable Characters:	63

REMOTE ENTRY/DISPLAY TERMINAL



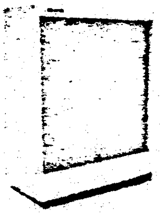
Summary Description	217-2
Controller:	Included in 217-2
No. of Devices per Controller:	1
Display Size:	6 inches x 8 inches
Format:	20 x 50
No. of Printable Characters:	63
Character Set:	BCD coded
Keyboard:	Included

DATA SET CONTROLLER



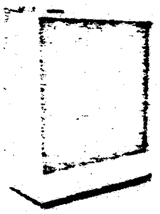
Summary Description	6671
Receives from:	Model 72, 73, or 74 Computer System
Controls:	Up to 16 103 (110 bps) or 201 (2000, 2400 bps) Data Sets

DATA SET CONTROLLER



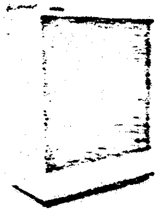
Summary Description	6673
Receives from:	(same as 6671)
Controls:	Up to 16 103 (110 bps) or 201 (2000, 2400, 4800 bps) Data Sets

DATA SET CONTROLLER



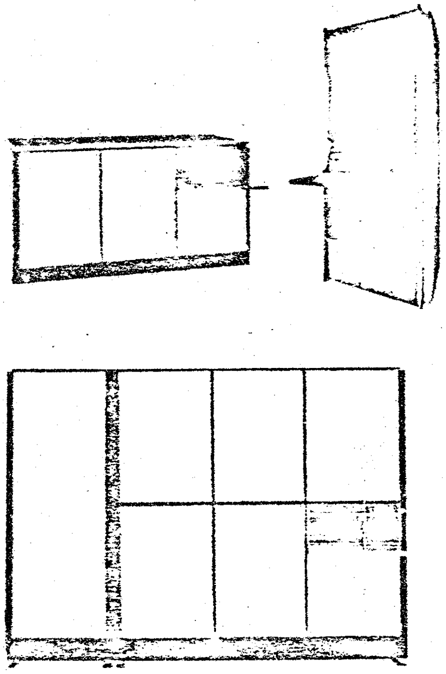
Summary Description	6674
Receives from:	(same as 6671)
Controls:	Up to 4 (40.8K bps) 301B Data Sets

DATA SET CONTROLLER



Summary Description	6676
Receives from:	(same as 6671)
Controls:	Up to 64 (110 bps) 103 Data Sets
Compatible with:	Model 33 or 35 TTY

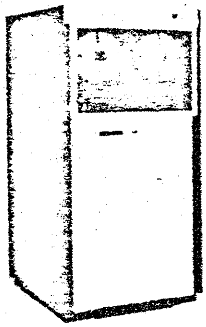
CENTRAL PROCESSOR



Summary Description	1704	1774	1714
Storage Capacity:	4,096 to 32,768 words	4,096 to 32,768 words	24,576 to 65,526 words
Cycle Time:	1.1 microseconds	1.5 microseconds	1.1 microseconds
Word Length:	18 bits*	18 bits*	18 bits*
Average Instruction Execution Time:	3.8 microseconds	10.0 microseconds	3.8 microseconds
Addressability:	word, relative	character, word, relative	word, relative
Memory Protect:	Yes	Yes	Yes
Number of Interrupts:	16	16	16
Number of Channels:	1 Buffered 1 Unbuffered	1 Buffered 1 Unbuffered	1 Buffered 1 Unbuffered
Number of Peripherals per Channel:	8	8	8
Memory Protect:	Yes	Yes	Yes
Memory Parity Check:	Yes	Yes	Yes
Number of Registers:	9	9	9

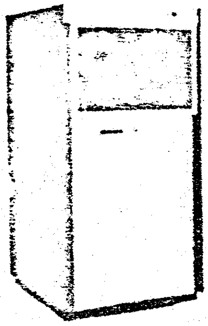
*Includes one parity and one memory protect bit.

TAPE TRANSPORTS



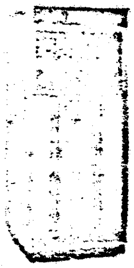
Summary Description	657-1	657-2	657-3	657-4
Tape Speed, IPS:	37.5	75	112.5	150
Densities:	200, 556, 800 bpi	200, 556, 800 bpi	200, 556, 800 bpi	200, 556, 800 bpi
Transfer Rate, KC:	7.5, 20.8, 30	15, 41.7, 60	22.5, 62.5, 90	30, 83.3, 120
Rev. Read:	Yes	Yes	Yes	Yes
No. of Tracks:	7	7	7	7
Controller:	3518-1 3518-2 3518-3 3528-1 3528-2 3528-3			

TAPE TRANSPORTS



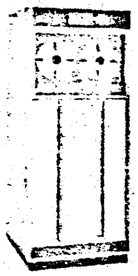
Summary Description	659-1	659-2	659-3	659-4
Tape Speed, IPS:	37.5	75	112.5	150
Densities:	800, 1600 bpi	800, 1600 bpi	800, 1600 bpi	800, 1600 bpi
Transfer Rate, KC:	30, 60	60, 120	90, 180	120, 240
Rev. Read:	Yes	Yes	Yes	Yes
No. of Tracks:	9	9	9	9
Controller:	3518-2 3518-3 3528-2 3528-3			

TAPE TRANSPORT



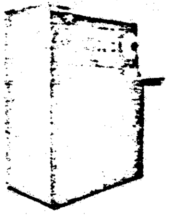
Summary Description	604
Tape Speed, IPS:	75
Densities:	200, 556, 800 bpi
Transfer Rate, KC:	15, 41.7, 60
Rev. Read:	Yes
No. of Tracks:	7
Controller:	3228 3229 3421 3423

TAPE TRANSPORT



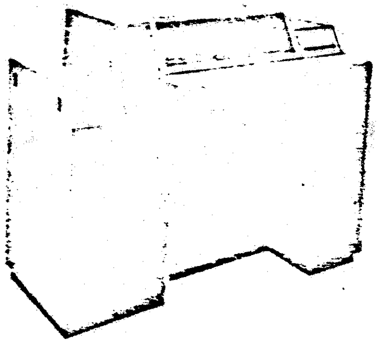
Summary Description	607
Tape Speed, IPS:	150
Densities:	200, 556, 800 bpi
Transfer Rate, KC:	30, 83.3, 120
Rev. Read:	Yes
No. of Tracks:	7
Controller:	3228 3229 3421 3423 3522 3621 3625 3626 3623 3624

TAPE CERTIFIER



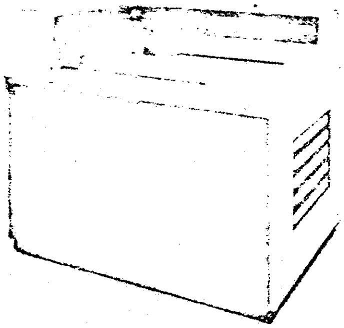
Summary Description	686
Tape Size:	½-inch
Tape Speed, IPS:	150
Densities:	556, 800, 1600 bpi or 3200 flux change per inch plus one additional density optional
Format:	7- or 9-track
Microscope:	10x or 20x

LINE PRINTER



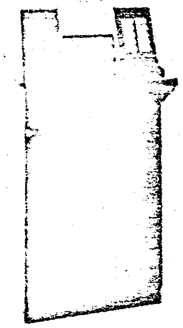
Summary Description	512
Controller:	3555
No. of Printers per Controller:	1
Printing Speed:	1200 lines per minute with 48 character font
No. of Characters per Line:	136
No. of Printable Characters:	64
Horizontal Spacing:	10 characters per inch
Vertical Spacing:	6 or 8 lines per inch
Form Advance Rate:	70 inches per second nominal
Form Width:	3 to 21 inches wide
Form Length:	Up to 22 inches long
No. of Copies:	Up to 6 copies

CARD READER



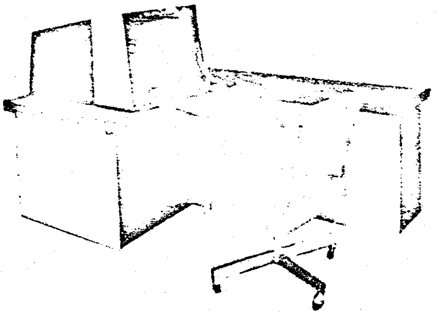
Summary Description	405
Controllers:	3447 (for one 405) 3649 (for two 405's)
Number of Controllers per Channel:	8
Number of Card Readers per Controller:	1
Card Read Speed:	1200 cpm
Read Check:	Light Dark Probe
Input Stacker Capacity:	4000 Cards
Output Stacker Capacity:	4000 Cards
Secondary Output Stacker Capacity:	240 Cards

CARD PUNCH



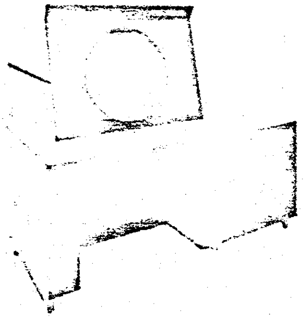
Summary Description	415
Controllers:	3446 (for one 415) 3644 (for two 415's)
Number of Controllers per Channel:	8
Number of Card Punches per Controller:	1
Card Punching Speed:	250 cpm
Punch Modes:	Row Punching
Input Hopper Capacity:	1200 Cards
Output Hopper Capacity:	1500 Cards

GRAPHICS TERMINAL SUBSYSTEM



Summary Description	240
Entry/Display:	12 inches x 12 inches
Capacity (12 bit words):	4,000 to 12,000 words
Memory Cycle Time:	1.2 microseconds
On-Line or Off-Line Operation:	Yes

DIGIGRAPHIC CONSOLE



Summary Description	274
Controller:	1744
Number of Controllers per Channel:	8
Number of Digigraphic Consoles per Controller:	1
Display Surface Area:	300 square inches with 20 inch diameter. Flat faced surface.
Display Capacity:	Up to 2000 inches of curves or up to 1800 characters of any size or font.

ROTATING MASS STORAGE - SUMMARY

MODEL	CHARACTER CAPACITY	CHAR/SEC TRANSFER RATE	POSITIONING TIME MSEC	ACCESS MECHANISMS
813 Disk File	133 million	196KC	25 - 110	1
814 Disk File	266 million	196KC	25 - 110	2
821-1 Data File	419 million	420KC	25 - 145	1
✓ 821-2 Data File	838 million	420KC	25 - 145	2
✓ 841-3 Multiple Disk File	107 million	420KC	25 - 135	3
841-4 Multiple Disk File	143 million	420KC	25 - 135	4
841-5 Multiple Disk File	179 million	420KC	25 - 135	5
841-6 Multiple Disk File	214 million	420KC	25 - 135	6
841-7 Multiple Disk File	250 million	420KC	25 - 135	7
841-8 Multiple Disk File	286 million	420KC	25 - 135	8
842-2 Disk Drive	116 million	420KC	10 - 55	1
854 Disk Drive	8.2 million	208KC	30 - 165	1
863 Drum Unit	4.1 million	62.5 - 2000KC	17	fixed heads
865 Drum Unit	8.3 million	1MC	17	fixed heads
6638 Disk File	131 million	1.68MC	25 - 110	2
6638-2 Disk File	65 million	1.68MC	25 - 110	1

CHANNEL CHARACTERISTICS FOR CYBER 72,73,74 AND 6000 SERIES COMPUTERS

- Any PP can read/write any channel

- Software interlock via M.RCH request to monitor provides but does not guarantee disciplined use of a channel by only one peripheral processor

- Channels can be "hung" by wrong code sequences

- PPU's are a kind of higher level processor (HLP) in contrast to a controller/equipment which has no processor.

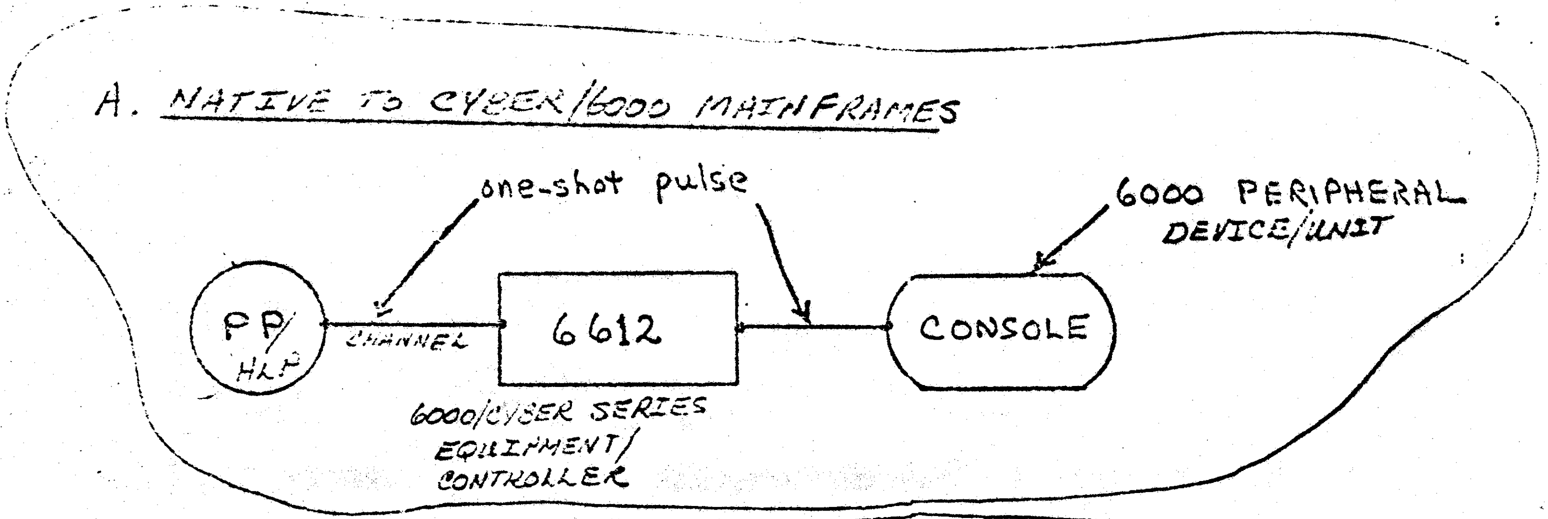
- The communication language spoken between PPU's and external controllers is composed of functions being sent by the PPU/HLP across the channel and received and acknowledge by the controller by means of a returned status back across the channel to the PPU/HLP

- The purpose of the communication language is to send data words (12-bits) across the channels

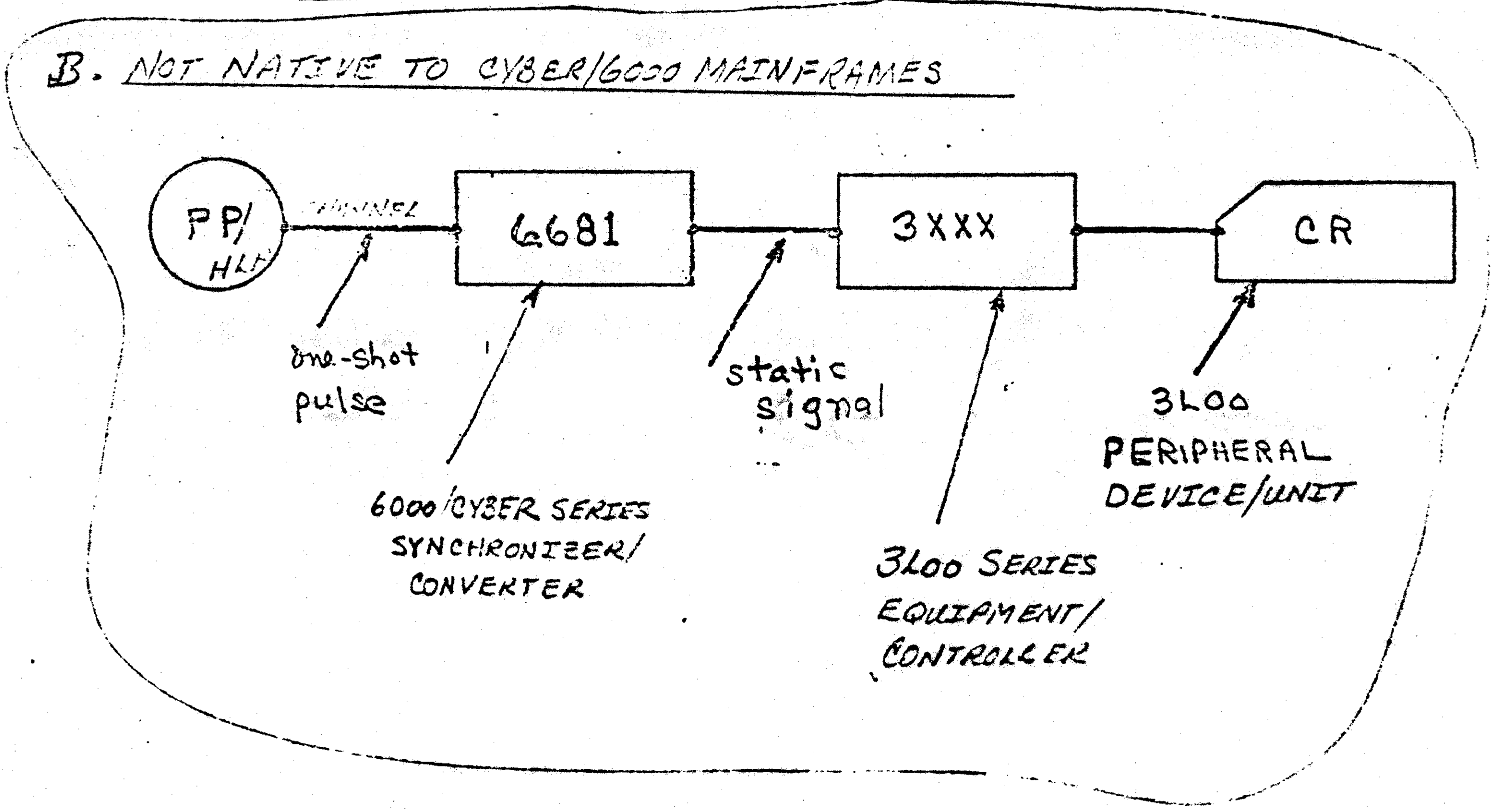
- The channel is composed of a 12-bit register to hold either a function or data and two flags to express active/inactive and full/empty states

THE TWO POSSIBLE EQUIPMENT CONFIGURATIONS FOR I/O

A. NATIVE TO CYBER/6000 MAINFRAMES



B. NOT NATIVE TO CYBER/6000 MAINFRAMES

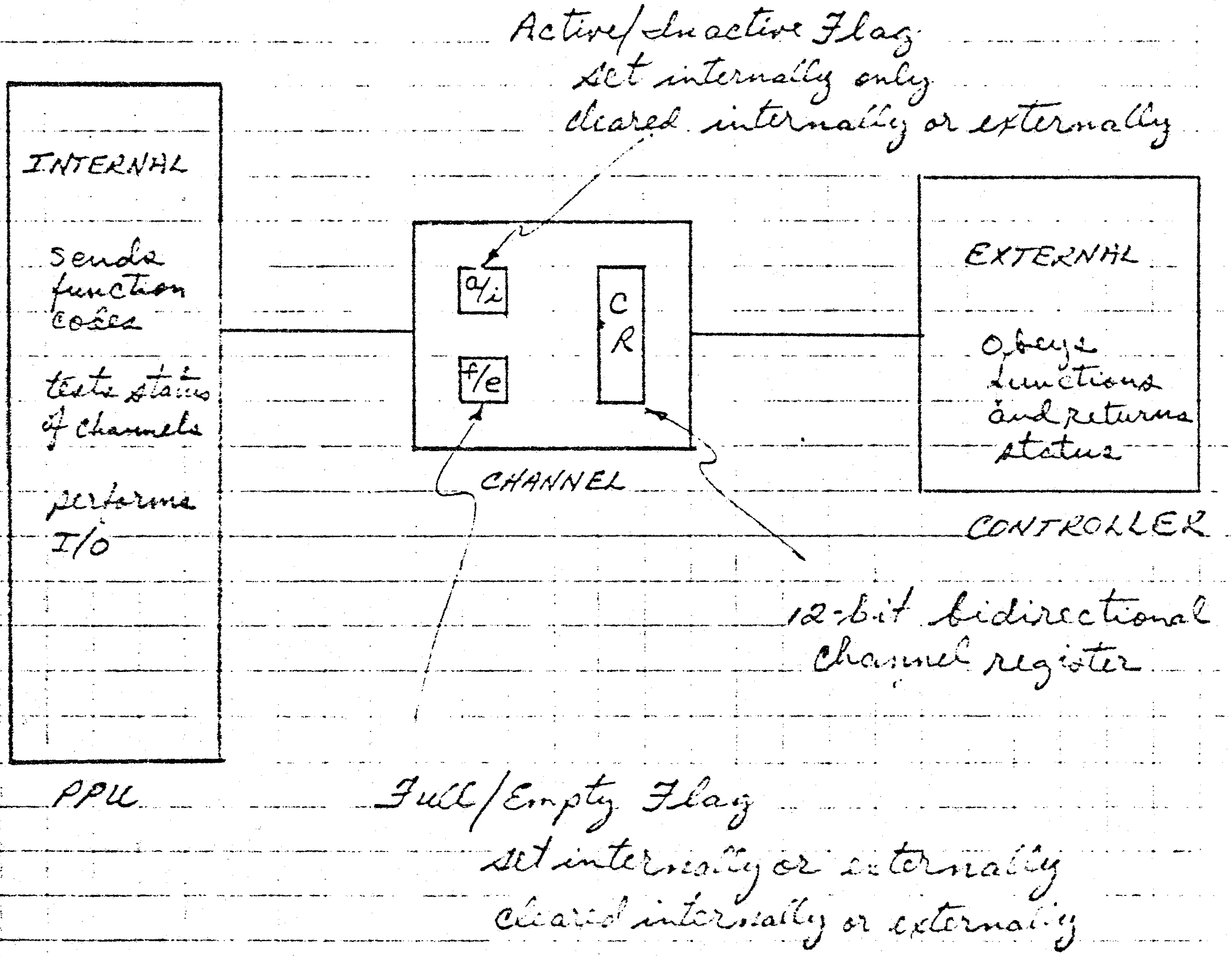


Reference:

I/O SPECS. CH 1 & 4

Pub. No. 60352500

MAIN COMPONENTS OF A CYBER/6000
DATA CHANNEL

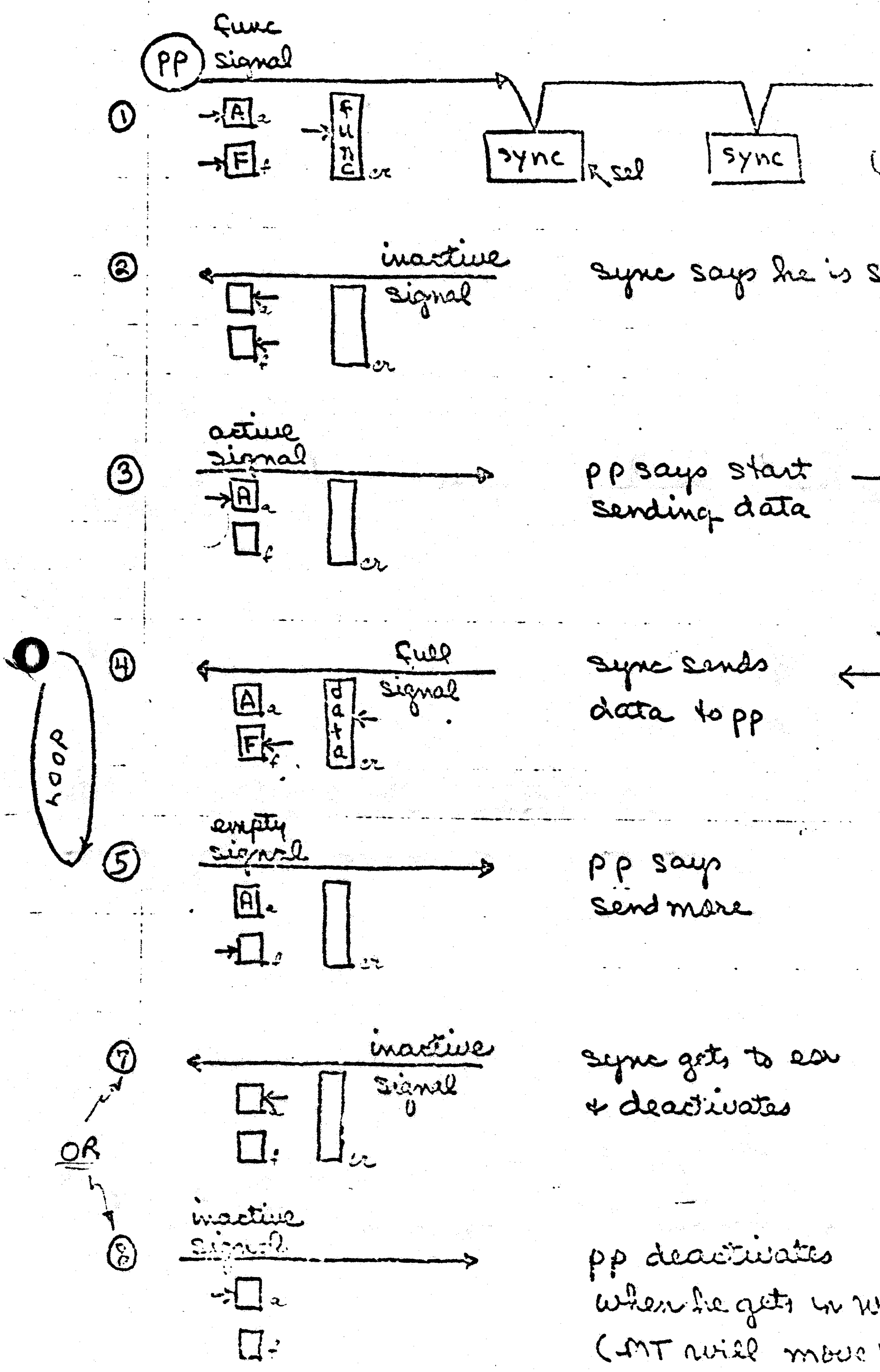


Reference:
I/O SPECS, 2-1,5
60352500

How Data Input Works

Data Input

I p.2-1



① select sync (which one on ch) + mode *input*

② sync says he is selected

③ pp says start sending data ⟶ sync tells device to do it

④ sync sends data to pp ⟵ device sends to sync

⑤ pp says send more

⑦ sync gets to eor + deactivates

⑧ pp deactivates when he gets n words desired (MT will move to EOR)

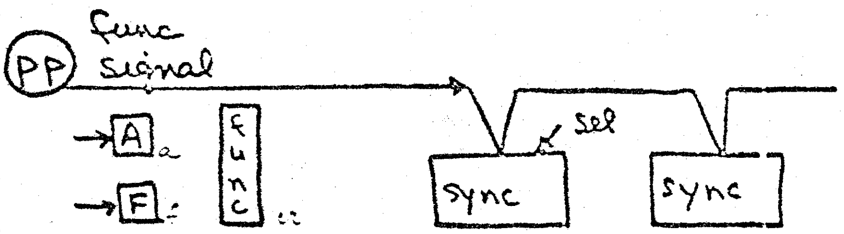
10-17

How Status Request Works

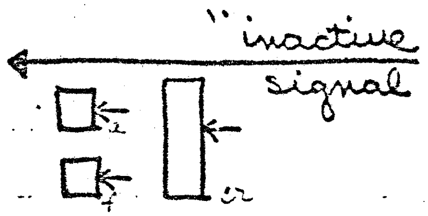
Status

I p. 20

FNC 7200, ch
7300, ch

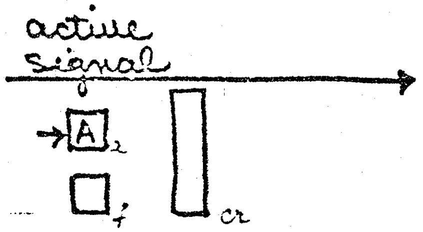
①


sel sync
mode: status

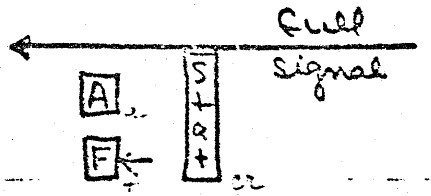
②


sync says he is selected

ACN ch

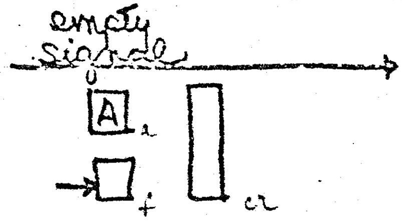
③


PP says send me status !

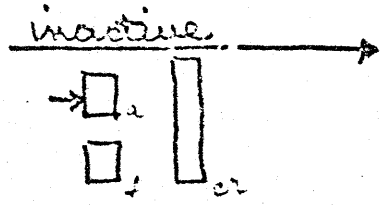
④


sync sends status

IAN ch

⑤


PP says he got it

⑥


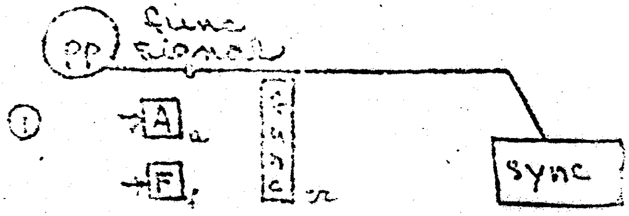
PP deactivates ch + sync

DCN ch

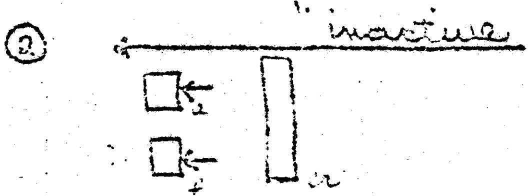
How Data Output Works

Output

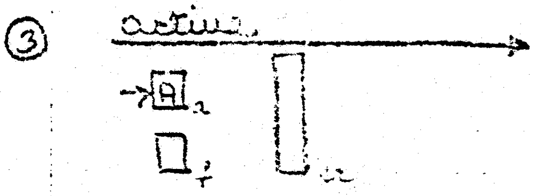
I p. 2-3



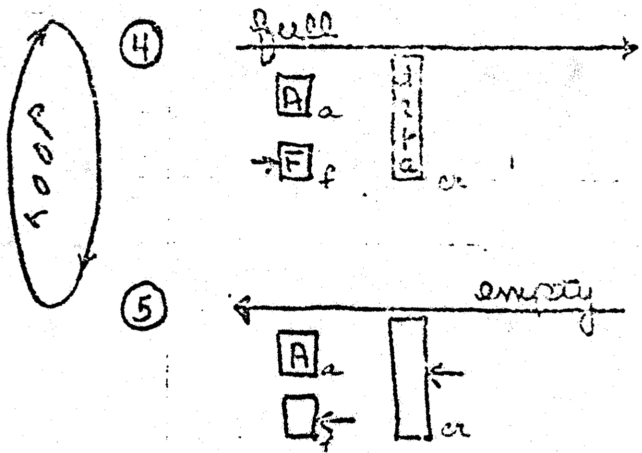
Select sync - mode: output



sync says he is selected



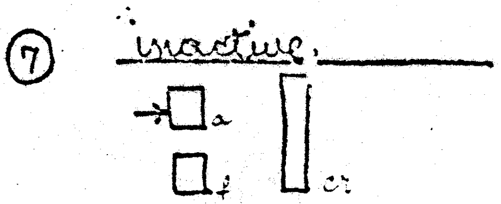
pp says here comes the data



pp sends the data + tells sync.

accepts data + asks for more

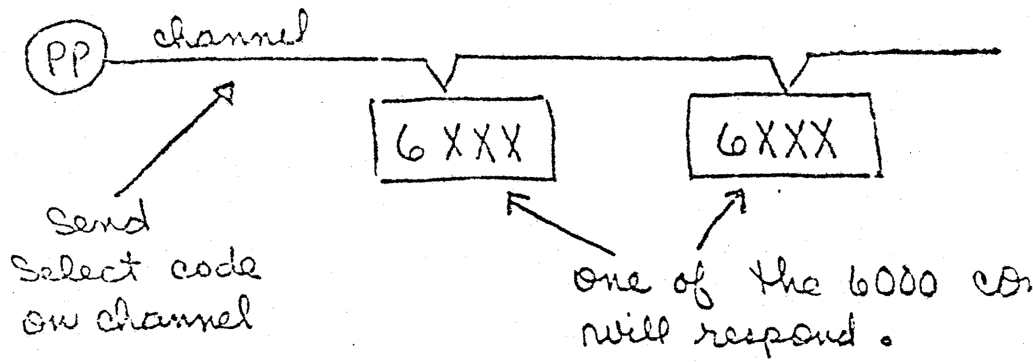
WAIT FOR EMPTY SIGNAL



pp deactivates sync. when thru

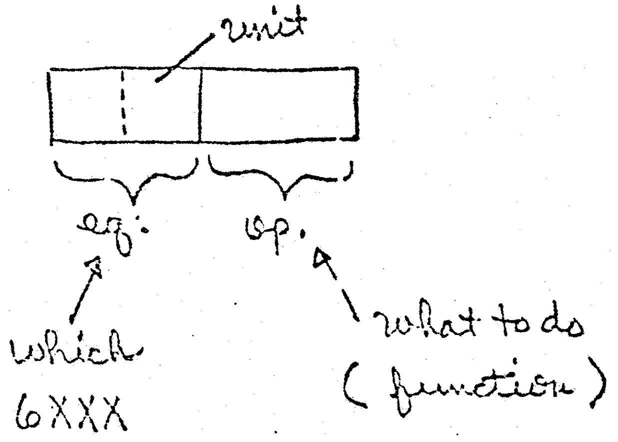
(Selecting, Connecting, and Functioning)
CYBER70/6000- Type Equipment Access

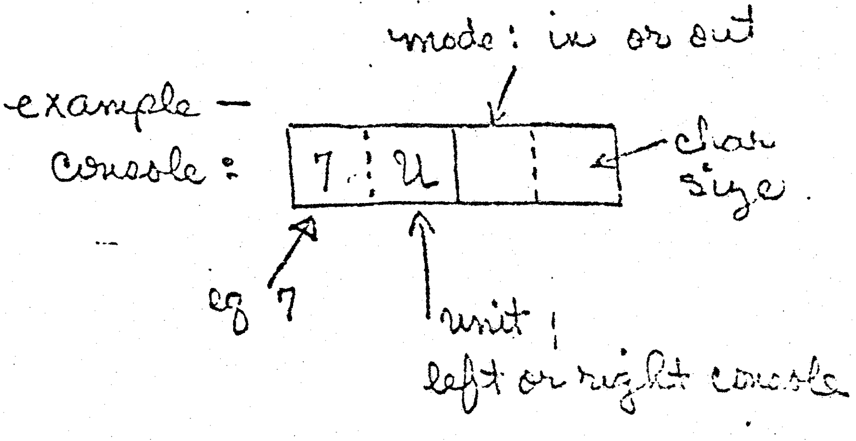
The SELECT code selects, connects, and functions a 6000 eq.



One of the 6000 controllers on the ch will respond.
He is selected and connected + functioned

ie
select code:



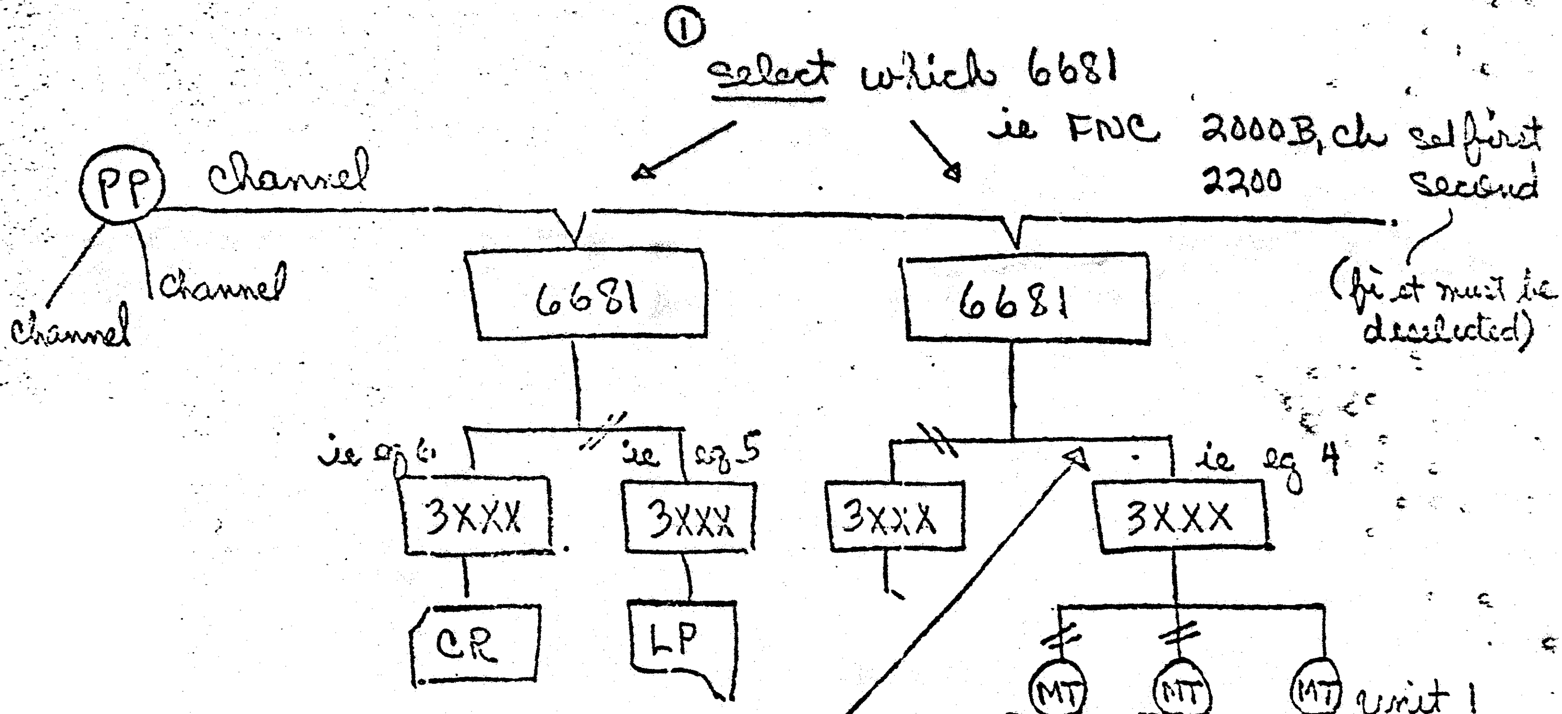


6/71

(Selecting, Connecting, and Functioning)
3000-Type Equipment Access

PE-6681 p.9

- We **Select** a 6681 interface
- we **connect** and then **function** a 3000 eq. } thru the selected 6681



PE-6681 p.12

② **Connect** the 3XXX
ie FNC 4001, ch
connects MT eq 4 unit 1
(thru the selected 6681)

③ then
function the 3XXX
(tell it what to do)
ie FNC 0XXX ch

func code
to start motion
page eject
etc

FNC
goes thru the
selected 6681 &
connected 3XXX

(look at codes under
3000 eq.)

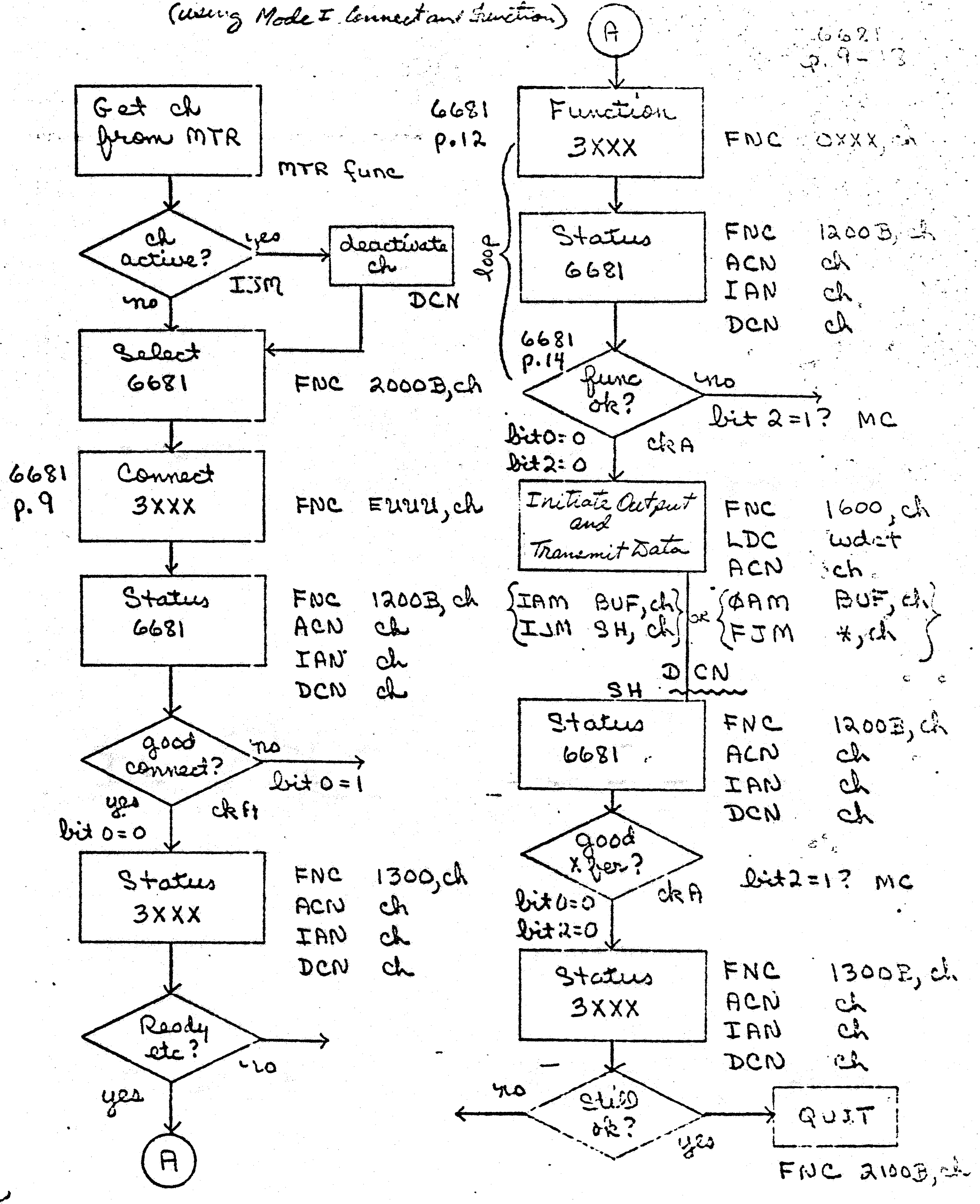
INPUT / OUTPUT INSTRUCTIONS

AJM IJM		check CHANNEL ACTIVE flag
FJM EJM		check CHANNEL FULL flag
ACN	*	activate channel
DCN	*	deactivate channel
FNC	*	send function code
FAN	*	send function from A
IAN	*	input a word to A
IAM	**	input a block to PP memory
OAN	*	output a word from A
OAM	**	output a block from PP memory

* THESE INSTRUCTIONS CAN HANG THE PP
**THESE INSTRUCTIONS CAN BE A NOP

2/72

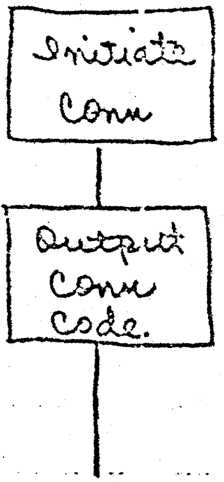
PROGRAMMING THE PERIPHERAL
(using Mode I Connect and Function)



cs
2/72

DIFFERENCES BETWEEN MODE I & II

Mode II Conn

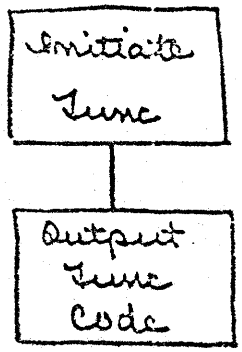


```
FNC  1000,ch
        0-3
LDC  Euuu  ie 1001
ACN  ch
OAN  ch
FJM  *,ch
DCN  ch
```

MODE I CONN

```
FNC   4000,ch
```

Mode II Func

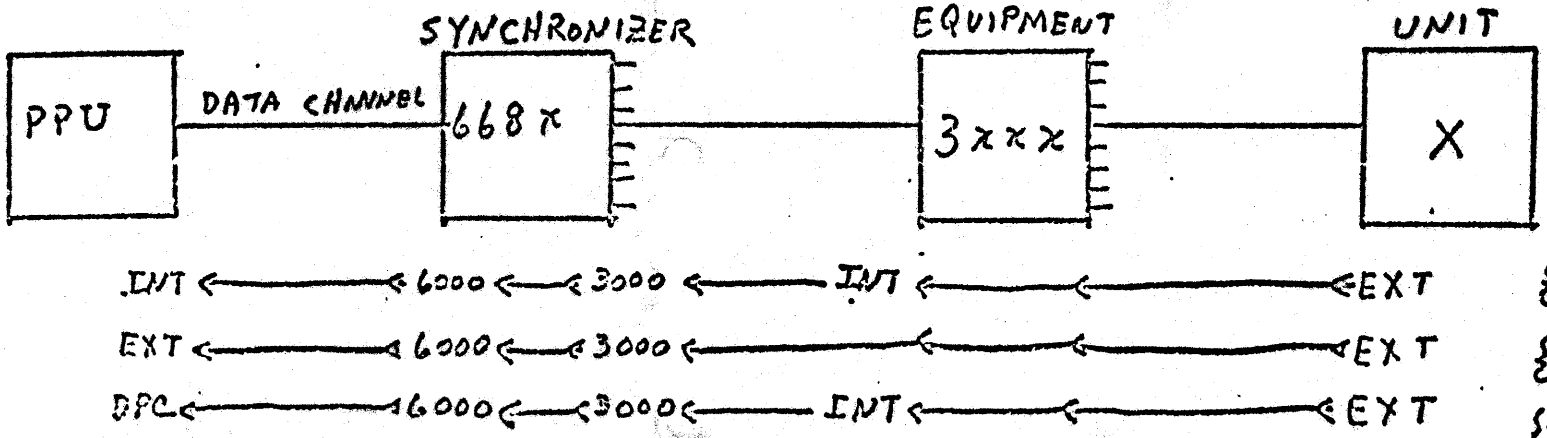


```
FNC  1100,ch
LDC  XXXX,ch   ie 5000
ACN  ch
OAN  ch
FJM  *,ch
DCN  ch
```

MODE I FUNC

```
FNC  0XXX,ch
```

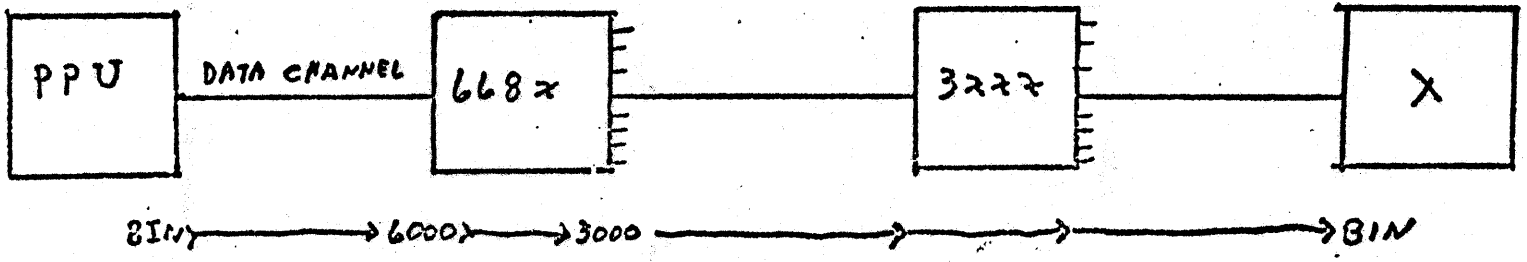
CODED INPUT - OUTPUT



(REVERSE ARROWS FOR OUTPUT)

FUNCTION CODES		SYNCHRONIZE TYPE	
INPUT	OUTPUT		
1400 1500	1600	6684	661
1401 1501	1601	6684	661
1410 1510	1610	6684	

BINARY INPUT - OUTPUT



1400 1500	1600	6684	668.

[illegible]

([illegible] 3400 [illegible] Manual)

If the number of "1" bits in a data byte is even, a "1" is transmitted on the parity line to make the total number of "1" bits odd.* If the number of "1" bits in the data byte is odd, a "1" is not transmitted on the parity line.

A transmission parity error exists if the total number of "1" bits transmitted on the 12 data lines plus the parity line is even, indicating that a bit has been lost or picked up.

Parity Error in a Connect Code: If a parity error is detected in a Connect code, the device does not connect** and neither a Reject nor a Reply is returned to the data channel. Instead, a red indicator in the Equipment Number switch of each equipment detecting the error lights. These parity error conditions must be cleared by either a Channel Clear or a Master Clear prior to a new connect attempt.

Parity Error in a Function Code: If a parity error is detected, the requested functions are not performed, a Parity Error signal is returned to the data channel, and a red indicator in the Equipment Number switch lights. Since neither a Reply nor a Reject is returned to the data channel, the central processor generates an Internal Reject after a wait of 100 microseconds. These parity error indications must be cleared by a Channel Clear or a Master Clear.*** The equipment must then be reconnected before a new function code can be examined by the controller.

Parity Error in Output Data: If a transmission parity error is detected by the control during a Write operation, the control lights a red indicator in its Equipment Number switch and sends both a Reply and a Parity Error signal to the data channel. The data is written on tape. All operations continue**** unless appropriate programming steps have been taken to sense the Parity Error signal and to rewrite the data. These parity error indications must be cleared by either a Channel Clear or a Master Clear. The equipment must then be reconnected and the appropriate functions reselected prior to the new output.

---

*Do not confuse this line with the parity error line.

**If the device is connected, it automatically disconnects.

***Though operations may continue normally, the validity of a new function code and/or data prior to a Master Clear or Channel Clear is questionable.

****The validity of the data received from this point until a Channel Clear or Master Clear is questionable.

Parity Errors in Input Data: Transmission parity errors may be detected by the data channel on data received from the equipment. If a parity error is detected, a parity error bit in the data channel is set and a Parity Error indicator on either the channel or console lights. The faulty data is entered into either core storage or the A register. All operations continue* unless appropriate programming steps have been taken to sense for the set bit and reread the data. These parity error indications may be cleared by a Channel Clear or a Master Clear issued by any 3000 Series system and by a new Read or Write from a 3100/3200 system. Following a Channel Clear or a Master Clear, the equipment must be reconnected and the appropriate functions reselected prior to a new input.

not [illegible]

Input/Output Parity Error Bit in the Data Channel: The input/output parity error bit is set whenever a transmission parity error is detected. If the error is detected by the equipment, the bit is set by the Parity Error signal.

In 3400/3600/3800 systems, an Interrupt signal may be generated when this bit sets. If the interrupt system has not been set to detect the setting of this bit, the bit may be sensed to detect parity error conditions.

In 3100/3200 systems, this bit must be sensed if transmission parity error conditions are to be detected by the central processor. Xmis. PE not included in abn. EOP

Refer to the appropriate system reference manual for more information on the input/output parity error bit.

p.C-10 322X

2 ms on 606+607

3 ms on 601/603 + 342X

Equipment Parity Checking

Each character, whether BCD or binary, transmitted between a control and a unit is checked for correct parity. For BCD characters, correct parity is even, and for binary characters it is odd. During a Write operation, the control adds the correct parity bit to each character and relays it to the tape unit. Approximately 2 milliseconds after writing, a vertical parity error check is made. This time interval is used to check-read the tape and transmit the data back to the control. At the conclusion of a record, a record check character is written. This character is used for longitudinal parity checking. During a Read, vertical and longitudinal parity checks are made by the control when the appropriate data is received.

---

*The validity of the data received from this point until the indicators are cleared is questionable.

2. How to Request a Channel

```
LDN    0
STD    D.T2
LDC    0506B
RJM    R.RCH
```

a. What is the first choice channel? ______________

b. Where will the assigned channel number be found? ______________

3. Another Way to Request a Channel:

```
LDC    0506B
STD    D.T1
STD    D.T4
LDN    0
STD    D.T2
LDN    M.RCH
RJM    R.MTR
```

a. Is there any difference in the reply which will be received by these two sets of code?

4. How to Request a Channel and Ask for Immediate Reply:

```
LDC   0506B
STD   D.T1
LDN   0
STD   D.T2
STD   D.T4
LDN   M.RCH
RJM   R.MTR
```

D.T4 being 0 will indicate to MTR to reply immediately if the desired channel is not available. R.RCH would wait until the channel could be assigned.

---

5. How to Drop a Channel:

a.

```
RJM   R.DCH
```

where must the channel number be? ____________

b.

```
LDN   M.DCH
RJM   R.MTR
```

Where must the channel number be? ____________

when requesting alternate channels,

6. the driver code must be changed to reflect which channel MTR assigned

```
X       EQU    5
CH      EQU    D.Z1        addr to contain ch. no.
               }
        LDN    0           clear 5 bytes
        CRD    D.T0
        ~~LDN~~
        LDC    0605B       Request 5 or 6
        RJM    R.RCH
        LDD    D.T1        get ch received
        STD    CH
        LDC    LIST
        RJM    R.STB       go change ch no.
               }
X1      FNC    2000B,X  )
               }        |
X2      FNC    6003B,X  }  driver code
               }        |  (assembled for
X3      FNC    1200B,X  |  channel 5 )
               }        )


LIST    VFD    12/CH       addr containing ch no.
                                       (direct cell)
        VFD    12/X1    )
        VFD    12/X2    }  all driver addresses
        VFD    12/X3    |  containing ch no.
               }        )
        DATA   0
```

The channel no. must be changed in *every* instruction using it. The ch. no. will always be in the ch portion of the instr.

(10-30)

77	ch
2000	

ie FNC 2000B, X

7. A Better Way to Handle Alternate Channels

```
X       SET     6          first choice ch
CH      EQU     D.Z1       loc. to hold current ch
        LDN     X          initialize CH
        STD     CH
LOOP    LDN     0
        STD     D.T2
        LDC     0506B      request ch 5 or 6
        RJM     R.RCH
        LDD     D.T1       get ch received
X0      SBD     CH         see if same as last time
        ZJN     X1
        LDD     D.T1       reset CH if different
        STD     CH
        LDC     LIST
        RJM     R.STB      go change
X1      FNC     2000B,X   \
         {                 |
X2      FNC     6003B,X    } driver code
                           |
         {                /

LIST    VFD     12/CH
        VFD     12/X1     \
        VFD     12/X2      } addresses where
         {                 } ch needs changing
                          /
        DATA    0
```

cw 9/71

cs
jb
2/72

STB from
Intercom
Driver

```
STB.C.PPFWA                                                    COMPASS - VER 2.     02/17/72  17.03.26.       PAGE    2

                                                  IDENT   STB.C.PPFWA
                                                  PERIPH
                                                  SST
1000                                              ORG     C.PPFWA
                                                  LIST    G

1000         0100 1006                            LJM     INIT

                                                  USE     CHINIT
1002         7776                       CHANTAB   VFD     12/CHANNEL

                                        * DEFINE MACROS TO CREATE TABLE FOR R.STB
                                         DCN.     MACRO
                                                  LOCAL   Q
                                         Q        DCN     0
                                                  USE     CHINIT
                                                  VFD     12/Q
                                                  USE     *
                                         DCN.     ENDM
                                         FAN.     MACRO
                                                  LOCAL   Q
                                         Q        FAN     0
                                                  USE     CHINIT
                                                  VFD     12/Q
                                                  USE     *
                                                  ENDM

                                                  USE     INIT
1005                                     INIT     BSS     0
1006         2000 1002                            LDC     CHANTAB
1010         0200 0663                            RJM     R.STB
                                        * REST OF CODE
1012         7500                                 DCN.
1003         1012                                 VFD     12/Q?000001                                     DCN.      .1
1013         7600                                 FAN.
1004         1013                                 VFD     12/Q?000002                                     FAN.      .1

                         7776            CHANNEL  EQU     7776B             LOCATION WHERE CHANNEL NUMBER
                                                                             WILL BE PLACED

                                                                            END OF CODE
                                                  USE     CHINIT
1005         0000                                 DATA    0

1014                                              END
                         54202          STORAGE USED              51 STATEMENTS       713 SYMBOLS       000002 INVENTED SYMBOLS
                                        6403 ASSEMBLY          0.737 SECONDS            9 REFERENCES
```

FROM
INTERCOM
DRIVER

10-32

BUILD CHANNEL INIT. TABLE FOR R.STB

7776	channel no.

	1000	0100	LJM	ZERO BLØCK
	1001	1006	INIT	
CHANTAB	1002	7776		CHINIT BLOCK
	1003	1012		
	1004	1013		
	1005	0000		
INIT	1006	2000	LDC	INIT BLOCK
	1007	1002	CHANTAB	
	1010	0200	RJM	
	1011	0663	R.STB	
	1012	7500	DCN.	
	1013	7600	FAN.	

R.RCH

SCOPE 3.3

R.RCH {560}

Calling Sequence: Load channel number

RJM R.RCH

The channel numbers contained in the A register will be stored in byte D.T1, monitor function M.RCH inserted in D.T0, and D.T0 - D.T4 written to the output register for that PP. Channels will be assigned by MTR on the following priority basis:

D.T0	D.T1	D.T2	D.T3	D.T4
	2 | 1	4 | 3		

If alternate channels are specified MTR will stop looking for alternate channels upon sensing 6 bits of zero. Thus, if one alternate channel is desired, the programmer must clear D.T2 before entering R.RCH so the search will be terminated at that point. The procedure for requesting channel 12 with alternate channel 13 would be:

```
LDN  0

STD  D.T2

LDC  1312B

RJM  R.RCH
```

Monitor will stop looking for alternate channels after four channels have been investigated or 6 bits of zero are detected.

When R.RCH is used, D.T4 is automatically set nonzero; in this case, the function is not considered complete {i.e., output register is not cleared} until a channel can be assigned. When complete, byte 0 of the output register is cleared, and if IP.CHTIM≠0, bytes 3 and 4 of the output register are set to channel start time in seconds and milliseconds respectively for hardware channel requests. PP Resident will then save these values in the lower two bytes of its status word {CHSEC and CHMSEC}.

```
                                   *    R.RCH     RESERVE CHANNEL                                  STL     479
                                   *                                                               STL     480
                                   *    CALLING SEQUENCE                                           STL     481
                                   *                                                               STL     482
                                   *    LOAD      CHANNEL NUMBER                                   STL     483
                                   *    RJM       R.RCH                                            STL     484
                                   *                                                               STL     485
                                   *    RETURNS WHEN ASSIGNMENT COMPLETE                           STL     486
                                   *                                                               STL     487

      0560                         R.RCH    ENM       X                                            STL     489
      0562      3411                        STD       D.T1          CHANNEL NUMBER                 STL     490
      0563      1402                        LDN       M.RCH                                        STL     491
      0564      3414                        STD       D.T4          DO NOT RETURN WITHOUT RESERVATION STL    492
      0565      0200 0516                   RJM       R.MTR         ISSUE REQUEST CHANNEL          STL     493
                                            IFNE      IP.CHTIM,0,4                                 STL     494
      0567      3013                        LDD       D.T3          GET                            STL     495
      0570      5400 0102                   STM       CHMSEC           START TIME                  STL     496
      0572      3012                        LDD       D.T2                        AND SAVE         STL     497
      0573      5400 0101                   STM       CHSEC                          IN STATUS WORD STL    498
      0575      0362                        UJN       R.RCHX                                       STL     499
                                            IFEQ      IP.CHTIM,0,1                                 STL     500
                                            BSSZ      R.DCH-1-*                                    STL     501
```

R.RCH

10-35

SCOPE
3.3

M.RCH - Request Channel

{0002,BBAA,DDCC,****,RRRR}

AA = 1st choice channel number
BB = 2nd choice channel number
CC = 3rd choice channel number
DD = 4th choice channel number

RRRR = 0000 Request immediate reply
RRRR ≠ 0000 No reply until a requested channel has been reserved.

When channel zero is requested, it must be field AA. When BB, CC or DD is zero, it is assumed that this is not a channel request and that there are no alternate choices beyond it.

If none of the requested channels are available and an immediate reply is requested, MTR will set bytes 0 and 4 of the PPU output register to zero.

When a channel is granted, the number of that channel will be returned in the PPU output register byte 1 (location of AA). Byte 4 will be set to a non-zero value.

On exit, if a channel has been reserved, the output register will look like: 0000 XXXX TTTT TTTT YYYY where

XXXX is channel number
TTTT TTTT is
- a. the current time if the IOTIME mods are assembled on and the PP is not not at control point zero, and the channel number is a hardware channel.
- b. 0 if IOTIME mods are on and the PP is at control point zero.
- c. the information from the channel status word if the IOTIME mods are assembled off.

YYYY is the PP input register address

M.REQP - Request Equipment
{0022,EEEE,****,****,***N}

The parameter EEEE consists of two display-coded characters: if numeric it gives an EST ordinal
if alphabetic it defines an equipment type

MTR will search the Equipment Status Table (EST) to locate the appropriate EST entry. This entry is updated to reflect the assignment to the control point of the requesting PPU or to control point N if the PPU is assigned to control point zero. Finally, MTR places the assigned equipment ordinal in the first byte of the PPU message buffer. If the equipment is not available, a zero byte is returned.

M.DEQP - Drop Equipment
{0023,EEEE,****,****,***N}

MTR drops equipment ordinal EEEE from the control point and updates the EST entry to indicate that this equipment is free for reassignment. There is no check by MTR to ensure that the dropped equipment was assigned to this control point. The parameter N gives the control point number to be considered if the requesting PP is attached to control point zero, otherwise, it is irrelevant.

10-37

M.REQP, M.DEQP

SCOPE
3.3

R.DCH

R.DCH {57$_8$}

Calling Sequence: LOAD channel number

RJM R.DCH

R.DCH will cause the specified channel to be dropped.

Since more than one PP can request the same channel at the same time, it is necessary to use a MTR request to reserve a channel.

The only PP which can release a channel, however, is the PP which reserved it and there is no need for an interlock. To release a channel reservation, a PP program loads the number of the channel into the A Register and executes a return jump to the PP Resident subroutine R.DCH. If the channel is assigned to the PP, R.DCH will modify the Channel Status Table entry for the channel to indicate that the channel is free. ~~If a PP calls R.DCH to release a channel it has not reserved, R.DCH will issue an illegal MTR function {77$_8$} which will hang the PP.~~ The time in seconds and milliseconds is taken and stored in a byte in status word.

not 3.

```
                                      *     R.DCH      DROP CHANNEL                                        STL      503
                                      *                                                                    STL      504
                                      *     CALLING SEQUENCE                                               STL      505
                                      *                                                                    STL      506
                                      *     LOAD       CHANNEL NUMBER                                      STL      507
                                      *     RJM        R.DCH                                               STL      508
                                      *                                                                    STL      509

      0576                            R.DCH     ENM     X                                                  STL      511
      0600     2100 0000                        ADC     **            T.CST INSERTED AT DEADSTART          STL      512
                           0000601    DCH1      EQU     *-1                                                STL      513
      0602     6010                             CRD     D.T0                                               STL      514
      0603     3414                             STD     D.T4          RELEASE THE CHANNEL                  STL      515
      0604     6210                             CWD     D.T0          WRITE OUT UPDATED CST ENTRY          STL      516
                                                IFEQ    IP.CHTIM,0,1                                       STL      517
                                                UJN     R.DCHX                                             STL      518
                                                IFNE    IP.CHTIM,0                                         STL      519
      0605     3076                             LDD     D.CPAD                                             STL      520
      0606     0467                             ZJN     R.DCHX        EXIT IF ASSIGNED TO CNT. PT. ZERO    STL      521
      0607     3011                             LDD     D.T1                                               STL      522
      0610     1714                             SBN     14B                                                STL      523
      0611     0664                             PJN     R.DCHX        EXIT IF NOT A HARDWARE CHANNEL       STL      524
      0612                                      LDK     T.MSC                                              STL      525
      0613     6010                             CRD     D.T0          OTHERWISE READ CURRENT SEC/MSEC CLOCK STL     526
      0614     3012                             LDD     D.T0+MSEC                                          STL      527
      0615     5200 0102                        SBM     CHMSEC        COMPUTE NUMBER OF MSEC USED          STL      528
      0617     5400 0102                        STM     CHMSEC                                             STL      529
      0621     0613                             PJN     NOCARRY       TEST FOR MSEC OVERFLOW               STL      530
      0622     2100 1750                        ADC     D1000         ADJUST IF MILLISECOND  OVERFLOW      STL      531
      0624     5400 0102                        STM     CHMSEC                                             STL      532
      0626     1501                             LCN     1                                                  STL      533
      0627     3111                             ADD     D.T0+SEC                                           STL      534
      0630     0302                             UJN     CARRY                                              STL      535
      0631     3011                   NOCARRY   LDD     D.T0+SEC                                           STL      536
      0632     5200 0101              CARRY     SBM     CHSEC         COMPUTE NUMBER OF SECONDS USED       STL      537
      0634     0603                             PJN     *+3                                                STL      538
      0635     2101 0000                        ADC     10000B        ADJUST IF CLOCK OVERFLOW             STL      539
      0637     5400 0101                        STM     CHSEC                                              STL      540
      0641     3077                             LDD     D.PPSTAT                                           STL      541
      0642     6276                             CWD     D.CPAD        WRITE UPDATED STATUS WORD TO C. M.   STL      542
      0643     0100 0576                        LJM     R.DCHX        EXIT ROUTINE                         STL      543

                           0000001    SEC       EQU     1             BYTE IN STATUS WORD FOR SECONDS      STL      545
                           0000002    MSEC      EQU     2             BYTE IN STATUS WORD FOR MILLISECONDS STL      546
                           0001750    D1000     EQU     1000D         CONSTANT FOR MILLISECOND OVERFLOW    STL      547
                                                ENDIF                                                      STL      548
                                                IFEQ    IP.CHTIM,0,1                                       STL      549
                                                BSSZ    R.STBMSK-3-*                                       STL      550
```

10-39

R.STB

R.STB {645}

```
Calling Sequence:   Load    L{List}

                    RJM     R.STB
```

where list has the form

```
        L {byte}

        L {word 1}

        L {word 2}

           .
           .
           .

        L {word n}

        zero
```

An entry point to R.STB called R.STBMSK is the address of the mask "anded" with each word in the list before the word is "exclusive ored" with the byte. This mask is initially 7700B and this value should be restored by any routine which substitutes an alternate mask. R.STB is used primarily to substitute channel numbers in driver overlays.

All the PP hardware instructions used for I/O contain a field which specifies the number of the channel over which the I/O is to take place. For example the instruction

```
IAM   BUFF,5
```

would be used to read data from hardware channel five into the PP starting at location BUFF.

When a programmer is coding a PP program, he normally does not know what channel will be used for the I/O. The channel number is normally obtained by the PP program from an entry in the EST table. For this reason the above I/O instruction would be written as follows:

```
IAM   BUFF,**
```

The double asterisks indicate that the value will be filled in by the program itself when it is executed. COMPASS assembles double asterisks as a zero.

Since the channel number goes into the first {or only} byte of an instruction along with the OP code, the first byte of the instruction would contain 7100_8 {the OP code for an IAM is 71_8}. The second byte of the instruction would contain the value of BUFF. When the PP program is called, and determines the channel number, it must modify all the I/O instructions in itself so that the first byte of each instruction contains the OP code followed by the correct channel number. Normally there would be a list somewhere in the program giving the addresses of all instructions to be modified in this way.

R.STB

SCOPE
3.3

The PP resident subroutine R.STB can be called to insert a channel number into one or more instructions, whether or not the fields to be altered previously contain zero. Before return-jumping to R.STB, the program loads the address of a list in the A register. The first byte in this list contains the address of some other PP cell that contains the new channel number. The second and following bytes of the list contain the addresses of the instruction words in which the new channel number is to be inserted. The first zero byte in the list terminates it.

Although R.STB is most often called to insert channel numbers into I/O instructions, it can also be called to perform general masking operations.

```
                                        *       R.STB     MASK BYTE INTO LISTED WORDS
                                        *
                                        *       CALLING SEQUENCE
                                        *
                                        *       LOAD      L(LIST)
                                        *       RJM       R.STB
                                        *
                                        *       WHERE LIST HAS THE FORM
                                        *
                                        *       L(BYTE),L(WORD1),L(WORD2),...,L(WORDN),0
                                        *


      0645     3412                     STB0    STD       D.T2           SAVE WORD LOCATION
      0646     4012                             LDI       D.T2           FETCH WORD
      0647     2200 7700                        LPC       7700B          CLEAR BYTE FIELD
                            0000650     R.STBMSK EQU      *-1            MAY BE USED TO CHANGE BYTE MASK
      0651     4300                             LMI       0              OR BYTE INTO WORD
      0652     4412                             STI       D.T2           RESTORE WORD
      0653     3610                     STB1    AOD       D.T0           ADVANCE IN LIST
      0654     4010                             LDI       D.T0
      0655     0567                             NJN       STB0           SENSE NOT END OF LIST

      0656                              R.STB   ENM                      STORE BYTE FOR I/O CHANNEL
      0660     3410                             STD       D.T0           SAVE LIST LOCATION
      0661     4010                             LDI       D.T0           FETCH BYTE LOCATION
      0662     3400                             STD       0
      0663     0567                             UJN       STB1
```

10-42

```
                              DSLCOM FOLLOWS                                  STL       658
                              LIST    -L                                      STL       559
                              LIST    L                                       STL       661

      0717                    BSSZ    1000B-*                                 STL       663
```

10-43

10-44

```
                                          IDENT  1RT,C.PPFWA
              003620    PROGRAM LENGTH

                        BLOCKS

      000000  003620    PROGRAM*   ABSOLUTE

                                    PERIPH



                  *          AUTHOR    T. R. STAEHLE, CONTROL DATA CORP.
                  *
                  *                    BASED ON DESIGN LOGIC SUPPLIED BY-
                  *                    H. R. GILLETTE, CONTROL DATA CORP.
                  *          REVISED BY MARY C. STEELE
                  *                     CDC--PALO ALTO
                  *                     JUNE-1969
                  *               REMOVE REEL INITIALLIZATION PROCEDURE
                  *               CALL 1RP FOR END-REEL PROCEDURES



                  *          FUNCTION
                  *          READ BCD OR BINARY RECORDS FROM HALF-INCH MAGNETIC TAPE IN THE
                  *          FORMAT OF EITHER EXTERNAL OR SCOPE STANDARD TAPES AND TRANSFER
                  *          THE DATA TO CENTRAL MEMORY.



                  *          ENTRY AND EXIT INFORMATION
                  *
                  *          D.EST - EST ENTRY - IF BIT 5 OF D.EST+1 IS SET, A 6684
                  *                              CONTRCLLER IS ASSUMED FOR BCD.
                  *          D.FNT  -SECOND AND THIRD WORDS OF THE FNT ENTRY
                  *          D.OUT  -CM BUFFER POINTER (RELATIVE) - OUT
                  *          D.IN   -CM BUFFER POINTER (RELATIVE) - IN
                  *          D.LIMIT-CM BUFFER POINTER (RELATIVE) - LIMIT
                  *          D.FIRST-CM BUFFER POINTER (RELATIVE) - FIRST
                  *          D.PPIRB-FET ADDRESS (RELATIVE)
                  *          D.RA   -REFERENCE ADDRESS
                  *          D.BA   -FET WORD 1
                  *          D.FA   -FNT WORD 2 ADDRESS
                  *          D.PPMES1- MESSAGE BUFFER ADDRESS



                  *          EXIT CONDITIONS
                  *          D.IN   MODIFIED
                  *          CODE AND STATUS REFLECTS END OF RECORD OR END OF FILE
                  *          ALSO END OF REEL OR PARITY ERROR IF APPLICABLE.
                  *          LEVEL NUMBER RETURNED IF NOT READ SKIP
                  *          PRU COUNT IS INCREMENTED
                  *
```

10-45

```
                     *            EXIT TO IDLE NORMALLY
                     *            EXIT TO 1RP IF EOR OR TAPE MARK
                     *            EXIT TO IDLE IF EOR AND UP IS SET
                     *            EXIT TO CIO TO CALL 1TF IF RDSKP AND PRUS MUST BE BYPASSED
                     *            EXIT TO IDLE WITH ERROR SET ON PARITY ERROR AND OP. DROP OPTN.
                     *            EXIT TO IDLE WITH PARITY ERROR BIT SET ON GO OPTION


                     *            MESSAGES
                     *            MTXX NOT READY - TAPE NOT READY
                     *            MTXX RESERVED  - RESERVED BIT SET IN STATUS
                     *            MTXX REJECT    - CONNECT REJECT FOR UNKNOWN REASON
                     *            MTXX XMSN PARITY ERROR - BIT SET IN CHANNEL STATUS
                     *            MTXX PARITY ERROR - UNCORRECTABLE PARITY ERROR OCCURED
                     *                                 - RESPOND GO OR DROP
                     *            MTXX DEVICE CAPACITY EXCEEDED - RECORD SIZE ON TAPE LARGER
                     *                                   THAN SPECIFIED.




                     *            .DIRECT LOCATION ASSIGNMENTS.
                     *
                     *
                                  SST
         0000005     WC           EQU     D.Z5               PRU SIZE - NUMBER OF WORDS READ
                     *
         0000004     CS           EQU     D.Z4               CONVERTER STATUS
                     *
         0000006     ST           EQU     D.Z6               EQUIPMENT STATUS
                     *
         0000007     FN           EQU     D.Z7               CURRENT FUNCTION CODE
                     *
         0000036     LEV          EQU     D.TH6              LEVEL NUMBER READ
                     *
         0000072     RLTH         EQU     D.SV2              NO. CM WORDS READ
                     *
         0000035     LA           EQU     D.TH5              ADDRESS OF LAST WORD READ.
                     *
         0000037     STATUS       EQU     D.TH7              STATUS TO BE RETURNED AT EXIT
                     *
         0000045     OVLPST       EQU     D.FR5              OVERLAP STATUS - 1=CM XMIT NOT
                     *                                      CONCURRENT WITH TAPE STARTUP
                     *
         0000046     CH           EQU     D.FR6              CURRENT CHANNEL ASSIGNMENT
                     *
         0000047     RC           EQU     D.FR7              RE-READ COUNT
                     *
         0001400     INREC        EQU     1400B              INPUT TO END OF RECORD
                     *
```

```
0000012   BKSPS     EQU    0012B          BACKSPACE
          *
0005000   BINPRU    EQU    5000B          PP WORD SIZE OF BINARY PRU
          *
          *
0001672   BEZE      EQU    1672B          BCD LINE TERMINATOR
          *
0000055   RA        EQU    D.RA
          *
0002400   PARITY    EQU    2400B          MASK TO DETECT PARITY STATUS
          *
0000020   EXTERN    EQU    20B            MASK TO DETECT EXTERNAL TAPES
          *
0000001   READY     EQU    1              READY UNIT STATUS MASK
          *
0000010   TAPMARK   EQU    10B            MASK TO DETECT TAPE MARK
          *
0000040   ENDTAPE   EQU    40B            MASK TO DETECT END OF TAPE
          *
0002400   BCDC      EQU    1280
          *
0001200   BCDPRU    EQU    BCDC/2         PP BYTES IN A CODED PRU
          *
0000002   BINMOD    EQU    2
          *
0001000   BINCM     EQU    BINPRU/5       NO. OF CM WORDS IN A PRU (BINARY)
          *
0000200   BCDCM     EQU    BCDPRU/5       NO. OF CM WORDS IN A PRU (BCD)
          *
0000040   CLRR      EQU    40B            CLEAR REVERSE READ CODE
          *
0006060   BLANKS    EQU    6060B          MASK FOR BCD BLANKS
          *
0001200   STAT      EQU    1200B          6681 STATUS REQUEST
          *
0000004   XMPER     EQU    0004B          TRANSMISSION PARITY ERROR
          *
0001700   MASCL     EQU    1700B          MASTER CLEAR
          *
0000002   BUSY      EQU    0002B          CHANNEL AND/OR READ/WRITE CONT
          *
0001300   EXSTS     EQU    1300B          EXT EQUIPMENT STATUS REQ
          *
0000104   XBCDPRU   EQU    68             PP BYTE COUNT OF X-BCD RECORD.
          *
0000016   XBCDCM    EQU    14             NUMBER OF CM WORDS PER X-BCD RECORD.
          *
0000003   DEN556    EQU    3              556 CPI DENSITY.
          *
0000007   DEN1600   EQU    7              1600 CPI DENSITY.
          *
0000044   CLCM      EQU    44B            CLEAR CONVERSION MODE.
          *
0000004   DCECNT    EQU    4              EXCESS BYTE COUNT CAUSED BY SKEW.

          *         .PP RESIDENT ENTRY POINTS.
```

10-46

```
*
*
*         R.DFM     DAYFILE MESSAGE
*         R.RCH     REQUEST CHANNEL
*         R.DCH     DROP CHANNEL
*         R.MTR     MONITOR REQUEST
*                M.PAUSE    PAUSE
*                M.DPP      PP DROP
*         CALL TO CIOCOM (CIO COMMON DECK) FOLLOWS AT THIS IDENT + 2.
*
```

10-47

```
                                                LIST   -R                                                         CIOCOM       3
                                        *       SCOPE 3 TAPE INSTALLATION PARAMETER COMMON DECK                   CIOCOM       4
                                        *                                                                         CIOCOM       5
                                        **      INSERT MODIFICATIONS HERE.....................................   CIOCOM       6
     0000                               IP.LDEN   CEQU     6                                                      NWL         16
     0000                               IP.TDEN   CEQU     2                                                      NWL         17
     0000                               IP.TSG    CEQU     4002030                                                NWL         18
                                        *                                                                         CIOCOM       7
                                        *       CDC DEFAULT DEFINITIONS FOR TAPE PARAMETERS                       CIOCOM       8
                                        *                                                                         CIOCOM       9
     0000                               IP.AL84   CEQU   0           =1 IF ALL TAPE CHANNELS HAVE A 6684          CIOCOM      10
     0000                               IP.LDEN   CEQU   3           TAPE LABEL DENSITY (3=556, 4=200, 5=800)     CIOCOM      11
     0000                               IP.NBCD   CEQU  0             9-TRACK CONVERSION MODE DEFAULT(0=US,1=EP)  CIOCOM      12
     0000                               IP.NDEN   CEQU   2           9-TRACK DEFAULT DENSITY (2=800, 3=1600)      CIOCOM      13
     0000                               IP.NOISE  CEQU   3           MAXIMUM LENGTH IN BYTES OF NOISE RECORD      CIOCOM      14
     0000                               IP.NOIS9  CEQU   170         MAX LGTH (8-BIT BYTES) FOR PACKED MODE       CIOCOM      15
                                        *                                 9-TRACK TAPES                           CIOCOM      16
     0000                               IP.NTCN   CEQU   2           NUMBER OF TAPE CHANNELS                      CIOCOM      17
     0000                               IP.PTCN   CEQU   138         PRIMARY TAPE CHANNEL NUMBER                  CIOCOM      18
     0000                               IP.RCYC   CEQU   30000       DEFAULT RETENTION CYCLE FOR TAPE LABELS      CIOCOM      19
     0000                               IP.RPE1   CEQU   120         TOTAL NO. READ PARITY RETRIES                CIOCOM      20
     0000                               IP.RPE2   CEQU   80          NO. READ PARITY RETRIES ON-THE-FLY           CIOCOM      21
     0000                               IP.TDEN   CEQU   0           DEFAULT TAPE DATA DENSITY(0=556,1=200,2=800  CIOCOM      22
     0000                               IP.TSG    CEQU   4000338     TAPE STAGING OPTIONS                         CIOCOM      23
                                        *                                                                         CIOCOM      24
                                        ENDER     MICRO  1,,**                                                    CIOCOM      25
                                                                                                                  CIOCOM      26
                                                  MACRO  IPTSG,Q,X                                                CIOCOM      27
                                                  LOCAL  Y,Z                                                      CIOCOM      28
                                        Y         SET    IP.TSG                                                   CIOCOM      29
                                        Z         SET    IP.TSG                                                   CIOCOM      30
                                                  DUP    X+1,1                                                    CIOCOM      31
                                        Y         SET    Y/2                                                      CIOCOM      32
                                                  DUP    X,1                                                      CIOCOM      33
                                        Z         SET    Z/2                                                      CIOCOM      34
                                        Q         IFNE   Y*2,Z                                                    CIOCOM      35
                                                  ENDM                                                            CIOCOM      36
                                                                                                                  CIOCOM      37
                                        *         THE FOLLOWING LOCATIONS ARE SET BY CIO BEFORE ANY OVERLAY       CIOCOM      38
                                        *         IS EXECUTED.                                                    CIOCOM      39
                                                                                                                  CIOCOM      40
                                        CIOCOM    MACRO                                                           CIOCOM      41
                                        FSTEOP    EQU    D.FNT             EOP CODE(6),ALLOC OR TAPE STYLE(6)     CIOCOM      42
                                        FST2UN    EQU    FSTEOP+1          2ND UNIT ORDINAL OR 1ST RBT WD PAIR    CIOCOM      43
                                        FSTFRB    EQU    FST2UN            FIRST RB ADDRESS                       CIOCOM      44
                                        FSTORD    EQU    FSTEOP+2          PRIME UNIT ORD                         CIOCOM      45
                                        FSTCRB    EQU    FSTORD            CURRENT RB ADDRESS                     CIOCOM      46
                                        FSTRBT    EQU    FSTEOP+3          CURRENT RBT ORD                        CIOCOM      47
                                        FSTPRC    EQU    FSTEOP+4          CURRENT PR COUNT                       CIOCOM      48
                                        FSTFT1    EQU    FSTPRC+1          FET ADDRESS (6)                        CIOCOM      49
                                        FSTFT2    EQU    FSTFT1+1          FET ADDRESS (12)                       CIOCOM      50
                                        FSTDSP    EQU    FSTFT2+1          FST DISPOSITION CODE                   CIOCOM      51
                                        FSTSEC    EQU    FSTDSP+1          PERMISSION (4),WRITE(1), E/N (1)       CIOCOM      52
                                        FSTCST    EQU    FSTSEC+1          CODE/STATUS                            CIOCOM      53
                                        ESTASG    EQU    D.EST             ALLOC ASSIGNMENT                       CIOCOM      54
                                        ESTCH12   EQU    ESTASG+1          CHANNEL CHOICE 1,2                     CIOCOM      55
                                        ESTCH34   EQU    ESTCH12+1         CHANNEL CHOICE 3,4                     CIOCOM      56
```

10-48

```
                                                                              DCOM      57
ESTHDW     EQU    ESTCH34+1        HARDWARE MNEMONIC                          CIOCOM    58
ESTUNT     EQU    ESTHDW+1         DSI ORD                                    CIOCOM    59
FETFN1     EQU    D.BA             FET FILE NAME CHARS 1,2                    CIOCOM    60
FETFN2     EQU    FETFN1+1         FET FILE NAME CHARS 3,4                    CIOCOM    61
FETFN3     EQU    FETFN1+2         FET FILE NAME CHARS 5,6                    CIOCOM    62
FETFN4     EQU    FETFN1+3         FET FILE NAME CHAR 7,REC LVL,ERR FLG       CIOCOM    63
FETCST     EQU    FETFN1+4         ERR FLAG(3), CODE/STATUS(9)                CIOCOM    64
BS         EQU    D.BA+5           LAST CODE/STATUS FROM FNT(3)               CIOCOM    65
FA         EQU    D.FA             FNT(2) ADDRESS                             CIOCOM    66
LGTFET     EQU    D.PPONE+1        BIT 0-10 LENGTH FET                        CIOCOM    67
RANDM      EQU    D.PPONE+1        BIT 11 RANDOM BIT FROM FET                 CIOCOM    68
NEWFNT     EQU    D.TR             EXISTING FNT =0, NEW FNT NE 0              CIOCOM    69
CIOCOM     ENDM                                                               CIOCOM    70
                                                                              CIOCOM    71
*          THE FOLLOWING ARE USED FOR COMMUNICATIONS WITH THE LABEL           CIOCOM    72
*          PROCESSORS.  SYMBOLS WITH -F- PREFIXES ARE FUNCTION CODES          CIOCOM    73
*          FOR LABEL PROCESSORS.  SYMBOLS WITH -S- PREFIXES ARE STATUS        CIOCOM    74
*          CODES RETURNED BY LABEL PROCESSORS.                                CIOCOM    75
                                                                              CIOCOM    76
                                                                              CIOCOM    77
LBLCOM     MACRO                                                              CIOCOM    78
D.CW       EQU    D.FR6            CONTROL WORD FOR 4LB/4LC                   CIOCOM    79
                                                                              CIOCOM    80
FWEOV      EQU    0                WRITE *EOV*                                CIOCOM    81
FWEOF      EQU    1                WRITE *EOF*                                CIOCOM    82
FWVOL      EQU    2                WRITE *VOL*                                CIOCOM    83
FWHDR      EQU    3                WRITE *HDR*                                CIOCOM    84
FWEOS      EQU    6                WRITE LABEL/DATA SEPARATOR                 CIOCOM    85
FWTM       EQU    7                WRITE TAPE MARK                            CIOCOM    86
FRSKPF     EQU    10B              READ/SKIP FORWARD TAPE MARK                CIOCOM    87
FREOS      EQU    11B              SKIP TO END OF LABEL SET                   CIOCOM    88
*                 12B              NOT USED                                   CIOCOM    89
FCHKEXP    EQU    13B              READ AND CHECK EXPIRATION                  CIOCOM    90
FREADC     EQU    14B              READ AND CHECK LABEL RECORD                CIOCOM    91
FNPEC      EQU    1                IF READ PARITY ERROR, NO MESSAGE           CIOCOM    92
*                 15B              FREADC WITH NC PARITY ERROR CHECK          CIOCOM    93
FREADCO    EQU    16B              READ/CHECK/DELIVER                         CIOCOM    94
*                 17B              NOT USED                                   CIOCOM    95
FSTAT      EQU    20B              RETURN TAPE STATUS                         CIOCOM    96
FSTATD     EQU    21B              RETURN DYNAMIC TAPE STATUS                 CIOCOM    97
FREW       EQU    30B              REWIND TAPE                                CIOCOM    98
FUNL       EQU    31B              REWIND/UNLOAD TAPE                         CIOCOM    99
FBKSP      EQU    32B              BACKSPACE ONE PHYSICAL RECORD              CIOCOM   100
FSKPF      EQU    33B              HARDWARE SKIP FORWARD TAPE MARK            CIOCOM   101
FSKPB      EQU    34B              SKIP BACK WARD TAPE MARK                   CIOCOM   102
FSETDEN    EQU    37B              SET DENSITY FOR DATA                       CIOCOM   103
FFMT       EQU    50B              FORMAT SPLTFET                             CIOCOM   104
FMSG       EQU    4000B                                                       CIOCOM   105

SEOV       EQU    0                 VOLUME TRAILER READ                       CIOCOM   106
SEOF       EQU    1                 FILE TRAILER READ                         CIOCOM   107
SVOL       EQU    2                 VOLUME HEADER READ                        CIOCOM   108
SHDR       EQU    3                 FILE HEADER READ                          CIOCOM   109
SGARBG     EQU    4              UNRECOGNIZABLE RECORD STATUS                 CIOCOM   110
STM        EQU    7                 TAPE MARK READ STATUS                     CIOCOM   111
SPARER     EQU    20B               READ PARITY ERROR                         CIOCOM   112
SUNEXP     EQU    40B               UNEXP-RED TAPE STATUS                     CIOCOM   113
SEXP       EQU    44B               EXPIRED LABEL STATUS
```

10-49

```
SEOR      EQU   2000B                 END-OF-REEL STATUS BIT                      CIOCOM   114

TPFWA     EQU   C.PPFWA-1             FWA OF T.TAPES                              CIOCOM   115
TPORD     EQU   TPFWA-1               RELATIVE ADDRESS OF CURRENT ENTRY           CIOCOM   116
FETFMTED  EQU   TPORD-1               SPLTFET-ALREADY-FORMATTED FLAG              CIOCOM   117
TPLWA     EQU   FETFMTED-1            LWA + 1 OF T.TAPES                          CIOCOM   118
LBLFLGS   EQU   TPORD-5               FLAGS FROM T.TAPES                          CIOCOM   119
BLKCNT    EQU   LBLFLGS-6             BLOCK COUNT                                 CIOCOM   120
SPLTFET   EQU   BLKCNT-40             LABEL INFO FIELDS(FET OR T.TAPES)           CIOCOM   121
SPLTLBL   EQU   7777B-46              LABEL INFO FIELDS (ACTUAL LABEL)            CIOCOM   122
LBLBUF    EQU   SPLTLBL-11                                                        CIOCOM   123
LBLCOM    ENDM                                                                    CIOCOM   124
                                                                                  CIOCOM   125
*         THE FOLLOWING ARE USED FOR COMMUNICATIONS WITH THE ERROR                CIOCOM   126
*         MESSAGE OVERLAY, 6WM.                                                   CIOCOM   127
*         LIST OF ERROR MESSAGES THAT CAN BE WRITTEN BY 6WM.                      CIOCOM   128
*         (F) ERRORS ARE FATAL--UNCONDITIONAL ABORT                               CIOCOM   129
*         (EP) ERRORS ARE FATAL IF THE EP BIT IS ZERO, OTHERWISE, 6WM             CIOCOM   130
*         RETURNS CONTROL TO THE USER WITH AN APPROPRIATE ERROR CODE.             CIOCOM   131
*         (NF) ERRORS ARE INFORMATIVE DIAGNOSTICS.  THE CALLING  PP               CIOCOM   132
*         MUST BE PREPARED TO RECEIVE CONTROL BACK FROM 6WM.                      CIOCOM   133
*         ANY ERROR NUMBER OTHER THAN THOSE LISTED WILL CAUSE                     CIOCOM   134
*         *UNDEFINED ERROR* TO BE PRINTED--THE JOB WILL BE ABORTED.               CIOCOM   135
*                                                                                 CIOCOM   136
                                                                                  CIOCOM   137
ERRCOM    MACRO                                                                   CIOCOM   138
ERRNO     EQU   D.FNT+1                    ERROR NUMBER FOR 6WM                   CIOCOM   139
ERRNO2    EQU   D.FNT+2                                                           CIOCOM   140
ERRN1     EQU   1D               FET OUTSIDE FL                               (F) CIOCOM   141
ERRN2     EQU   2D               CIO CODE NOT DEFINED ON DEVICE              (EP) CIOCOM   142
ERRN3     EQU   3D               ILLEGAL FILE NAME                            (F) CIOCOM   143
ERRN4     EQU   4D               READ OR SKIP F AFTER WRITE                  (EP) CIOCOM   144
ERRN5     EQU   5D               SYSTEM ERROR TAPES-TABLE                     (F) CIOCOM   145
ERRN6     EQU   6D               WAITING FOR FNT SPACE                  (SPECIAL) CIOCOM   146
ERRN7     EQU   7D               PHYSICAL/LOGICAL POSITIONS DISAGREE         (NF) CIOCOM   147
ERRN8     EQU   8D               BUFFER ARGUMENT ERROR                        (F) CIOCOM   148
ERRN9     EQU   9D               ERROR CONDITION NOT CLEARED LAST REQUEST(F) CIOCOM   149
ERRN10    EQU   10D              AUTO-TAPE ASSIGNMENT UNSUCCESSFUL            (F) CIOCOM   150
ERRN11    EQU   11D              READ PERMISSION NOT GRANTED                 (EP) CIOCOM   151
ERRN12    EQU   12D              ILLEGAL FUNCTION CODE                        (F) CIOCOM   152
ERRN13    EQU   13D              DATA BLOCK TO LONG                           (F) CIOCOM   153
ERRN14    EQU   14D              ILL-FORMED MULTIFILE SET                     (F) CIOCOM   154
ERRN15    EQU   15D              MUST POSITION MEMBER FILE                    (F) CIOCOM   155
ERRN16    EQU   16D              BLANK TAPE READ                     (SPECIAL NF) CIOCOM   156
ERRN17    EQU   17D              (UNUSED)                                         CIOCOM   157
ERRN18    EQU   18D              FET TOO SHORT                                (F) CIOCOM   158
ERRN19    EQU   19D              TRIED TO WRITE NOISE RECORD                  (F) CIOCOM   159
ERRN20    EQU   20D              UBC TOO LARGE                                (F) CIOCOM   160
ERRN21    EQU   21D              MLRS TOO LARGE                               (F) CIOCOM   161
ERRN22    EQU   22D              DATA EXCEEDS MLRS                            (F) CIOCOM   162
ERRN23    EQU   23D              NO MULTIFILE ENTRY                           (F) CIOCOM   163
ERRN24    EQU   24D              RELEASE ILLEGAL ON PERMANENT FILE           (EP) CIOCOM   164
ERRN25    EQU   25D              WRITE NOT AT EOI ON PERMANENT FILE          (EP) CIOCOM   165
ERRN26    EQU   26D              (UNUSED)                                         CIOCOM   166
ERRN27    EQU   27D              INDEX BUFFER TOO SMALL                      (EP) CIOCOM   167
ERRN28    EQU   28D              INDEX ADDRESS NOT IN FL                      (F) CIOCOM   168
ERRN29    EQU   29D              REWRITE REQUIRES MODIFY PERMISSION          (EP) CIOCOM   169
```

```
                               ERRN30   EQU    300        WRITE REQUIRES EXTEND PERMISSION       (EP) CIOCOM    171
                               ERRN31   EQU    310        EVICT ILLEGAL ON PERMANENT FILE        (EP) CIOCOM    172
                               ERRN32   EQU    320        DEVICE FULL/FILE MAY NOT OVERFLOW      (EP) CIOCOM    173
                               ERRN33   EQU    330        FILE MAY NOT RESIDE ON DEVICE ASSIGNED  (F) CIOCOM    174
                               ERRN34   EQU    340        SPECIFIED DEVICE NON-EXISTENT           (F) CIOCOM    175
                               ERRN35   EQU    350        MMTC MEMORY PARITY ERROR                (F) CIOCOM    176
                               ERRCOM   ENDM                                                          CIOCOM    177
                                                                                                      CIOCOM    178
                                                                                                      CIOCOM    179
                               *        THE FOLLOWING ARE BIT SETTINGS FOR USE IN GENERATING STACK    CIOCOM    180
                               *        REQUESTS.                                                     CIOCOM    181
                                                                                                      CIOCOM    182
                                                                                                      CIOCOM    183
      0000                     STRFNTP  BIT    S.STFNTP+S.STF                                         CIOCOM    184
      0000                     STRREL   BIT    S.STFREL+S.STF                                         CIOCOM    185
      0000                     STRFETP  BIT    S.STFETP+S.STF                                         CIOCOM    186
      0000                     STREOF   BIT    S.STFEOF+S.STF                                         CIOCOM    187
      0000                     STRF     BIT    S.STF                                                  CIOCOM    188
      0000                     STRFA    BIT    S.STFA+S.STF                                           CIOCOM    189
      0000                     STRPRI   BIT    S.STFPRI+S.STF                                         CIOCOM    190
                    0000040    STRPRIN  EQU    STRPRI/STRFA                                           CIOCOM    191
                    0000007    STREXA   EQU    S.STF+1                                                CIOCOM    192
                                        LIST   R
```

10-51

```
                                                                                   1RT     158
                                                CIOCOM                             1RT     159
          0000                                  ERRCOM
          0000

                                                                                   1RT     161
                                   *            LOCAL     MACROS


                                                                                   1RT     163
                                   DIF          MACRO     X,Y                      1RT     164
                                                LDD       X                        1RT     165
                                                SBD       Y                        1RT     166
                                                SHN       12                       1RT     167
                                                ADD       X+1                      1RT     168
                                                SBD       Y+1                      1RT     169
                                                ENDM


                                                                                   1RT     171
                                   VRM          MACRO     M                        1RT     172
                                                VFD       12/M                     1RT     173
                                                ENDM                               1RT     174
                                   VRM                                             1RT     175
                                   CVRT         MACRO                              1RT     176
                                                SHN       12         RIGHT JUSTIFY HIGH ORDER CHARACTER   1RT     177
                                                STD       D.Z4       SAVE FOR INDEXING                    1RT     178
                                                SHN       -12        DO THE SAME FOR LOW ORDER            1RT     179
                                                STD       D.Z1                                            1RT     180
                                                LDM       BCDA,D.Z4  HIGH ORDER DISPLAY CODED             1RT     181
                                                SHN       6                                               1RT     182
                                                LMM       BCDA,D.Z1  AND LOW ORDER                        1RT     183
                                                STI       D.Z2       STORED BACK IN PLACE                 1RT     184
                                                AOD       D.Z2       READY FOR NEXT PAIR
                                                ENDM
```

10-53

```
1000                             ORG    C.PPFWA
1000      0100 3145              LJM  PRS


                           *           CONNECT/FUNCTION ROUTINE - MODE I
                           *     TRANSMISSION PARITY, REJECTS BECAUSE OF EQUIPMENT HARDWARE
                           *     RESERVED AND OF UNKNOWN CAUSE, AND EQUIPMENT NOT-READY WILL
                           *     RESULT IN A REJECTED CALL (A IS RETURNED NON-ZERO).  REJECTS
                           *     BECAUSE OF BUSY RESULT IN A LOOP ON THE FUNCTION.  THE
                           *     FUNCTION IS ARTIFICIALLY REJECTED IF NOT-READY STATUS IS
                           *     FOUND - THIS HAS NO ADVERSE EFFECT BECAUSE THIS ROUTINE DOES
                           *     NOT PERFORM AN UNLOAD AT ANY TIME.

1002      3006             CFGO  LDD    ST                  CHECK READY AND EXIT
1003      1201                   LPN    READY               WITH (A)=0 IF IT IS, ELSE GO
1004      0422                   ZJN  CFR2                  ISSUE NOT READY MESSAGE.
1005      1400                   LDN    0
1006      0100 1006        CFRX  LJM  *
                0001007    CFR   EQU    *-1
1010      7613             CFR1  FAN    IP.PTCN             OUTPUT FUNCTION OR
1011      0200 1161              RJM  STS                   CONNECT AND GO READ STATUS.  A
1013      3004                   LDD    CS                  REFLY WAS RETURNED IF THE
1014      0465                   ZJN  CFGO                  CONVERTER STATUS IS ZERO.
1015      1017                   SHN    15                  JUMP IF REJECTED ON
1016      0721                   MJN  CFR9                  TRANSMISSION PARITY.
1017      3006                   LDD    ST
1020      1006                   SHN    6                   JUMP IF RESERVE
1021      0710                   MJN  CFR6                  REJECT,
1022      1012                   SHN    10                  JUMP IF BUSY
1023      0722                   MJN  CFR11                 REJECT,
1024      1001                   SHN    1                   JUMP IF UNIT IS READY -
1025      0707                   MJN  CFR7                  THERE IS NO APPARENT REASON FOR
                           *                                   THE REJECT - WE WILL TRY AGAIN.
1026      2000 1220        CFR2  LDC    STSA                DISPLAY MESSAGE (*NOT
1030      0312                   UJN  CFR10                 READY*)

1031      2000 1226        CFR6  LDC    STSB                DISPLAY MESSAGE
1033      0310                   UJN  CFR10                 (*RESERVED*)

1034      2000 1233        CFR7  LDC    CFRA                DISPLAY MESSAGE
1036      0305                   UJN  CFR10                 (*REJECT*)

1037      0200 1066        CFR9  RJM  DLYMCL                DELAY AND MASTER CLEAR XMSN
1041      2000 1237              LDC    CFR8                PARITY, DISPLAY MESAGE (*XMSN
                           *                                   PARITY*)
1043      0200 1205        CFR10 RJM  MSG                   DISPLAY THE ERROR
1045      0200 1076        CFR11 RJM  RELEQ                 MESSAGE AND RELEASE EQUIPMENT
1047      1401                   LDN    1                   SET REJECT RETURN VALUE,
1050      0100 1006              LJM  CFRX                  (A) NON-ZERO AND EXIT
```

10-54

```
                                    *
                                    *           EXECUTE FUNCTION ON AN EQUIPMENT
                                    *
1052        0100 1052               FCNX      LJM   *
                      0001053       FCN       EQU     *-1                 SAVE FUNCTION CODE
                                              STD     FN                  AND GO TRANSMIT IT TO THE UNIT.
1054        3407                    FCN1      RJM   CFR                   EXIT IF ACCEPTED AND UNIT READY
1055        0200 1007                         ZJN   FCNX                  RESERVE AND RECONNECT
1057        0472                              RJM   RES                   THE UNIT, THEN RETRY THE
1060        0200 1131                         LDD     FN                  FUNCTION.  NOTE *** THIS INSTR.IS SET
1062        3007                    MMTCSW1   UJN   FCN1                  * TO A PSN WHEN THE PRESET SECTION
1063        0371                              UJN   FCNX                  * IS VERIFYING 657 TRANSPORT ASSIGN-
1064        0365                                                          * MENT, AFTER WHICH IT IS RESET TO AN
                                                                          * UJN FCN1.


                                    *
                                    *       DELAY FOR HALF A SECOND THEN MASTER CLEAR THE CHANNEL.
                                    *
1065        0100 1065               DLYMCLX   LJM   *
                      0001066       DLYMCL    EQU     *-1
                                              LDN     0
1067        1400                    DLYMCL1   SBN     1
1070        1701                              NJN   DLYMCL1
1071        0576                    DLYMCL2   FNC     MASCL,IP.PTCN        MASTER CLEAR
1072        7713 1700                         UJN   DLYMCLX                RETURN.
1074        0370


                                    *
                                    *       RELEASE EQUIPMENT WITHOUT CHECKING STATUS
                                    *
1075        0100 1075               RELEQX    LJM   *
                      0001076       RELEQ     EQU     *-1
                                    *
                                              LDN     0
1077        1400                    REL1      FAN     IP.PTCN              DESELECT 668X
1100        7613                              LDC     2100B
1101        2000 2100               REL2      FAN     IP.PTCN
1103        7613                              LDD     CH                   DROP CHANNEL
1104        3046                              RJM   R.DCH
1105        0200 0577                         RJM   PAUSE                  RETURN IF NO ERRORS
1107        0200 1153                         ZJN   RELEQX                 EXIT IF ERRORS
1111        0463                              LJM   RTD
1112        0100 2236
```

```
                                   *                                                                1RT        286
                                   *       RESERVE EQUIPMENT. CHANNEL WILL BE RESERVED AND          1RT        287
                                   *                                                                1RT        288
                                   *       THE EQUIPMENT WILL BE CONNECTED.                         1RT        289
                                   *                                                                1RT        290
1114        2000 2000              RES3    LDC    2000B                SELECT 668X                  1RT        291
                                   *                                                                1RT        292
1116        7613                   RES4    FAN    IP.PTCN                                           1RT        293
1117        2000 0000                      LDC    0                                                 1RT        294
                    0001120        CONECT  EQU    *-1                                               1RT        295
1121        0200 1007                      RJM  CFP                                                 1RT        296
1123        0507                           NJN  RES1                   LOOP IF NOT ACCEPTED         1RT        297
1124        6010                           CRD    D.T0                 CLEAR CONSOLE MESSAGE        1RT        298
1125        3076                           LDD    D.CPAD                                            1RT        299
1126        1635                           ADN    X.CPDFM+5                                         1RT        300
1127        6210                           CWD    D.T0                                              1RT        301
1130        0100 1130                      LJM  *                                                   1RT        302
                                                                                                    1RT        303
                    0001131        RES  -  EQU    *-1                                               1RT        304
1132        3034                   RES1    LDD    D.EST+2                                           1RT        305
1133        3412                           STD    D.T2                                              1RT        306
1134        3033                           LDD    D.EST+1                                           1RT        307
1135        1340                           SCN    40B                  CLEAR THE 6684 DESIGNATOR    1RT        308
1136        0200 0561                      RJM  R.RCH                  REQUEST CHANNEL              1RT        309
                                                                                                    SCAL33R      1
                                   IF1     IFNE   1,IP.NTCN                                         SCAL33R      2
1140        5000 1175                      LDM    SISB                 WORD CONTAINING LAST CHANNEL NO.  1RT   310
1142        1277                           LPN    77B                                               1RT        311
1143        3211                           SBD    D.T1                 IF SAME AS PRESENT ASSIGNMENT,  1RT     312
1144        0447                           ZJN  RES3                 SKIP TABLE.                    1RT        313
1145        2000 1271                      LDC    RESA                 FUNCTION ADDRESS TABLE       1RT        314
1147        0200 0663                      RJM  R.STB                  ASSIGN CHANNELS              1RT        315
                                                                                                    SCAL33R      3
                                   IF1     ENDIF                                                    SCAL33R      4
1151        0342                           UJN  RES3                                                1RT        316
                                                                                                    1RT        317



                                   *                                                                1RT        319
                                   *       PAUSE FOR STORAGE RELOCATION EXIT WITH ERROR FLAG = (A)  1RT        320
                                   *                                                                1RT        321
1152        0100 1152              PAUSEX  LJM  *                                                   1RT        322
                                                                                                    1RT        323
                    0001153        PAUSE   EQU    *-1                                               1RT        324
1154        0200 0446                      RJM  R.RAFL                                              1RT        325
1156        3011                           LDD    D.T0+C.CPEF                                       1RT        326
1157        0372                           UJN  PAUSEX                                              1RT        327
                                                                                                    1RT        328


                                   *                                                                1RT        330
                                   *           READ CONVERTER AND UNIT STATUS.  ON EXIT,            1RT        331
                                   *       CS=CONVERTER STATUS .AND. 7, (A)=ST=UNIT STATUS.         1RT        332
```

10-55

10-56

```
                                 *
1160      0100 1160        STSX     LJM  *
                0001161    STS      EQU    *-1
1162      7713 1200        STS1     FNC    STAT,IP.PTCN      READ THE CONVERTER STATUS AND
1164      7413             STS2     ACN    IP.PTCN           SAVE THE TRANSMISSION PARITY,
1165      7013             STS3     IAN    IP.PTCN           EXTERNAL AND INTERNAL REJECT
1166      7513             STS4     DCN    IP.PTCN           BITS.
1167      1207                      LPN    7
1170      3404                      STD    CS
1171      7713 1300        STS5     FNC    EXSTS,IP.PTCN     READ UNIT
1173      7413             STS6     ACN    IP.PTCN           STATUS AND
1174      7013             STS7     IAN    IP.PTCN           SAVE IN ST.
1175      7513             STS8     DCN    IP.PTCN
1176      3406                      STD    ST
1177      0360                      UJN  STSX                EXIT, ST IN A-REG


                           *
                           *        COPY MESSAGE TO CONSOLE
                           *
                           *        ENTER WITH (A) = ADDRESS OF MESSAGE
                           *
1200      3076             MSG2     LDD    D.CPAD            COPY TO CP
1201      1635                      ADN    35B
1202      6373 1251                 CWM    MSGA,D.TR
1204      0100 1204                 LJM    *                 RETURN EXIT
                0001205    MSG      EQU    *-1
                           *
1206      3401                      STD    D.Z1              STORE MESSAGE ADDRESS
1207      2000 1254                 LDC    MSGA+3
1211      3402                      STD    D.Z2
1212      4001             MSG1     LDI    D.Z1
1213      4402                      STI    D.Z2
1214      0463                      ZJN  MSG2                EXIT WHEN MESSAGE MOVED
1215      3601                      AOD    D.Z1
1216      3602                      AOD    D.Z2
1217      0372                      UJN  MSG1                LOOP


                           *
                           *        2TR MESSAGES
                           *
1220      1617             STSA     DIS    ,*NOT READY*
                           *
1226      2205             STSB     DIS    ,*RESERVED*
                           *
1233      2205             CFRA     DIS    ,*REJECT*
                           *
1237      3015             CFRB     DIS    ,*XMSN PARITY ERROR*
```

10-57

```
                                  *                                                                1RT      384
                                  *                                                                1RT      385
        1251      5315            MSGA      DIS   ,*$MT00*                                         1RT      386
        1255                                BSSZ  12                                               1RT      387


                                  IF1       IFNE  1,IP.NTCN        IF ONLY ONE TAPE CHANNEL, SKIP THIS.  1RT      389
                                  *         CHANNEL TABLE                                          1RT      390
                                  *                                                                1RT      391
        1271                      RESA      VRM   D.T1             CHANNEL ADDRESS                 1RT      392
        1272                                VRM   STS1                                             1RT      393
        1273                                VRM   STS2                                             1RT      394
        1274                                VRM   STS3                                             1RT      395
        1275                                VRM   STS4                                             1RT      396
        1276                                VRM   RES4                                             1RT      397
        1277                                VRM   REL1                                             1RT      398
        1300                                VRM   REL2                                             1RT      399
        1301                                VRM   CFR1                                             1RT      400
        1302                                VRM   STS5                                             1RT      401
        1303                                VRM   STS6                                             1RT      402
        1304                                VRM   STS7                                             1RT      403
        1305                                VRM   STS8                                             1RT      404
        1306                                VRM   DLYMCL2                                          1RT      405
        1307                                VRM   BLANK6                                           1RT      406
        1310                                VRM   RD1                                              1RT      407
        1311                                VRM   RD2                                              1RT      408
        1312                                VRM   RD3                                              1RT      409
        1313                                VRM   RD4                                              1RT      410
        1314                                VRM   RD5                                              1RT      411
        1315                                VRM   RD6                                              1RT      412
        1316                                VRM   RB1                                              1RT      413
        1317                                VRM   RB2                                              1RT      414
        1320                                VRM   RB3                                              1RT      415
        1321                                VRM   RB4                                              1RT      416
        1322                                VRM   RB5                                              1RT      417
        1323                                VRM   BYTE1                                            1RT      418
        1324                                VRM   BYTE2                                            1RT      419
        1325                                VRM   BYTE3                                            1RT      420
        1326                                VRM   EB1                                              1RT      421
        1327                                VRM   EB2                                              1RT      422
        1330                                VRM   EB3                                              1RT      423
        1331                                VRM   LD1                                              1RT      424
        1332                                VRM   LD2                                              1RT      425
        1333                                VRM   IAN184                                           1RT      426
        1334                                VRM   IAM184                                           1RT      427
        1335                                VRM   IAM284                                           1RT      428
        1336                                VRM   EB584                                            1RT      429
        1337                                VRM   B584                                             1RT      430
        1340                                VRM   RDN1                                             1RT      431
        1341                                VRM   RDN2                                             1RT      432
        1342                                VRM   BLANK2                                           1RT      433
        1343                                VRM   BLANK4                                           1RT      434
        1344                                VRM   BLANK5                                           1RT      435
        1345                                VRM   C4                                               1RT      436
        1346                                VRM   CEF3                                             1RT      437
```

```
 1347                            BSSZ  1
                         IF1     ENDIF


 1350     0000           SKPSIZ  DATA   0


                         *       THIS ROUTINE SELECTS MODE, SELECTS INPUT TO END OF RECORD, AND
                         *       BEGINS TAPE MOTION. IT IS ENTERED FROM THE PREREAD ROUTINE TO
                         *       BEGIN A SERIES OF READS, OR FROM READ TO CONTINUE TAPE MOTION
                         *       WHILE TRANSMITTING THE PREVIOUS RECORD TO CM.
 1351     0100 1351      RDSX    LJM  *

               0001352   RDSIN   EQU    *-1
 1353     1401           RDS     LDN    1                 (BINARY)      SELECT THE MODE, AND
 1354     0200 1053              RJM  FCN                 CHECK TAPE READY. CHANGE TO BCD ON BC
 1356     7713 1400      RD1     FNC    INREC,IP.PTCN     SELECT INPUT TO EOR AND START
 1360     7413           RD2     ACN    IP.PTCN           TAPE MOTION.
 1361     0367                   UJN  RDSX                EXIT



 1362                    PREREAD DIF    D.OUT,D.IN        THE AVAILABLE BUFFER SPACE IS GIVEN
 1367     1701                   SBN    1                 BY OUT-IN-1 OR BY LIMIT-FIRST+OUT-IN-
 1370     0603                   PJN    *+3               WHICH EVER IS POSITIVE.
 1371     2100 0000              ADC    0                 (LIMIT-FIRST)
               0001372   SPACECK EQU    *-1
 1373     2177 6777              ADC    -BINCM
               0001374   CMPRU   EQU    *-1               ONE PRU, START THE TAPE AND READ.
 1375     0612                   PJN  INIT                ELSE TERMINATE AND RETURN.

                         *
                         *       THE FOLLOWING INSTRUCTION IS A NOP IF A READSKP IS BEING
                         *       EXECUTED. .THIS ALLOWS THE CIRCULAR BUFFER TO BE
                         *       COMPLETELY FILLED FOR THIS OPERATION.
                         *


 1376     0315           RDSKP   UJN  TORET

 1377     2100 1000              ADC    BINCM             NO. OF CM WORDS UNFILLED
               0001400   RDSKP1  EOU    *-1
 1401     0412                   ZJN  TORET               JUMP IF NONE
 1402     3410                   STD    D.T0              X5 - COMPUTE BYTES NEEDED
 1403     1002                   SHN    2
```

```
1404    3110                        ADD    D.T0              
1405    5400 1350                   STM    SKPSIZ            SAVE CM BYTE COUNT.
1407    1414              INIT      LDN    IP.RPE1           INITIALIZE ERROR RETRY COUNT
1410    3447                        STD    RC
1411    3037                        LDD    STATUS            EXIT IF STATUS SAYS SO
1412    0403                        ZJN    *+3
1413    0100 2136         TORET     LJM    RETURN

1415    0200 1352         PRERD1    RJM    RDSIN             TO BEGIN TAPE MOTION
1417    2000 3145         CVSH      LDC    BCDBUF
1421    3435                        STD    LA
1422    5400 2776                   STM    B484              INITIALIZE 6684 CONV-ON-FLY LOOP.
1424    1603                        ADN    3
1425    5400 2631                   STM    R84+1             INITIALIZE 6681 CONV-ON-FLY LOOP.

1427    1400              BLANK1    LDN    0                 BEGIN BLANK TAPE CHECK.
1430    6613 1461         BLANK2    FJM    RDI,IP.PTCN      JUMP IF DATA--TO RDI ON BIN, TO
               0001431    BLANK3    EQU    *-1               * RDCV ON BCD/81, TO RDCV84 ON BCD/84
1432    1701                        SBN    1
1433    0574                        NJN    BLANK2            JUMP AND CONTINUE SCANNING THE TAPE.

                          *
                          *         THE TAPE IS BLANK SO STOP MOTION AND ISSUE MESSAGE.
                          *
1434    6513 1437         BLANK6    IJM    BLANK5,IP.PTCN    JUMP IF CHANNEL DROPPED ACTIVE
1436    7513              BLANK4    DCN    IP.PTCN
1437    7713 1700         BLANK5    FNC    MASCL,IP.PTCN     STOP TAPE MOTION.
1441    1420                        LDN    ERRN16            *BLANK TAPE READ*

1442    3421              CALL6WM   STD    ERRNO
1443    0200 1076                   RJM    RELEQ             RELEASE EQUIPMENT AND CHANNEL.
1445    2000 4127                   LDC    2R6W
1447    3416                        STD    D.T6
1450    2000 1500                   LDC    1RM*100B
1452    3417                        STD    D.T7
1453    2000 5773                   LDC    C.PP6WA-L.PPHDR

               0412715    CV.6WM    EQU    OV.6WM

1455    0200 0135                   RJM    R.OVL             LOAD AND TRANSFER
1457    0200 6001                   RJM    C.PP6WA+1         * CONTROL TO 6WM.




                          *
                          *                   DIRECT READS STORE THE ENTIRE PHYSICAL RECORD INTO
                          *         THE PP MEMORY AND THEN TRANMITS THE DATA TO CENTRAL MEMORY
                          *
1461    2000 5000         RDI       LDC    BINPRU            THE BYTE COUNT IS PRESET FOR BINARY--
               0001462    PRUSIZ    EQU    *-1               * I.E., FOR A 512 CM WORD PRU.

1463    7113 2540         RD3       IAM    BINBUF,IP.PTCN    READ A PRU OF BYTES
1465    3405                        STD    WC                SAVE RESIDUAL BYTE COUNT.
1466    0403                        ZJN    RD4-1             JUMP IF MAX.NO.OF BYTES WERE READ.
1467    0100 1534                   LJM    WTCM              JUMP IF LESS THAN MAX READ.

1471    1404                        LDN    DCECNT            CHECK FOR EXCESSIVE RECORD LENGTH.
```

10-59

```
                                                                                            1RT   536
 1472    7113 0010     RD4       IAM    D.T0,IP.PTCN                                        1RT   537
 1474    0536          RD44      NJN   WTCMA           JUMP IF POSSIBLY NOT EXCESSIVE.      1RT   538
                                                                                            1RT   539
 1475    5400 1474     RD44A     STM    RD44           RECORD LENGTH IS EXCESSIVE SO BLOCK  1RT   539
 1477    0471                    ZJN   RD4-1           * NORMAL PROCESSING AND              1RT   540
 1500    3030          DCEXIT    LDD    D.FNT+8        *  SET                               1RT   541
 1501    1301                    SCN    1              *   DEVICE                           1RT   542
 1502    1601                    ADN    1              *    CAPACITY                        1RT   543
 1503    3430                    STD    D.FNT+8        *     EXCEEDED                       1RT   544
 1504    2000 0321               LDC    INT2-RDSKIP+300B *                                  1RT   545
 1506    5400 2151               STM    RDSKIP         *                                    1RT   546
 1510    2400          RPHRSW3   PSN                   =PSN IF NOT A RPHR REQUEST.          1RT   547
                                                                                            1RT   548
 1511    0200 1161     DCEP      RJM   STS                                                  1RT   549
 1513    1010                    SHN    8              WAIT FOR END-OPERATION.              1RT   550
 1514    0674                    PJN   DCEP                                                 1RT   551
 1515    2220 0001               LPC    200001B                                             1RT   552
 1517    0506                    NJN   DCEP1           JUMP IF BAD PARITY (POSSIBLE SKEW).  1RT   553
 1520    3624          DCEP2     AOD    D.FNT+4                                             1RT   554
 1521    1063                    SHN    -12            INCREMENT PRU COUNT.                 1RT   555
 1522    3523                    RAD    D.FNT+3                                             1RT   556
 1523    0100 2136               LJM   RETURN          RETURN TO USER.                      1RT   557
                                                                                            1RT   558
 1525    2000 0536     DCEP1     LOC    500B+WTCMA-RD44  RESTORE D.C.E. LOOP.               1RT   559
 1527    5400 1474               STM    RD44                                                1RT   560
 1531    0303                    UJN   WTCM            ATTEMPT SKEW RECOVERY.               1RT   561
                                                                                            1RT   562
                       *                                                                    1RT   563
                       *               THE BUFFER NOW CONTAINS A PHYSICAL RECORD OF DATA,   1RT   564
                       *         CONVERTED IF REQUIRED.  (WC)=PRU SIZE - RECORD LENGTH.     1RT   565
                       *                                                                    1RT   566
                                                                                            1RT   567
 1532    1704          WTCMA     SBN    DCECNT         GO TO DEV CAP EXCEEDED ROUTINE IF    1RT   568
 1533    0544                    NJN   DCEXIT          * BYTE COUNT IS EXCESSIVE.           1RT   569
 1534    5000 1462     WTCM      LDM    PRUSIZ         COMPUTE RECORD LENGTH IN CM          1RT   570
 1536    3205                    SBD    WC             WORDS BY DIVIDING THE RECORD SIZE BY 1RT   571
 1537    3435                    STD    LA                                                  1RT   572
 1540    3472                    STD    RLTH           *1     FIVE WHICH IS ACCOMPLISHE BY T 1RT  573
 1541    5000 1350               LDM    SKPSIZ                                              1RT   574
 1543    0503                    NJN   *+3                                                  1RT   575
 1544    3072                    LDD    RLTH                                                1RT   576
 1545    0305                    UJN   RPHRSW1                                              1RT   577
 1546    3272                    SBD    RLTH                                                1RT   578
 1547    0603                    PJN   RPHRSW1         JUMP IF SHORT PRU WILL FIT IN C.B.   1RT   579
 1550    3172                    ADD    RLTH           SINCE THE PRU IS G.T. THE SPACE IN   1RT   580
 1551    3472                    STD    RLTH           * C.B., SET BYTE COUNT TO FIT IN C.B. 1RT  581
 1552    0304          RPHRSW1   UJN   NORMAL1         THIS IS A PSN FOR RPHR  REQUESTS.    1RT   582
                                                                                            1RT   583
 1553    1604                    ADN    4              FOR RPHR INCREMENT BYTE COUNT BY 4   1RT   584
 1554    3435                    STD    LA             * TO ALLOW FOR TRUNCATION BY THE     1RT   585
 1555    3472                    STD    RLTH           * DIVIDE BY 5 ALGORITHM.             1RT   586
                                                                                            1RT   587
 1556    3072          NORMAL1   LDD    RLTH                                                1RT   588
 1557    1001                    SHN    1              *2     FOLLOWING APPROX. SUITABLY    1RT   589
 1560    3172                    ADD    RLTH           *3     REARRANGED.  X/5 IS APPROXIMAT 1RT  590
 1561    1002                    SHN    2              *12    EQUAL TO (2**14+1)*X/2**14*5, 1RT   591
 1562    3172                    ADD    RLTH           *13    IN FACT THE RESULT IS EXACT FO 1RT  592
```

10-60

```
1563    1014                  SHN   12            /2**6 2**12.  THE FORMULA DECOMPOSES
1564    3411                  STD   D.T1          (51*X+13*X/2**6)/2**8 AND IS CREDITED
1565    1010                  SHN   8             *52    TO W. SILVERMAN.
1566    3272                  SBD   RLTH          *51
1567    3111                  ADD   D.T1          +13/2**6
1570    1067                  SHN   -8            /2**8
1571    3472                  STD   RLTH
                   *
                   *                 THE READ CHECK ROUTINE CONTROLS THE ERROR RECOVERY
                   *          PROCEDURE ON READS.  IF THE DEFINED PROCEDURE IS EXTENDED
                   *          BEYOND THE PRESENT CODE, USE OF AN OVERLAY IS RECOMMENDED.
                   *
1572    0200 1161  RDCK       RJM   STS           COPY STATUS FROM UNIT
1574    1010                  SHN   8             DELAY UNTIL THE END-OF-OPERATION
1575    0674                  PJN   RDCK          COMES UP.
1576    1006                  SHN     6           EXIT TO RETURN IF FILE MARK
1577    0721                  MJN   RDCKR         STATUS IS UP.
1600    1220                  LPN     20B
1601    0512                  NJN   LD1           JUMP ON LOST DATA.
1602    5000 1462             LDM     PRUSIZ      CHECK FOR A NOISE RECORD
1604    3205                  SBD     WC
1605    1704                  SBN     4           READ A NEW RECORD IF A VERY
1606    0714                  MJN   RDCKL         SHORT (NOISE) RECORD WAS READ
1607    3006                  LDD     ST          TEST IF A PARITY ERROR
1610    2200 2400             LPC     PARITY      AND TRY TO RECOVER IF TRUE.
1612    0412                  ZJN   RDOK          ELSE CONTINUE TO PROCESS DATA
1613    6513 1616  LD1        IJM   *+3,IP.PTCN  DE-ACTIVATE CHANNEL IF NECESSARY
1615    7513       LD2        DCN     IP.PTCN
1616    0100 2333             LJM   RCVR          GO TO RECOVERY PROCEDURE

1620    0100 2124  RDCKR      LJM   TURNEOF       RETURN TO USER.

1622    0100 1415  RDCKL      LJM   PRERD1        REREAD RECORD

1624    3004       RDOK       LDD     CS
1625    1204                  LPN     4
1626    0564                  NJN   LD1           JUMP IF XMSN PARITY ERROR ON DATA.
1627    3030                  LDD     D.FNT+8
1630    0312       RDOKC      UJN   RDOKA         THIS INSTRUCTION IS SET TO A PSN IF
                                                  * A GO WAS GIVEN TO AN UNRECOVERED
                                                  * READ PARITY ERROR (EP BIT OFF).
1631    1201                  LPN     1
1632    0403                  ZJN   *+3           JUMP IF REC.LENGTH NOT EXCESSIVE.
1633    0100 1520             LJM   DCEP2         JUMP IF DEVICE CAPACITY EXCEEDED.

1635    2000 0312             LDC     300B+RDOKA-RDOKC
1637    5400 1630             STM     RDOKC
1641    0303                  UJN   RDOKB

1642    1301       RDOKA      SCN     1           CLEAR SKEW INDICATOR.
1643    3430                  STD     D.FNT+8
1644    1414       RDOKB      LDN     IP.RPE1     RESET ERROR RETRY COUNT
1645    3447                  STD     RC
1646    3624                  AOD     D.FNT+4
1647    1063                  SHN     -12         INCREMENT THE PRU COUNT
1650    3523                  RAD     D.FNT+3
                   *            FOLLOWING IS   *PSN*   IF LP ON AND UNLABELLED.
```

10-61

```
                                          *                   HE IS ALLOWED TO KNOW HE IS PAST REFLECT SPOT.
1651      0316                   EXPAS1     UJN  RDOK1
                     0001651     UPPAS1A    EQU    *-1

1652      3006                              LDD    ST
1653      1240                              LPN    ENDTAPE                IF END-OF-TAPE STATUS IS UP,
1654      0413                              ZJN  RDOK1                    RETURN END OF REEL TO STATUS
1655      3037                              LDD    STATUS
1656      2277 5777                         LPC    775777B                TURN OFF AND
1660      2300 2000                         LMC    2000B                    TURN ON E-O-REEL.
1662      3437                              STD    STATUS
                                 *                  IF THE UP FLAG IS SET IN THE FET, THE NEXT
                                 *          INSTRUCTION IS CHANGED TO INCREMENT A NON-SENSE CELL.
1663      5600 2265                         AOM    RTDX                   INCREMENT RETURN ADDRESS
                     0001664     UPPAS1     EQU    *-1                    AND CONTINUE
1665      0100 1725                         LJM  WTDATA

1667      3005                   RDOK1      LDD    WC                     IF FULL PRU CONTINUE, ELSE
1670      0413                              ZJN  READ                     SET LEVEL NUMBER INTO ITS CELL.  THE
1671      5000 1764                         LDM    BUFLOC                 LEVEL NUMBER IS FOUND IN THE LAST
1673      3535                              RAD    LA                     CELL READ.
1674      5035 7776                         LDM    -1,LA
1676      1277                              LPN    77B
1677      3436                              STD    LEV
                                 *                  ON READ WITH CONVERTS, THE FOLLOWING IS A PASS
1700      0325                   BCDPAS1    UJN  WTDATA

1701      0100 2556                         LJM  FILL                     GO CHECK FOR FILL ON SHORT PRU

1703                             READ       DIF    D.OUT,D.IN             THE AVAILABLE BUFFER SPACE IS GIVEN
1710      1701                              SBN    1                      BY OUT-IN-1 OR BY LIMIT-FIRST+OUT-IN-
1711      0603                              PJN  *+3                      WHICH EVER IS POSITIVE.
1712      2100 0000                         ADC    0                      (LIMIT-FIRST)
                     0001713     BUFLTH     EQU    *-1
1714      2177 5777                         ADC    -BINCM*2
                     0001715     CMPRUSZ    EQU    *-1                    TAPE MOTION FOR NEXT RCD. MAY BE
1716      0603                   RPHRFIX    PJN  READOK                   STARTED EARLY, AT THIS TIME)--JUMP IF
                                                                          MOTION HAS STARTED.***NOTE* THIS
                                                                          INSTR. IS MODIFIED TO AN UJN---WTDATA
                                                                          FOR A RPHR  REQUEST.

1717      3645                              AOD    OVLPST                 SET FLAG TO INDICATE MOTION NOT START
1720      0505                              NJN  WTDATA                   GO TO WRITE DATA TO CM.
1721      3037                   READOK     LDD    STATUS                 CHECK STATUS. IF LAST RCD. NOT OK, GO
1722      0503                              NJN  WTDATA                   WRITE DATA TO CM WITHOUT TAPE MOTION.
1723      0200 1352                         RJM  RDSIN                    TO BEGIN TAPE MOTION
1725                             WTDATA     DIF    D.LIMIT,D.IN           COMPUTE THE CM TRANSMIT VALUES
1732      3411                              STD    D.T1                   WHICH REQUIRE TWO PARTS IF THE BUFFER
1733      3272                              SBD    RLTH                   WRAPS AROUND.  D.T1 CONTAINS THE
1734      0706                              MJN  WT1                      LENGTH OF THE FIRST SEGMENT AND D.T2
1735      0402                              ZJN  *+2
1736      1401                              LDN    1
1737      3413                              STD    D.T3
1740      3072                              LDD    RLTH                   CONTAINS THE LENGTH OF THE SECOND
1741      3411                              STD    D.T1                   SEGMENT.  D.T3 IS ZERO IF IN + RLTH
1742      3072                   WT1        LDD    RLTH                   IS EXACTLY LIMIT
1743      3211                              SBD    D.T1
```

10-62

```
 1744      3412                            STD     D.T2
 1745      3062                            LDD     D.IN          SAVE OLD IN POINTER IN CASE DORKIN IS
 1746      3415                            STD     D.T5          A NOP.
 1747      3063                            LDD     D.IN+1
 1750      3416                            STD     D.T6
 1751      3072                            LDD     RLTH          CHECK FOR ZERO DATA
 1752      0503                            NJN    *+3            JUMP IF DATA EXISTS
 1753      0100 2023                       LJM    WT4            GO ON WITHOUT WRITING TO CM

 1755                                      LOCA    D.IN          WRITE THE FIRST SEGMENT OF
 1763      6311 2540                       CWM     BINBUF,D.T1   LENGTH D.T1 INTO THE CM BUFFER
                   0001764    BUFLOC       EQU     *-1
 1765      3012                            LDD     D.T2
 1766      0510                            NJN    WT2            JUMP SECOND SEGMENT NOT EMPTY
 1767      3013                            LDD     D.T3
 1770      0426                            ZJN    WT3            JUMP IF IN+RLTH=LIMIT
 1771      3072                            LDD     RLTH          UPDATE IN POINTER BY ADDING THE
 1772      3563                            RAD     D.IN+1        RECORD LENGTH, THEN GO UPDATE IT IN
 1773      1063                            SHN     -12           CENTRAL MEMORY
 1774      3562                            RAD     D.IN
 1775      0326                            UJN    WT4

 1776      3011               WT2          LDD     D.T1          COMPUTE SECOND DATA SEGMENT
 1777      1002                            SHN     2             ORIGIN AS BUFLOC+5*SEG1 LENGTH AND
 2000      3111                            ADD     D.T1          WRITE THE SECOND SEGMENT TO CENTRAL
 2001      5100 1764                       ADM     BUFLOC        MEMORY.
 2003      5400 2014                       STM     SEG2
 2005                                      LOCA    D.FIRST
 2013      6312 0000                       CWM     0,D.T2
                   0002014    SEG2         EQU     *-1
 2015      3012                            LDD     D.T2          UPDATE THE IN POINTER BY ADDING
 2016      3161               WT3          ADD     D.FIRST+1     THE SECOND SEGMENT LENGTH TO FIRST
 2017      3463                            STD     D.IN+1
 2020      1063                            SHN     -12
 2021      3160                            ADD     D.FIRST
 2022      3462                            STD     D.IN
 2023      1400               WT4          LDN     P.ZERO        CLEAR WORK AREA
 2024      6010                            CRD     D.T0


                              *
                              *      THE FOLLOWING INSTRUCTION IS A NOP WHEN AN UNCORRECTABLE READ
                              *      ERROR IS BEING PROCESSED BECAUSE THE EP BIT IS SET.
                              *


 2025      0321               DORKIN       UJN    WT0

 2026      3023                            LDD     D.FNT+3
 2027      3412                            STD     D.T2
 2030      3024                            LDD     D.FNT+4
 2031      3413                            STD     D.T3          SEND PRU COUNT TO USERS CIR.BUFFER.
 2032                                      LOCA    D.IN          GET NEW IN LOCATION
```

```
 2040       6212                           CWD    D.T2        WRITE OLD IN POINTER THERE            1RT        758
                     0002040     RDSK      EQU    *-1                                                SC30019      1
 2041       1400                           LDN    0           CLEAR T2 SO THAT IN CAN BE UPDATED    1RT        759
 2042       3412                           STD    D.T2        * WITH THE H.C. BITS ALL ZERO.        1RT        760
 2043       2000 4000                      LDC    4000B                                              1RT        761
 2045       3544                           RAD    D.BA+4      SET PARITY ERROR STATUS.              1RT        762
 2046       3062                 WT0       LDD    D.IN                                               1RT        763
 2047       3413                           STD    D.T3                                               1RT        764
 2050       3063                           LDD    D.IN+1      WRITE NEW IN VALUE INTO THE           1RT        765
 2051       3414                           STD    D.T4        USER FET(3)                           1RT        766
 2052                                      LDCA   D.PPIRB+3                                          1RT        767
 2060       6240                           CWD    D.BA        SET PARITY ERROR STATUS IN USERS FET. 1RT        768
 2061       1602                           ADN    2                                                  1RT        769
 2062       6210                           CWD    D.T0                                               1RT        770
 2063       1601                           ADN    1           READ CURRENT OUT FROM THE USER        1RT        771
 2064       6010                           CRD    D.T0        FET(4).  UPDATE THE POINTER           1RT        772
 2065       3013                           LDD    D.T3                                               1RT        773
 2066       1237                           LPN    37B                                                1RT        774
 2067       3464                           STD    D.OUT                                              1RT        775
 2070       1014                           SHN    12                                                 1RT        776
 2071       3114                           ADD    D.T4                                               1RT        777
 2072       3465                           STD    D.OUT+1                                            1RT        778
 2073       1063                           SHN    -12         CHECK TO SEE IF OUT HAS BEEN UPDATED  1RT        779
 2074       3266                           SBD    D.LIMIT     * PAST CIRCULAR BUFFER LIMIT.         1RT        780
 2075       1014                           SHN    12                                                 1RT        781
 2076       3165                           ADD    D.OUT+1                                            1RT        782
 2077       3267                           SBD    D.LIMIT+1                                          1RT        783
 2100       0707                           MJN    PNTROK      JUMP IF OUT POINTER IS O.K.           1RT        784
 2101       6513 2104            RDN1      IJM    DCE,IP.PTCN                                        1RT        785
 2103       7513                 RDN2      DCN    IP.PTCN                                            1RT        786
 2104       1410                 DCE       LDN    ERRNB       *BUFFER ARG ERROR*                    1RT        787
 2105       0100 1442                      LJM    CALL6WM                                            1RT        788
                                                                                                     1RT        789
 2107       3037                 PNTROK    LDD    STATUS      EXIT IF STATUS SHOWS ANY              1RT        790
 2110       0505                           NJN    PHYS+1      UNUSUAL CONDITIONS                    1RT        791
 2111       3045                           LDD    OVLPST      CHECK IF NEXT TAPE MOTION HAS STARTED 1RT        792
 2112       0506                           NJN    WT6         IF NOT, GO TO SEE IF IT NOW CAN BE.   1RT        793
 2113       3005                           LDD    WC                                                 1RT        794
                                 *             THE FOLLOWING INSTRUCTION HAS CLEARED TO A PASS TO    1RT        795
                                 *         CAUSE AN UNCONDITIONAL EXIT ON A RPHR REQUEST.  EXIT IS   1RT        796
 2114       0402                 PHYS      ZJN    WT5         ALSO TAKEN ON A SHORT PRU, IE ON END  1RT        797
 2115       0321                           UJN    RETURN      OF LOGICAL RECORD.  OTHERWISE WE TRY  1RT        798
                                                                                                     1RT        799
 2116       0100 1417            WT5       LJM    CVSW        TO READ THE NEXT PRU.                 1RT        800
                                                                                                     1RT        801
 2120       1400                 WT6       LDN    0           CLEAR FLAG                            1RT        802
 2121       3445                           STD    OVLPST                                             1RT        803
 2122       0100 1362                      LJM    PREREAD     TO CHECK FOR POSSIBLE CONTINUATION    1RT        804
                                                                                                     1RT        805
```

10-64

```
                                  *                    * TAPE MARK HAS READ...
2124        3006                  TURNEOF  LDD    ST                  IF
2125        1240                           LPN    ENDTAPE              EOT MARKER
2126        0410                           ZJN  RETURN                   (NO, GO)
2127        3037                           LDD    STATUS
2130        2277 5777                      LPC    775777B              TURN OFF
2132        2300 2000                      LMC    2000B                  AND TURN ON E-O-REEL.
2134        5600 2265                      AOM    RTDX                CAUSE A CALL TO 1RP.

                                  *
                                  *        RELEASE TAPE AND RETURN PROPER STATUS TO LOW CORE FNT BUFFER
                                  *
2136        0200 1076             RETURN   RJM  RELEQ                 RELEASE TAPE AND DROP CHANNEL
2140        3006                  INT      LDD    ST
2141        1210                           LPN    TAPMARK
2142        0407                           ZJN  RDSKIP                TAPE MARK/PHYS EOR NOT HIT
2143        2000 1000                      LDC    1000B               RETURN EOI
2145        3537                           RAD    STATUS
2146        5600 2265                      AOM    RTDX                INCREMENT RETURN ADDRESS
2150        0324                           UJN  INT2A                 GO TO


                                  *        THE FOLLOW ING INST IS ZEROED TO A PASS IF A READ SKIP
                                  *        IS BEING PROCESSED
2151        0321                  RDSKIP   UJN  INT2

2152        3005                           LDD    WC                  JUMP IF FULL PRU WAS READ
2153        0407                           ZJN  INT5
2154        3030                           LDD    D.FNT+8             COMPARE LEVEL NUMBER READ
2155        1274                           LPN    74B                 TO THAT REQUESTED
2156        1075                           SHN    -2
2157        3236                           SBD    LEV                 EXIT IF REQUEST IS LESS THAN
2160        0412                           ZJN  INT2                  OR EQUAL TO  THE ONE READ
2161        0711                           MJN  INT2                    SET UP TO CALL 2TF.
2162        5600 2265             INT5     AOM    RTDX
2164        3031                           LDD    D.FNT+9
2165        1202                           LPN    2
2166        2100 0240                      ADC    240B                SET FNT TO A SKIPF REQUEST.
2170        0100 2300                      LJM  RECYTF

2172        3005                  INT2     LDD    WC                  JUMP IF SHORT RECORD READ
2173        0504                           NJN  INT3
2174                              INT2A    BSS    0
2174        3031                           LDD    D.FNT+9
2175        3537                           RAD    STATUS
2176        0316                           UJN  INT4

2177        1420                  INT3     LDN    20B                 RETURN END OF RECORD
2200        3537                           RAD    STATUS
2201        3036                           LDD    LEV                 AND LEVEL NUMBER
2202        1002                           SHN    2
2203        3330                           LMD    D.FNT+8
2204        1274                           LPN    74B
2205        3330                           LMD    D.FNT+8
2206        3430                           STD    D.FNT+8
2207        1274                           LPN    74B                 IF LEVEL 17, RETURN
2210        1774                           SBN    74B
```

10-65

```
                                                                                    1RT   865
 2211     0503                        NJN  INT4                                     1RT   866
 2212     1410                        LDN    10B        END OF FILE                 1RT   867
 2213     3537                        RAD    STATUS                                 1RT   868
 2214     3037                INT4    LDD    STATUS     FETCH STATUS BITS THAT HAVE TO BE SET 1RT 869
 2215     2300 7777                   LMC    7777B                                  1RT   870
 2217     5400 2230                   STM    STAT1      * AND PUT THEM IN LPC INSTRUCTION. 1RT 871
 2221     1220                        LPN    20B                                    1RT   872
 2222     0504                        NJN  STAT2                                    1RT   873
 2223     3031                        LDD    D.FNT+9                                1RT   874
 2224     1310                        SCN    10B                                    1RT   875
 2225     3431                        STD    D.FNT+9                                1RT   876
 2226     3031                STAT2   LDD    D.FNT+9                                1RT   877
 2227     2200 0000                   LPC    **         CLEAR OUT BITS TO BE SET.   1RT   878
                    0002230   STAT1   EQU    *-1                                    1RT   879
 2231     3137                        ADD    STATUS     TURN ON REQUIRED BITS.      1RT   880
 2232     3431                        STD    D.FNT+9                                1RT   881
 2233     5000 2265                   LDM    RTDX                                   1RT   882
 2235     0532                        NJN  RECYC                                    1RT   883
 2236     3030                RTD     LDD    D.FNT+8                                1RT   884
 2237     3343                        LMD    D.BA+3     EOR STATUS BITS             1RT   885
 2240     1277                        LPN    77B                                    1RT   886
 2241     3343                        LMD    D.BA+3     REPLACES CLEARED BITS       1RT   887
 2242     3443                        STD    D.BA+3                                 1RT   888
 2243     3631                        AOD    D.FNT+9    MAKE FNT COMPLETE           1RT   889
 2244     3444                        STD    D.BA+4     MAKE FET COMPLETE           1RT   890
 2245     3057                        LDD    D.FA                                   1RT   891
 2246     6220                        CWD    D.FNT      UPDATE FNT(2)               1RT   892
 2247     1601                        ADN    1                                      1RT   893
 2250     6225                        CWD    D.FNT+5    UPDATE FNT(3)               1RT   894
 2251                                 LDCA   D.PPIRB+3  UPDATE FET(1)               1RT   895
 2257     6240                        CWD    D.BA                                   1RT   896
 2260     1412                        LDN    M.DPP                                  1RT   897
 2261     0200 0516                   PJM  R.PROCES     DROP PP                     1RT   898
 2263     0100 0103                   LJM  R.IDLE       EXIT TO IDLE                1RT   899
                                                                                    1RT   900
 2265                         RTDX    BSSZ   2





                                                                                    1RT   902
                              *       IF A TAPE MARK HAS BEEN READ ON A SCOPE-STANDARD   1RT   903
                              *       TAPE, 1RP MUST BE EXECUTED REGARDLESS OF LABEL     1RT   904
                              *       DECLARATIONS AS ALL SUCH TAPES ARE TERMINATED      1RT   905
                              *       WITH TRAILER LABELS.  EOI STATUS IS ON.            1RT   906
                                                                                    1RT   907
                    0342220   OV.1RP  EQU    3R1RP                                  1RT   908
                    0002267   RECYC   EQU    *                                      1RT   909
 2267     2034 2220                   LDC    OV.1RP                                 1RT   910
 2271     1014                        SHN    12                                     1RT   911
 2272     3416                        STD    D.T6                                   1RT   912
 2273     1071                        SHN    -6                                     1RT   913
 2274     1377                        SCN    77B                                    1RT   914
 2275     3417                        STD    D.T7
```

```
 2276      0100 0125              LJM  R.OVLJ                LOAD AND EXECUTE 1RP                1RT       915
                                                                                                 1RT       916
                                                                                                 1RT       917
                        *         CALL CIO  TO RECALL DRIVER OR CALL ANOTHER DRIVER              1RT       918
                        *         CALL CIO TO LOAD 1TF                                           1RT       919
                        *

                                                                                                 1RT       921
 2300      3431         RECYTF    STD     D.FNT+9        SET INPUT REGISTER TO CIO               1RT       922
 2301      2000 0311              LDC     OV.CIO/64                                              1RT       923
 2303      3450                   STD     D.PPIRB                                                1RT       924
 2304      3051                   LDD     D.PPIRB+1                                              1RT       925
 2305      1277                   LPN     77B                                                    1RT       926
 2306      2100 1700              ADC     1RO*64                                                 1RT       927
 2310      3451                   STD     D.PPIRB+1                                              1RT       928
 2311      3074                   LDD     D.PPIR                                                 1RT       929
 2312      6250                   CWD     D.PPIRB                                                1RT       930
 2313      3631                   AOD     D.FNT+9        COMPLETE THE FNT                        1RT       931
 2314      1301                   SCN     1                                                      1RT       932
 2315      3444                   STD     D.BA+4                                                 1RT       933
 2316      3057                   LDD     D.FA                                                   1RT       934
 2317      6220                   CWD     D.FNT                                                  1RT       935
 2320      1601                   ADN     1                                                      1RT       936
 2321      6225                   CWD     D.FNT+5                                                1RT       937
 2322                             LDCA    D.PPIRB+3                                              1RT       938
 2330      6240                   CWD     D.BA                                                   1RT       939
 2331      0100 0103              LJM   R.IDLE           EXIT TO IDLE                            1RT       940


                                                                                                 1RT       943
                        *         RECOVERY CONSISTS OF A SERIES OF BACKSPACES FOLLOWED BY A      1RT       944
                        *         NEW ATTEMPT TO READ. THE LAST (IP.RPE2) ATTEMPTS BACK UP 3     1RT       945
                        *         RECORDS IF POSSIBLE AND MAKE A RUNNING START AT RECOVERY.      1RT       946
                        *         FAILURE OF THESE ATTEMPTS WILL RESULT IN SOLICITATION OF AN    1RT       947
                        *         OPERATOR DECISION OR DIRECT RETURN IF THE EP BIT IS SET.       1RT       948
                        *         IF THE EP BIT IS SET, LOCATION IN WILL CONTAIN THE             1RT       949
                        *         PRE-UPDATED ADDRESS OF IN.                                     1RT       950
                        *                                                                        1RT       951
                                                                                                 1RT       952
 2333      3004         RCVR      LDD     CS                                                     1RT       953
 2334      5400 2536              STM     CEF4           CONV STATUS TO ERROR FILE               1RT       954
 2336      1204                   LPN     4                                                      1RT       955
 2337      0407                   ZJN   RCVRE            JUMP IF NO TRANSMISSION PARITY ERROR.   1RT       956
 2340      0200 1066              RJM   DLYMCL           DELAY AND MASTER CLEAR.                 1RT       957
 2342      0200 1076              RJM   RELEQ            RELEASE AND                             1RT       958
 2344      0200 1131              RJM   RES              * RE-RESERVE THE EQUIPMENT.             1RT       959
 2346      3006         RCVRE     LDD     ST                                                     1RT       960
 2347      5400 2537              STM     CEF5           EQUIP STATUS TO ERROR FILE              1RT       961
 2351      1401                   LDN     1                                                      1RT       962
 2352      3410                   STD     D.T0
```

10-67

```
2353     3075                    LDD   D.PPMES1
2354     6310 2533               CWM   CEF1,D.T0          MOVE WORD TO MESSAGE BUFFER
2356     3412                    STD   D.T2               STORE LWA+1 OF ENTRY
2357     1420                    LDN   20B
2360     3411                    STD   D.Y1               STORE CE ERROR FILE FLAG.
2361     1401                    LDN   M.DFM
2362     0200 0516               RJM   R.MTR              ENTER ERROR IN CE ERROR FILE.

2364     3747                    SOD   RC                 CHECK RECOVERY COUNTER
2365     0503                    NJN   *+3
2366     0100 2446               LJM   EPPAS1

2370     1710                    SBN   IP.RPE2            CONTINUE IF ITS TIME FOR DOUBLE
                                                          * BACKSPACE AND READ ON THE FLY,ELSE
2371     0617                    PJN   RCVRA              JUMP IF SINGLE BACKSPACE REQUIRED
2372     3023                    LDD   D.FNT+3            CHECK PRU COUNT FRO LESS THAN THREE
2373     1277                    LPN   77B
2374     1014                    SHN   12
2375     3124                    ADD   D.FNT+4
2376     1703                    SBN   3
2377     0711                    MJN   RCVRA              JUMP IF 3 BKSP. CANNOT BE DONE
2400     1412                    LDN   BKSPS
2401     0200 1053               RJM   FCN                BACKSPACE 1 RECORD
2403     5400 2413               STM   RCVRB
2405     1412                    LDN   BKSPS
2406     0200 1053               RJM   FCN                BACKSPACE ANOTHER RECORD
2410     1412          RCVRA     LDN   BKSPS
2411     0200 1053               RJM   FCN                BACKSPACE A RECORD


                       *
                       *         THE FOLLOWING INSTRUCTION IS ZEROED IF A RUNNING START
                       *         IS BEING ATTEMPTED.
                       *
2413     0331          RCVRB     UJN   RCVRC
2414     2000 0331               LDC   0300B+RCVRC-RCVRB
2416     5400 2413               STM   RCVRB              RSET RUNNING START SWITCH

2420     0200 1352               RJM   RDSIN              BYPASS RECORD 1
2422     1401                    LDN   1
2423     7113 0010     RD5       IAM   D.T0,IP.PTCN
2425     0474                    ZJN   RD5-1
2426     0200 1161               RJM   STS                WAIT FOR END OF OPERATION
2430     1010                    SHN   8
2431     0674                    PJN   *-3
2432     0200 1352               RJM   RDSIN              BYPASS RECORD 2
2434     1401                    LDN   1
2435     7113 0010     RD6       IAM   D.T0,IP.PTCN
2437     0474                    ZJN   RD6-1
2440     0200 1161               RJM   STS                WAIT FOR END OF OPERATION
2442     1010                    SHN   8
2443     0674                    PJN   *-3
2444     0100 1415     RCVRC     LJM   PRERD1             TO REREAD RECORD IN ERROR
```

10-68

10-69

```
                                 *                                                                     1RT     1017
                                 *        IF THE EP FLAC IS ON, THE NEXT INSTRUCTION HAS               1RT     1018
                                 *        BEEN RESET TO A PASS.                                        1RT     1019
                                 *                                                                     1RT     1020
      2446      0311             EPPAS1   UJN  RCVR3                                                   1RT     1021
                                                                                                       1RT     1022
      2447      2000 4000                 LDC    4000B             SET ERROR FLAG IN STATUS            1RT     1023
      2451      3537                      RAD    STATUS            AND PROCESS BAD DATA AS IS          1RT     1024
      2452      1400                      LDN    0                 SET FLAG TO DORK IN                 1RT     1025
      2453      5400 2025                 STM    DORKIN                                                1RT     1026
      2455      0100 1644        RCVR2    LJM  RDOKB                                                   1RT     1027
                                                                                                       1RT     1028
      2457      0200 1161        RCVR3    RJM  STS                 CHECK FOR UNIT BUSY.                1RT     1029
      2461      1202                      LPN    BUSY                                                  1RT     1030
      2462      0574                      NJN  RCVR3               JUMP IF STILL BUSY.                 1RT     1031
      2463      0200 1076                 RJM  RELEQ               RELEASE EQUIP.UNTIL OPERATOR ACTION. 1RT    1032
                                                                                                       1RT     1033
      2465      2000 1241                 LDC    CFRB+2            SEND PARITY ERROR MESSAGE TO THE    1RT     1034
      2467      0200 1205                 RJM  MSG                 CONTROL POINT AND TO THE            1RT     1035
      2471      2001 1251                 LDC    MSGA+10000B       USERS DAYFILE.                      1RT     1036
      2473      0200 0671                 RJM  R.DFM                                                   1RT     1037
      2475      2000 0000        GOBIT1   LDC    **                                                    1RT     1038
      2477      6010                      CRD    0.T0                                                  1RT     1039
      2500      3010                      LDD    0.T0              SET *GO* BIT                        1RT     1040
      2501      1007                      SHN    7                                                     1RT     1041
      2502      1301                      SCN    1                                                     1RT     1042
      2503      1601                      ADN    1                                                     1RT     1043
      2504      1013                      SHN    13B                                                   1RT     1044
      2505      3410                      STD    0.T0                                                  1RT     1045
      2506      2000 0000        GOBIT    LDC    **                                                    1RT     1046
      2510      6210                      CWD    0.T0                                                  1RT     1047
      2511      0200 1153        PAR10    RJM  PAUSE                                                   1RT     1048
      2513      0403                      ZJN  ESTNTRY             IF DROP NOT GIVEN, LOOP             1RT     1049
      2514      0100 2236                 LJM  RTD                  ELSE COMPLETE FET+FNT, EXIT TO IDLE 1RT    1050
                                                                                                       1RT     1051
      2516      2000 0000        ESTNTRY  LDC    **                ADDRESS OF WORD CONTAINING GO BIT   1RT     1052
      2520      6010                      CRD    0.T0                                                  1RT     1053
      2521      3010                      LDD    0.T0                                                  1RT     1054
      2522      1006                      SHN    6                 *GO* NOW IN BIT 17                  1RT     1055
9     2523      0765                      MJN  PAR10               LOOP UNTIL BIT SET BY OPERATOR 'GO' 1RT     1056
      2524      1400                      LDN    0                                                     1RT     1057
      2525      5400 1630        RCVR6    STM    RDOKC             BYPASS DCE SUPPRESSION ON THIS PATH. 1RT    1058
      2527      0200 1131                 RJM  RES                 RE-RESERVE THE EQUIP. ON GO RESPONSE 1RT    1059
      2531      0100 1624                 LJM  RDOK                * AND TRANSMIT BAD DATA TO C.M.     1RT     1060
                                                                                                       1RT     1061




      2533      0110             CEF1     DATA   0110B             SYS I/O DRIVER /READ PARITY ERROR   1RT     1063
      2534      1000             CEF2     DATA   1000B             PROGRAM CODE = 1RT/EST ORDINAL      1RT     1064
      2535      0013             CEF3     VFD    12/IP.PTCN        PP NUMBER/CHANNEL NUMBER            1RT     1065
      2536      0000             CEF4     DATA   0                 LAST CHANNEL (6681-4) STATUS        1RT     1066
```

```
     2537       0000             CEFS      DATA    0                       LAST EQUIPMENT STATUS


                                 *
                                 *         TAPE BUFFER STARTS HERE EXCEPT WHEN CONVERSION TO DISPLAY
                                 *         CODE IS CALLED FOR
                                 *
                                           IFGT    *,7777B-BINPRU,1     NEXT CARD IS *ERR* (1RT TOO BIG)
                       0002540   BINBUF    EQU     *
                                 *
                                 *         THIS ROUTINE CONVERTS THE LEVEL NUMBER (WHICH IS ALREADY
                                 *         CONVERTED TO DISPLAY CODE) BACK TO AN OCTAL NUMBER. THE
                                 *         CONVERSION IS FROM 0 TO 16, OR 33-12, 31-4, 35-2, 36-3, 37-4,
                                 *         40-5, 41-6, 42-7, 43-10, 44-11, 54-13, 55-0, 61-17, 64-14,
                                 *         74-15.  LEVEL NUMBERS 1 THROUGH 11, 16, AND 12 ARE OBTAINED BY
                                 *         ARITHMETIC, THE REST BY A TABLE LOOK UP.
                                 *
     2540       0012              TABL     DATA    12B
     2541       0013                       DATA    13B
     2542       0000                       DATA    0
     2543       0000                       DATA    0
     2544       0000                       DATA    0
     2545       0015                       DATA    15B
     2546       1103             FILLB     LMN     3                     FLIP 14 TO 17 OR 17 TO 14
                                 *
                                 *         FOR OTHER THAN 14 OR 17 THIS PROCESSING DOES NOT MATTER.
                                 *
     2547       5400 2543                  STM     TABL+3                SET UP TABLE ENTRY (ONLY 14 OR 17)
     2551       1075                       SHN     -2                    REDUCE MAGNITUDE OF INCREMENT
                                 *
                                 *         IN THE ABOVE INSTRUCTION, 7 GOES TO 1, 10 TO 2, 14 TO 3,
                                 *         17 TO 3, AND 27 TO 5.
                                 *
     2552       3410             FILLC     STD     D.T0
     2553       5010 2540                  LDM     TABL,D.T0             TABLE LOOK UP
     2555       0307                       UJN    FILLA.

     2556       1745             FILL      SBN     45B                   CHECK FOR LEVEL 1 TO 11, 16, 12
     2557       0666                       PJN    FILLB                  JUMP IF NOT
     2560       1612                       ADN     12B
     2561       0470                       ZJN    FILLC                  IF LEVEL 12
     2562       0602                       PJN    *+2                    IF NOT LEVEL 16
     2563       1416                       LDN     16B                   SET LEVEL 16
     2564       3436             FILLA     STD     LEV
     2565       0100 1725                  LJM    WTDATA                 RETURN TO MAIN LINE LOOP.

                                 *
                                 *         THE CONVERSION ROUTINE STARTS HERE.  CONVERSION IS DONE IN
                                 *         TWO PHASES.  THE FIRST AND THIRD BYTES ARE CONVERTED AS DATA
                                 *         IS READ.  THE OTHER THREE BYTES ARE STORED IN LINE AND ARE
                                 *         CONVERTED AT THE END OF RECORD.  THE TAPE IS ALREADY IN
                                 *         MOTION WHEN THE ROUTINE IS ENTERED.
                                 *
     2567       6713 2645        BYTE1     EJM    EB1,IP.PTCN          IF CHANNEL BUFFER IS EMPTY,DELAY
     2571       7013             RB1       IAN     IP.PTCN               AS LONG AS IT IS ACTIVE. THEN
```

10-20

```
2572   1014                 SHN   12                  READ BYTE 1, SPLIT THE CHARACTERS,
2573   3411                 STD   D.T1                CONVERT THEM BY TABLE LOOKUP, AND
2574   1063                 SHN   -12                 LEAVE THE RESULT IN THE BUFFER.
2575   3412                 STD   D.T2
2576   5011 3045            LDM   CVTBL,D.T1
2600   1006                 SHN   6
2601   5312 3045            LMM   CVTBL,D.T2
2603   4435                 STI   LA
2604   3635                 AOD   LA
2605   6713 2647    BYTE2   EJM   EB2,IP.PTCN         READ BYTE 2 AND STORE IT IN
2607   7013         RB2     IAN   IP.PTCN               THE BUFFER. TO PICK UP TIME, THIS
2610   4435                 STI   LA                    BYTE IS NOT CONVERTED.
2611   3635                 AOD   LA
2612   6713 2651    BYTE3   EJM   EB3,IP.PTCN         READ BYTE 3, CONVERT IT AND
2614   7013         RB3     IAN   IP.PTCN               LEAVE IT IN THE BUFFER
2615   1014                 SHN   12
2616   3411                 STD   D.T1
2617   1063                 SHN   -12
2620   3412                 STD   D.T2
2621   5011 3045            LDM   CVTBL,D.T1
2623   1006                 SHN   6
2624   5312 3045            LMM   CVTBL,D.T2
2626   4435                 STI   LA
2627   1402                 LDN   2
2630   7113 0006    RB4     IAM   ,IP.PTCN              READ BYTES 4 + 5 DIRECTLY
2632   0522                 NJN   EB4                   INTO THE BUFFER, UNCONVERTED.
2633   1403                 LDN   3
2634   3535                 RAD   LA
2635   1603                 ADN   3
2636   5400 2631            STM   RB4+1
2640   2177 3432            ADC   -BCDBUF-BCDPRU        CHECK FOR A FULL PRU.
          0002641   XBCD1   EQU   *-1
2642   0615                 PJN   RBTA                  JUMP IF FULL PRU HAS BEEN READ.
2643   0100 2567            LJM   BYTE1

2645   6413 2567    EB1     AJM   BYTE1,IP.PTCN       CHANNEL EMPTY FIRST BYTE
2647   6413 2605    EB2     AJM   BYTE2,IP.PTCN       CHANNEL EMPTY SECOND BYTE
2651   6413 2612    EB3     AJM   BYTE3,IP.PTCN       CHANNEL EMPTY THIRD BYTE
2653   0313                 UJN   EB5                   CONTINUE

2654   1103         EB4     LMN   3                     ADJUST LA TO POINT TO THE NEXT
2655   3535                 RAD   LA                    AVAILABLE BYTE IN THE BUFFER
2656   0310                 UJN   EB5

2657   1401         RBTA    LDN   1                     FLUSH THE REST OF THE RECORD
2660   7113 0010    RB5     IAM   D.T0,IP.PTCN
2662   0504                 NJN   EB5                   JUMP IF PRU LENGTH IS NOT EXCESSIVE.

2663   3405                 STD   WC
2664   0100 1475            LJM   RD44A                 GO PROCESS DEVICE CAPACITY EXCEEDED.

2666   2000 3145    EB5     LDC   BCDBUF
2670   3405                 STD   WC
2671   5005 0001    CVT2    LDM   1,WC                  CONVERT BYTES 2, 4, AND 5
2673   1014                 SHN   12
2674   3411                 STD   D.T1
2675   1063                 SHN   -12
```

```
26         3412                         STD    D.T2
26         5011 3045                    LDM    CVTBL,D.T1
2701       1006                         SHN    6
2702       5312 3045                    LMM    CVTBL,D.T2
2704       5405 0001                    STM    1,WC
2706       5005 0003                    LDM    3,WC
2710       1014                         SHN    12
2711       3411                         STD    D.T1
2712       1063                         SHN    -12
2713       3412                         STD    D.T2
2714       5011 3045                    LDM    CVTBL,D.T1
2716       1006                         SHN    6
2717       5312 3045                    LMM    CVTBL,D.T2
2721       5405 0003                    STM    3,WC
2723       5005 0004                    LDM    4,WC              ON BYTE 5, CHECK FOR EXTERNAL
2725       2300 1672                    LMC    1672B             1632B CODE WHICH CONVERTS TO 0000B.
2727       0414               CVT3      ZJN   CVT4
2730       2300 1672                    LMC    1672B
2732       1014                         SHN    12
2733       3411                         STD    D.T1
2734       1063                         SHN    -12
2735       3412                         STD    D.T2
2736       5011 3045                    LDM    CVTBL,D.T1
2740       1006                         SHN    6
2741       5312 3045                    LMM    CVTBL,D.T2
2743       5405 0004          CVT4      STM    4,WC
2745       1405                         LDN    5                 ADD TO START OF NEXT CM WORD
2746       3505                         RAD    WC
2747       3235               CVT6      SBD    LA
2750       0603                         PJN   CVT5               DONE IF WC .GE. LA
2751       0100 2671                    LJM   CVT2

2753       0406               CVT5      ZJN   CVT7               FILL LAST FULL WORD
2754       1400                         LDN    0                 WITH ZEROS.
2755       5405 7776                    STM    -1,WC
2757       3705                         SOD    WC
2760       0366                         UJN   CVT6

2761       5000 1462          CVT7      LDM    PRUSIZ            COMPUTE THE WORD COUNT
2763       2100 3145                    ADC    BCDBUF
2765       3235                         SBD    LA
2766       3405                         STD    WC
2767       0603                         PJN   *+3                CHECK FOR DEVICE CAPACITY EXCEEDED---
2770       0100 1500                    LJM   DCEXIT             * AND EXIT IF DETECTED.
2772       0100 1534                    LJM   WTCM               GO EXIT

                              *
                              *         THIS ROUTINE PERFORMS LINE TERMINATOR CONVERSION (ON THE FLY)
                              *         FOR BCD SCOPE TAPES THAT ARE ACCESSED THROUGH A 6684.
                              *

2774       1404               B184      LDN    4
2775       7113 3145          IAM184    IAM    BCDBUF,IP.PTCN    READ 4 BYTES FROM TAPE
                0002776       B484      EQU    *-1
2777       0532                         NJN   END84              JUMP IF END OF PRU WAS DETECTED.
3000       1404                         LDN    4
3001       3535                         RAD    LA                UPDATE LWA+1.
```

10-72

10-73

```
3002     1405                 LDN   5
3003     5500 2776            RAM   B484               SET IAN INSTR.FOR NEXT READ.

3005     6713 3042    B584    EJM   EB584,IP.PTCN
3007     7013         IAN184  IAN    IP.PTCN           READ BYTE POSITION 5
3010     4435                 STI    LA
3011     1162                 LMN    62B               CHECK FOR A LINE TERMINATOR.
3012     0502                 NJN   *+2                JUMP IF NOT A LINE TERMINATOR.
3013     4435                 STI    LA                STORE ZERO FOR A LINE TERMINATOR.
3014     3635                 AOD    LA                INCREMENT LWA+1.
3015     2177 3432            ADC    -BCDBUF-BCDPRU    CHECK FOR A FULL PRU.
3017     0554                 NJN   B184               JUMP IF NOT A FULL PRU YET.
3020     1401         RBTA84  LDN    1
3021     7113 0010    IAM284  IAM    D.T0,IP.PTCN      FLUSH EXCESS OF PRU
3023     0504                 NJN   RB55               JUMP IF PRU LENGTH IS NOT EXCESSIVE.

3024     3405                 STD    WC
3025     0100 1475            LJM   RD44A              GO PROCESS DEVICE CAPACITY EXCEEDED.

3027     0100 2761    RA55    LJM   CVT7               MERGE WITH BCD/6681 ROUTINE.
3031     1701         END84   SBN    1
3032     1103                 LMN    3                 ADJUST LA TO POINT TO THE NEXT AVAIL-
3033     3535                 RAD    LA                * ABLE BYTE IN THE PP BUFFER.
3034     2000 3145    END84X  LDC    BCDBUF
3036     3135                 ADD    LA
3037     3405                 STD    WC
3040     0100 2747            LJM   CVT6               MERGE WITH BCD/6681 ROUTINE.
3042     6413 3005    EB584   AJM   B584,IP.PTCN
3044     0367                 UJN   END84X

                      *
                      *       CONVERSION TABLE - INTERNAL BCD TO DISPLAY CODE.
                      *
3045     0033         CVTBL   DATA   1R0
3046     0034                 DATA   1R1
3047     0035                 DATA   1R2
3050     0036                 DATA   1R3
3051     0037                 DATA   1R4
3052     0040                 DATA   1R5
3053     0041                 DATA   1R6
3054     0042                 DATA   1R7
3055     0043                 DATA   1R8
3056     0044                 DATA   1R9
3057     0063                 DATA   1R:
3060     0054                 DATA   1R=
3061     0064                 DATA   1R'
3062     0074                 DATA   1R[
3063     0000                 DATA   0
3064     0061                 DATA   1R(
3065     0045                 DATA   1R+
3066     0001                 DATA   1RA
3067     0002                 DATA   1RB
3070     0003                 DATA   1RC
3071     0004                 DATA   1RD
3072     0005                 DATA   1RE
3073     0006                 DATA   1RF
3074     0007                 DATA   1RG
```

10-74

```
      3075      0010                         DATA   1RH
      3076      0011                         DATA   1RI
      3077      0072                         DATA   1R<
      3100      0057                         DATA   1R.
      3101      0052                         DATA   1R)
      3102      0075                         DATA   1R?
      3103      0076                         DATA   1R"
      3104      0077                         DATA   77B
      3105      0046                         DATA   1R-
      3106      0012                         DATA   1RJ
      3107      0013                         DATA   1RK
      3110      0014                         DATA   1RL
      3111      0015                         DATA   1RM
      3112      0016                         DATA   1RN
      3113      0017                         DATA   1RO
      3114      0020                         DATA   1RP
      3115      0021                         DATA   1RQ
      3116      0022                         DATA   1RR
      3117      0066                         DATA   1R!
      3120      0053                         DATA   1R$
      3121      0047                         DATA   1R*
      3122      0070                         DATA   1R#
      3123      0071                         DATA   1R\
      3124      0073                         DATA   1R>
      3125      0055                         DATA   1R
      3126      0050                         DATA   1R/
      3127      0023                         DATA   1RS
      3130      0024                         DATA   1RT
      3131      0025                         DATA   1RU
      3132      0026                         DATA   1RV
      3133      0027                         DATA   1RW
      3134      0030                         DATA   1RX
      3135      0031                         DATA   1RY
      3136      0032                         DATA   1RZ
      3137      0062                         DATA   1R]
      3140      0056                         DATA   1R,
      3141      0051                         DATA   1R(
      3142      0065                         DATA   65B
      3143      0060                         DATA   1R_
      3144      0067                         DATA   1R^
                                    *
                                    *        THE INPUT BUFFER STARTS HERE WHEN CONVERSION TO DISPLAY CODE
                                    *        IS REQUIRED.
                                    *
                         0003145    BCDBUF   EQU    *



                                    *        PRESET OVERLAY
                                    *
      3145      3036                PRS      LDD    D.EST+4                FORM CONECT CODE
      3146      5400 1120                    STM    CONECT                 STORE CONECT FUNCTION IN-LINE
      3150      1421                         LDN    21B
      3151      3432                         STD    D.EST                  INTERNAL COMMUNICATION
      3152      3022                         LDD    D.FNT+2                SET EQUIPMENT NUMBER IN MESSAGE
      3153      1074                         SHN    -3                     FIRST DIGIT
```

10-75

```
3154        5500 1252              RAM     MSGA+1                                         1RT      1347
3156        3022                   LDD     D.FNT+2                                        1RT      1348
3157        5500 2534              RAM     CEF2               EST ORDINAL TO CE ERROR FILE 1RT      1349
3161        1207                   LPN     7                  SECOND DIGIT                1RT      1350
3162        1006                   SHN     6                  TO LEFT CHARACTER           1RT      1351
3163        5500 1253              RAM     MSGA+2                                         1RT      1352
3165        1400                   LDN     0                  CLEAR RE-READ COUNT         1RT      1353
3166        3447                   STD     RC                                             1RT      1354
3167        3437                   STD     STATUS             RETURN STATUS               1RT      1355
3170        3445                   STD     OVLPST                                         1RT      1356
3171        3436                   STD     LEV                                            1RT      1357
3172        1403                   LDN     3                                              1RT      1358
3173        3473                   STD     D.TR               SET CONSTANT 3              1RT      1359
3174        3074                   LDD     D.PPIR                                         1RT      1360
3175        1750                   SBN     50B                                            1RT      1361
3176        1003                   SHN     3                                              1RT      1362
3177        5500 2535              RAM     CEF3               PP NUMBER TO ERROR FILE     1RT      1363
3201        1413                   LDN     IP.PTCN                                        1RT      1364
3202        3446                   STD     CH                 INITIALIZE CHANNEL NUMBER   1RT      1365
                          *                                                               1RT      1366
                          *        CHECK PARAMETERS IN FET AND FNT AND ALTER INSTRUCTIONS 1RT      1367
                          *        ACCORDINGLY                                            1RT      1368
                          *                                                               1RT      1369
3203                               LDCA    D.PPIRB+3                                      1RT      1370
3211        1601                   ADN     1                  READ FET(2)                 1RT      1371
3212        6010                   CRD     D.T0                                           1RT      1372
3213        3011                   LDD     D.T1               IF EP BIT                   1RT      1373
3214        2200 0400              LPC     4008                                           1RT      1374
3216        0404                   ZJN   PRS5                 IS ON, ZERO                 1RT      1375
3217        1400                   LDN     0                                              1RT      1376
3220        5400 2446              STM     EPPAS1             SWITCH TO PASS INSTRUCTION  1RT      1377
3222        3011          PRS5     LDD     D.T1                                           1RT      1378
3223        2200 1000              LPC     1000B              IF UP BIT IS NOT ON         1RT      1379
3225        0411                   ZJN   PRS7                 JUMP.                       1RT      1380
3226        5600 1664              AOM     UPPAS1                                         1RT      1381
3230        3020                   LDD     D.FNT              IF                          1RT      1382
3231        1214                   LPN     14B                  LABELLED,                 1RT      1383
3232        0504                   NJN   PRS7                   GO. ELESE                 1RT      1384
3233        1400                   LDN     0                    LET HIM SEE REFLECTIVE SPOT. 1RT   1385
3234        5400 1651              STM     UPPAS1A                                        1RT      1386
3236        0200 1131     PRS7     RJM   RES                                              1RT.     1387
3240        1414                   LDN     P.TAPES            TAPES TABLE SEARCH FOR GO BIT 1RT    1388
3241        6001                   CRD     D.Z1                                           1RT      1389
3242        1400                   LDN     0                                              1RT      1390
3243        3400                   STD     D.Z0                                           1RT      1391
3244        3001          TTAP1    LDD     D.Z1               FWA/8 OF T.TAPES            1RT      1392
3245        1003                   SHN     3                                              1RT      1393
3246        3100                   ADD     D.Z0               ADDRESS OF AN ENTRY         1RT      1394
3247        6003                   CRD     D.Z3                                           1RT      1395
3250        5400 2517              STM     ESTNTRY+1          STORE FOR LATER USE         1RT      1396
3252        1063                   SHN     -12                                            1RT      1397
3253        5400 2516              STM     ESTNTRY                                        1RT      1398
3255        3003                   LDD     D.Z3               EST ORDINAL                 1RT      1399
3256        3322                   LMD     FSTORD             ASSIGNED EST ORDINAL        1RT      1400
3257        0410                   ZJN   TTAP2                JUMP IF MATCH               1RT      1401
                                                                                          1RT      1402
3260        1410                   LDN     LE.TAPES           THEN ADVANCE SEARCH         1RT      1403
```

```
 3261     3500                        RAD   D.Z0          INCREASE INCREMENT                 1RT     1404
 3262     3202                        SBD   D.Z2          LENGTH OF T.TAPES                  1RT     1405
 3263     0760                        MJN  TTAP1          TRY AGAIN IF STILL WITHIN LIMIT    1RT     1406
 3264     1405                        LDN   ERRN5         * SYSTEM TAPES-TABLE ERROR*        1RT     1407
 3265     0100 1442                   LJM  CALLGHM        MESSAGE AND ABORT                  1RT     1408
                                                                                             1RT     1409
 3267     1405              TTAP2     LDN   H.TFLGS       CORRECT ENTRY HAS BEEN FOUND       1RT     1410
 3270     5500 2517                   RAM   ESTNTRY+1     INCREMENT FOR FLAG WORD            1RT     1411
 3272     5400 2476                   STM   GOBIT1+1                                         1RT     1412
 3274     5400 2507                   STM   GOBIT+1       STORE FOR LATER USE                1RT     1413
 3276     1063                        SHN   -12                                              1RT     1414
 3277     2100 2000                   ADC   2000B                                            1RT     1415
 3301     5400 2516                   STM   ESTNTRY                                          1RT     1416
 3303     5400 2475                   STM   GOBIT1                                           1RT     1417
 3305     5400 2506                   STM   GOBIT         STORE FOR LATER USE                1RT     1418
                                                                                             1RT     1419
                                                                                             1RT     1420
 3307     2000 5670                   LDC   3000D         SET WAIT TO 100 MS                 1RT     1421
 3311     3412                        STD   D.T2                                             1RT     1422
 3312     1400                        LDN   0                                                1RT     1423
 3313     3405                        STD   WC            WORD COUNT                         1RT     1424
                                                                                             1RT     1425
 3314     0200 1161         STCK      RJM  STS            GET STATUS                         1RT     1426
 3316     1202                        LPN   BUSY                                             1RT     1427
 3317     0404                        ZJN  NBSY           GO AHEAD IF NOT BUSY               1RT     1428
 3320     3712                        SOD   D.T2                                             1RT     1429
 3321     0704                        MJN  PRA3           IF TIME IS UP, ASSUME REWINDING    1RT     1430
 3322     0371                        UJN  STCK           GO TRY AGAIN                       1RT     1431
 3323     0100 3361         NBSY      LJM  PRS2           GO AHEAD IF NOT BUSY               1RT     1432
 3325     0200 1076         PRA3      RJM  RELEQ          RELEASE EQUIP. AND DROP CHANNEL.   1RT     1433
 3327     3631              PRA3A     AOD   D.FNT+9       SET UP FOR 2 SEC. DELAY.           1RT     1434
 3330     3057                        LDD   D.FA                                             1RT     1435
 3331     1601                        ADN   1                                                1RT     1436
 3332     6225                        CHD   D.FNT+5                                          1RT     1437
 3333     2000 0311                   LDC   2RCI          RESTORE CIO TO THE INPUT REGISTER  1RT     1438
 3335     3450                        STD   D.PPIRB       * COPY IN DIRECT CELLS.            1RT     1439
 3336     3051                        LDD   D.PPIRB+1                                        1RT     1440
 3337     1277                        LPN   77B                                              1RT     1441
 3340     2100 1700                   ADC   1RO*100B                                         1RT     1442
 3342     3451                        STD   D.PPIRB+1                                        1RT     1443
 3343     3075                        LDD   D.PPMES1      WRITE CIO CALL INTO MSG BUFFE9     1RT     1444
                                                                                             1RT     1445
                   0031117  OV.CIO    EQU   OV.CIO                                           1RT     1446
                                                                                             1RT     1447
 3344     6250                        CHD   D.PPIRB                                          1RT     1448
 3345     1400                        LDN   P.ZERO                                           1RT     1449
 3346     6010                        CRD   D.T0                                             1RT     1450
 3347     1402                        LDN   2             DELAY 2 SECONDS                    1RT     1451
 3350     3411                        STD   D.T1                                             1RT     1452
 3351     1437                        LDN   M.RPJ                                            1RT     1453
 3352     0200 0516                   RJM  R.MTR                                             1RT     1454
 3354     1412                        LDN   M.DPP         DROP PP                            1RT     1455
 3355     0200 0516                   RJM  R.MTR                                             1RT     1456
 3357     0100 0103                   LJM  R.IDLE         EXIT TO IDLE LOOP                  1RT     1457
                                                                                             1RT     1458
 3361     5000 1120         PRS2      LDM   CONECT                                           1RT     1459
 3363     1013                        SHN   11            CHECK FOR ACCESS VIA MMTC.         1RT     1460
```

```
      3364    0705                  MJN  MHTC               JUMP IF 65X TRANSPORT.              1RT     1461
      3365    1440         CLRR1    LDN    CLRR                                                 1RT     1462
      3366    0200 1053             RJM  FCN                CLEAR REVERSE READ.                 1RT     1463
      3370    0327                  UJN  DENSITY            JUMP IF 60X TRANSPORT.              1RT     1464
                                                                                                1RT     1465
      3371    1444         MMTC     LDN    CLCM                                                 1RT     1466
      3372    0200 1053             RJM  FCN                CLEAR CONVERSION MODE.              1RT     1467
      3374    1400                  LDN    0                SET FCN ROUTINE TO RETURN ON A      1RT     1468
      3375    5400 1063             STM    MMTCSW1          * CONVERTER REJECT.                 1RT     1469
      3377    1403         MMTC1    LDN    DEN556                                               1RT     1470
      3400    0200 1053             RJM  FCN                ATTEMPT 556 DENSITY(VERIFY 657).    1RT     1471
      3402    0506                  NJN  MMTC2              JUMP IF 556 CPI IS REJECTED.        1RT     1472
      3403    2000 0371             LDC    3778+FCN1-MMTCSW1                                    1RT     1473
      3405    5400 1063             STM    MMTCSW1          RESET FCN ROUTINE.                  1RT     1474
      3407    0355                  UJN  CLRR1              JUMP IF UNIT IS A 657.              1RT     1475
      3410    1407         MMTC2    LDN    DEN1600                                              1RT     1476
      3411    0200 1053             RJM  FCN                ATTEMPT 1600 DENSITY(CHECK FOR 658/9) 1RT   1477
      3413    0563                  NJN  MMTC1              JUMP ON HDWE FAULT.                 1RT     1478
      3414    1441                  LDN    ERRN33           *FILE MAY NOT RESIDE ON DEV ASSIGNED* 1RT   1479
      3415    0100 1442             LJM  CALL6WM                                                1RT     1480
                                                                                                1RT     1481
      3417    3020         DENSITY  LDD    D.FNT                                                1RT     1482
      3420    1203                  LPN    3                ISOLATE DENSITY FLAGS FROM FNT.     1RT     1483
      3421    0407                  ZJN  SET                JUMP IF 556 BPI.                    1RT     1484
      3422    1201                  LPN    1                                                    1RT     1485
      3423    0503                  NJN  D200               JUMP IF 200 BPI.                    1RT     1486
      3424    1406                  LDN    6                SET DENSITY FOR 800 BPI.            1RT     1487
      3425    0304                  UJN  SETA                                                   1RT     1488
      3426    5600 3430    D200     AOM    SET              SET DENSITY FOR 200 BPI.            1RT     1489
      3430    1403         SET      LDN    3                                                    1RT     1490
      3431    0200 1053    SETA     RJM  FCN                SET THE DENSITY.                    1RT     1491
      3433    3031                  LDD    D.FNT+9                                              1RT     1492
      3434    1270                  LPN    70B              CHECK FOR READ PHYSICAL RECORD      1RT     1493
      3435    0413                  ZJN  PRS8                                                   1RT     1494
      3436    1720                  SBN    20B              CHECK FOR READ SKIP REQUEST         1RT     1495
      3437    0507                  NJN  XBX                JUMP IF NOT A READSKP.              1RT     1496
      3440    5400 2151             STM    RDSKIP           IF READ SKIP ZERO JUMP TO CAUSE     1RT     1497
      3442    5400 1376             STM    RDSKP                                                1RT     1498
      3444    5400 2040             STM    RDSK             DONT WRITE OLD IN INTO NEW IN       SC30019    2
      3446    0100 3524    XBX      LJM  PRS3                                                   1RT     1499
                                                                                                1RT     1500
      3450    3062         PRS8     LDD    D.IN             SET IN = OUT                        1RT     1501
      3451    3464                  STD    D.OUT                                                1RT     1502
      3452    3413                  STD    D.T3                                                 1RT     1503
      3453    3063                  LDD    D.IN+1                                               1RT     1504
      3454    3465                  STD    D.OUT+1                                              1RT     1505
      3455    3414                  STD    D.T4                                                 1RT     1506
      3456    1400                  LDN    0                                                    1RT     1507
      3457    3410                  STD    D.T0                                                 1RT     1508
      3460    3411                  STD    D.T1                                                 1RT     1509
      3461    3412                  STD    D.T2                                                 1RT     1510
      3462                          LOCA   D.PPIRB+3                                            1RT     1511
      3470    1603                  ADN    3                                                    1RT     1512
      3471    6210                  CHD    D.T0             SET OUT POINTER TO DISCARD OLD DATA. 1RT    1513
      3472    1400                  LDN    0                ZERO JUMP INST                      1RT     1514
      3473    5400 2114             STM    PHYS                                                 1RT     1515
      3475    5400 1552             STM    RPHRSW1          SET DIVIDE BY 5 CODE FOR A RPHR.    1RT     1516
```

10-77

10-78

```
3477      5400 1642                   STM    RDOKA                                      1RT       1517
3501      5400 2525                   STM    RCVR6                                      1RT       1518
3503      5400 2526                   STM    RCVR6+1                                    1RT       1519
3505      2000 0324                   LDC    3008+WTCM-RPHRSW3                          1RT       1520
3507      5400 1510                   STM    RPHRSW3          SET D.C.E. ROUTINE FOR A RPHR.  1RT   1521
3511      3031                        LDD    D.FNT+9          IF BCD MODE, INITIALIZE   1RT       1522
3512      1202                        LPN    BINMOD                                     1RT       1523
3513      0503                        NJN    *+3                                        1RT       1524
3514      5600 1353                   AOM    RDS              RESET FOR BCD PARITY      1RT       1525
3516      2000 0307                   LDC    WTDATA-RPHRFIX+3008   INHIBIT TAPE MOTION ON RPHR.  1RT  1526
3520      5400 1716                   STM    RPHRFIX                                    1RT       1527
3522      0100 3600                   LJM    PRS4                                       1RT       1528
                                                                                        1RT       1529
3524      3031             PRS3       LDD    D.FNT+9                                    1RT       1530
3525      1202                        LPN    BINMOD                                     1RT       1531
3526      0403                        ZJN    *+3                                        1RT       1532
3527      0100 3600                   LJM    PRS4             SKIP IF BINARY - ELSE SET 1RT       1533
                                                                                        1RT       1534
3531      5400 1700                   STM    BCDPAS1          SET FOR SCOPE TAPE.       1RT       1535
3533      2000 2567                   LDC    BYTE1            MODIFY BLANK TAPE SCAN FOR BCD.  1RT  1536
3535      5400 1431                   STM    BLANK3                                     1RT       1537
3537      3033                        LDD    D.EST+1          CHECK FOR 6684            1RT       1538
3540      1240                        LPN    40B                                        1RT       1539
3541      0410                        ZJN    PRS1A            GO SET BCD MODE IF NO 6684.  1RT    1540
3542      2000 2774                   LDC    B184                                       1RT       1541
3544      5400 1431                   STM    BLANK3           MODIFY BLANK TAPE SCAN FOR BCD/6684.  1RT  1542
3546      1410                        LDN    10B              SET UP CONVERSION TO DISPLAY CODE  1RT  1543
3547      5500 1357                   RAM    RD1+1                                      1RT       1544
                                                                                        1RT       1545
3551      2000 1200        PRS1A      LDC    BCDPRU           RESET BUFLOC, AND RESET THE NOISE  1RT  1546
3553      5400 1462                   STM    PRUSIZ           * TEST.                   1RT       1547
3555      2000 0200                   LDC    BCDCM            RESET FOR READSKP.        1RT       1548
3557      5400 1400                   STM    RDSKP1                                     1RT       1549
3561      2077 7577                   LDC    -BCDCM           SET THE BCD RECORD SIZE (IN CM WORDS)  1RT  1550
3563      5400 1374        MERGE1     STM    CMPRU            * INTO THE INSTRUCTIONS THAT CHECK  1RT  1551
3565      1001                        SHN    1                * THE CIR.BUF. FOR ROOM. CMPRUSZ  1RT  1552
3566      5400 1715                   STM    CMPRUSZ          * LOOKS FOR ROOM FOR 2 RECORDS.  1RT  1553
3570      5600 1353                   AOM    RDS              SET BCD MODE.             1RT       1554
3572      2000 3145                   LDC    BCDBUF                                     1RT       1555
3574      5400 1764                   STM    BUFLOC                                     1RT       1556
3576      5400 1464                   STM    RD3+1            IN CASE OF 6684           1RT       1557
3600                       PRS4       DIF    D.LIMIT,D.FIRST  SET BUFFER LENGTH AND GO  1RT       1558
3605      5400 1713                   STM    BUFLTH           START THE READ            1RT       1559
3607      5400 1372                   STM    SPACECK                                    1RT       1560
3611      1063                        SHN    -12                                        1RT       1561
3612      5500 1712                   RAM    BUFLTH-1                                   1RT       1562
3614      5400 1371                   STM    SPACECK-1                                  1RT       1563
3616      0100 1362                   LJM    PREREAD                                    1RT       1564
                                                                                        1RT       1565



3620                                  END                                               1RT       1567

                036277     UNUSED STORAGE          2142 STATEMENTS      1026  SYMBOLS
```

```
                           ERROR DIRECTORY

         9 TYPE ERROR              MICRO SUBSTITUTION ERROR OR ; CONVERTED TO "
               OCCURRED ON PAGES          26
```

10-79

```
BCDBUF     0003145          001417,   002640,   002666,   002763,   002775,   003015,   003034,   003572
BCDC       0002400          000000
BCDCM      0000200          003555,   003561
BCDPAS1    0001700          003531
BCDPRU     0001200          000000,   002640,   003015,   003551
BINBUF     0002540          001463,   001763
BINCM      0001000          001373,   001377,   001714
BINMOD     0000002          003512,   003525
BINPRU     0005000          000000,   001461
BKSPS      0000012          002400,   002405,   002410
BLANK2     0001430          001342,   001433
BLANK3     0001431          003535,   003544
BLANK4     0001436          001343
BLANK5     0001437          001344,   001434
BLANK6     0001434          001307
BUFLOC     0001764          001671,   002001,   003574
BUFLTH     0001713          003605,   003612
BUSY       0000002          002461,   003316
BYTE1      0002567          001323,   002643,   002645,   003533
BYTE2      0002605          001324,   002647
BYTE3      0002612          001325,   002651
B184       0002774          003017,   003542
B484       0002776          001422,   003003
B584       0003005          001337,   003042
CALL6WM    0001442          002105,   003265,   003415
CEF1       0002533          002354
CEF2       0002534          003157
CEF3       0002535          001346,   003177
CEF4       0002536          002334
CEF5       0002537          002347
CFGO       0001002          001014
CFR        0001007          001055,   001121
CFRA       0001233          001034
CFRB       0001237          001041,   002465
CFRX       0001006          001050
CFR1       0001010          001301
CFR10      0001043          001030,   001033,   001036
CFR11      0001045          001023
CFR2       0001026          001004
CFR6       0801031          001021
CFR7       0001034          001025
CFR9       0001037          001016
CH         0000046          001104,   001345,   003202
CLCM       0000044          003371
CLRR       0000040          003365
CLRR1      0003365          003407
CMPRU      0001374          003563
CMPRUSZ    0001715          003566
CONECT     0001120          003146,   003361
CS         0000004          001013,   001170,   001624,   002333
CVSW       0001417          002116
CVTBL      0003045          002576,   002601,   002621,   002624,   002677,   002702,   002714,   002717,   002736,
                            002741
CVT2       0002671          002751
CVT4       0002743          002727
CVT5       0002753          002750
CVT6       0002747          002760,   003040
```

10-80

```
CVT7      0002761               002753,  003027
C.CPEF    0000001   SCPTEXT     001156
C.PPFWA   0001000   SCPTEXT     003620,  000000
C.PP6WA   0006000   SCPTEXT     001453,  001457
DCE       0002104               002101
DCECNT    0000004               001471,  001532
DCEP      0001511               001514
DCEP1     0001525               001517
DCEP2     0001520               001633
DCEXIT    0001500               001533,  002770
DENSITY   0003417               003370
DEN1600   0000007               003410
DEN556    0000003               003377
DLYMCL    0001066               001037,  002340
DLYMCLX   0001065               001074
DLYMCL1   0001070               001071
DLYMCL2   0001072               001306
DORKIN    0002025               002453
D200      0003426               003423
D.BA      0000040   SCPTEXT     000000,  000000,  002045,  002060,  002237,  002241,  002242,  002244,  002257,
                                002315,  002330
D.CPAD    0000076   SCPTEXT     001125,  001200
D.EST     0000032   SCPTEXT     000000,  001132,  001134,  003145,  003151,  003537
D.FA      0000057   SCPTEXT     000000,  002245,  002316,  003330
D.FIRST   0000060   SCPTEXT     002005,  002012,  002016,  002021,  003601,  003604
D.FNT     0000020   SCPTEXT     000000,  000000,  000000,  001500,  001503,  001520,  001522,  001627,  001643,
                                001646,  001650,  002026,  002030,  002154,  002164,  002174,  002203,  002205,
                                002206,  002223,  002225,  002226,  002232,  002236,  002243,  002246,  002250,
                                002300,  002313,  002317,  002321,  002372,  002375,  003152,  003156,  003230,
                                003327,  003332,  003417,  003433,  003511,  003524
D.FR5     0000045   SCPTEXT     000000
D.FR6     0000046   SCPTEXT     000000
D.FR7     0000047   SCPTEXT     000000
D.IN      0000062   SCPTEXT     001363,  001366,  001704,  001707,  001726,  001731,  001745,  001747,  001755,
                                001762,  001772,  001774,  002017,  002022,  002032,  002037,  002046,  002050,
                                003450,  003453
D.LIMIT   0000066   SCPTEXT     001725,  001730,  002074,  002077,  003600,  003603
D.OUT     0000064   SCPTEXT     001362,  001365,  001703,  001706,  002067,  002072,  002076,  003451,  003454
D.PPIR    0000074   SCPTEXT     002311,  003174
D.PPIRB   0000050   SCPTEXT     002052,  002057,  002251,  002256,  002303,  002304,  002310,  002312,  002322,
                                002327,  003203,  003210,  003335,  003336,  003342,  003344,  003462,  003467
D.PPMES1  0000075   SCPTEXT     002353,  003343
D.PPONE   0000070   SCPTEXT     000000,  000000
D.RA      0000055   SCPTEXT     000000,  001760,  002010,  002035,  002055,  002254,  002325,  003206,  003465
D.SV2     0000072   SCPTEXT     000000
D.TH5     0000035   SCPTEXT     000000
D.TH6     0000036   SCPTEXT     000000
D.TH7     0000037   SCPTEXT     000000
D.TR      0000073   SCPTEXT     000000,  001202,  003173
D.T0      0000010   SCPTEXT     001124,  031127,  001156,  001402,  001404,  001472,  002024,  002062,  002064,
                                002352,  002354,  002423,  002435,  002477,  002500,  002505,  002510,  002520,
                                002521,  002552,  002553,  002660,  003021,  003212,  003346,  003457,  003471
D.T1      0000011   SCPTEXT     001143,  001271,  001564,  001567,  001732,  001741,  001743,  001763,  001776,
                                002000,  002360,  002573,  002576,  002616,  002621,  002674,  002677,  002711,
                                002714,  002733,  002736,  003213,  003222,  003350,  003460
```

```
1.1       1RT·  'AD X ANC I TAPE DRIVER
                /MBOLIC REFFERENCE TABLF

D.T2      0000012   SCPTEXT     001133,  001744,  001765,  002013,  002015,  002027,  002040,  002042,  002356,
                                002575,  002601,  002620,  002624,  002676,  002702,  002713,  002717,  002735,
                                002741,  003311,  003320,  003461
D.T3      0000013   SCPTEXT     001737,  001767,  002031,  002047,  002065,  003452
D.T4      0000014   SCPTEXT     002051,  002071,  003455
D.T5      0000015   SCPTEXT     001746
D.T6      0000016   SCPTEXT     001447,  001750,  002272
D.T7      0000017   SCPTEXT     001452,  002275
D.Z0      0000000   SCPTEXT     003243,  003246,  003261
D.Z1      0000001   SCPTEXT     001206,  001212,  001215,  003241,  003244
D.Z2      0000002   SCPTEXT     001211,  001213,  001216,  003262
D.Z3      0000003   SCPTEXT     003247,  003255
D.Z4      0000004   SCPTEXT     000000
D.Z5      0000005   SCPTEXT     000000
D.Z6      0000006   SCPTEXT     000000
D.Z7      0000007   SCPTEXT     000000
EB1       0002645               001326,  002567
EB2       0002647               001327,  002605
EB3       0002651               001330,  002612
EB4       0002654               002632
EB5       0002666               002653,  002656,  002662
EB5*4     0003042               001336,  003005
ENDTAPE   0000040               001653,  002125
END84     0003031               002777
END84X    0003034               003044
EPPAS1    0002446               002366,  003220
ERRNO     0000021               001442
ERRN16    0000020               001441
ERRN33    0000041               003414
ERRN5     0000005               003264
ERRN8     0000010               002104
FSTASG    0000032               000000
ESTCH12   0000033               000000
ESTCH34   0000034               000000
ESTHOW    0000035               000000
ESTNTRY   0002516               002513,  003250,  003253,  003270,  003301
EXSTS     0001300               001171
FCN       0001053               001354,  002401,  002406,  002411,  003366,  003372,  003400,  003411,  003431
FCNX      0001052               001057,  001064
FCN1      0001055               001063,  003403
FETFN1    0000040               000000,  000000,  000000,  000000
FILL      0002556               001701
FILLA     0002564               002555
FILLB     0002546               002557
FILLC     0002552               002561
FN        0000007               001054,  001062
FSTDSP    0000027               000000
FSTEOP    0000020               000000,  000000,  000000,  000000
FSTFT1    0000025               000000
FSTFT2    0000026               000000
FSTORD    0000022               000000,  003256
FSTPPC    0000024               000000
FSTSEC    0000030               000000
FST2UN    0000021               000000
GOBIT     0002506               003274,  003305
GCPIT1    0002475               003272,  003303
IAM184    0002775               001334
```

78-01

```
IAM284      0..3021                     001335
IAN184      0003007                     001333
INIT        0001407                     001375
INREC       0001400                     001356
INT2        0002172                     001504,   002151,   002160,   002161
INT2A       0002174                     002150
INT3        0002177                     002173
INT4        0002214                     002176,   002211
INT5        0002162                     002153
IP.PTCN     0000013                     001010,   001072,   001100,   001103,   001116,   001162,   001164,   001165,   001166,
                                        001171,   001173,   001174,   001175,   001356,   001360,   001430,   001434,   001436,
                                        001437,   001463,   001472,   001613,   001615,   002101,   002103,   002423,   002435,
                                        002535,   002567,   002571,   002605,   002607,   002612,   002614,   002630,   002645,
                                        002647,   002651,   002660,   002775,   003005,   003007,   003021,   003042,   003201
IP.RPE1     0000014                     001407,   001644
IP.RPE2     0000010                     002370
LA          0000035                     001421,   001537,   001554,   001673,   001674,   002603,   002604,   002610,   002611,
                                        002626,   002634,   002655,   002747,   002765,   003001,   003010,   003013,   003014,
                                        003033,   003036
LD1         0001613                     001331,   001601,   001626
LD2         0001615                     001332
LEV         0000036                     001677,   002157,   002201,   002564,   003171
LF.TAPES    0000010     SCPTEXT         003260
L.PPHDR     0000005     SCPTEXT         001453
MASCL       0001700                     001072,   001437
MMTC        0003371                     003364
MMTCSW1     0001063                     003375,   003403,   003405
MMTC1       0003377                     003413
MMTC2       0003410                     003402
MSG         0001205                     001043,   002467
MSGA        0001251                     001202,   001207,   002471,   003154,   003163
MSG1        0001212                     001217
MSG2        0001200                     001214
M.DFM       0000001     SCPTEXT         002361
M.DPP       0000012     SCPTEXT         002260,   003354
M.RPJ       0000037     SCPTEXT         003351
NBSY        0003323                     003317
NORMAL1     0001556                     001552
OVLPST      0000045                     001717,   002111,   002121,   003170
OV.CIO      0031117     SCPTEXT         002301,   003344
OV.1RP      0342220     SCPTEXT         002267
OV.6WM      0412715     SCPTEXT         001455
PARITY      0002400                     001610
PAR10       0002511                     002523
PAUSE       0001153                     001107,   002511
PAUSEX      0001152                     001157
PHYS        0002114                     002110,   003473
PNTROK      0002107                     002100
PRA3        0003325                     003321
PRERD1      0001415                     001622,   002444
PREREAD     0001362                     002122,   003616
PRS         0003145                     001000
PRS1A       0003551                     003541
PRS2        0003361                     003323
PRS3        0003524                     003446
PRS4        0003600                     003522,   003527
PRS5        0003222                     003216
```

```
PRS7      0003236                       003225,   003232
PRS8      0003450                       003435
PRUSIZ    0001462                       001534,   001602,   002761,   003553
P.TAPES   0000014    SCPTEXT            003240
P.ZERO    0000000    SCPTEXT            002023,   003345
RBTA      0002657                       002642
RB1       0002571                       001316
RB2       0002607                       001317
RB3       0002614                       001320
RB4       0002630                       001321,   001425,   002636
RB5       0002660                       001322
RB55      0003027                       003023
RC        0000047                       001410,   001645,   002364,   003166
RCVR      0002333                       001616
RCVRA     0002410                       002371,   002377
RCVRB     0002413                       002403,   002414,   002416
RCVRC     0002444                       002413,   002414
RCVRE     0002346                       002337
RCVR3     0002457                       002446,   002462
RCVR6     0002525                       003501,   003503
RDCK      0001572                       001575
RDCKL     0001622                       001606
RDCKR     0001620                       001577
RDI       0001461                       001430
RDN1      0002101                       001340
RDN2      0002103                       001341
RDOK      0001624                       001612,   002531
RDOKA     0001642                       001630,   001635,   003477
RDOKB     0001644                       001641,   002455
RDOKC     0001630                       001635,   001637,   002525
RDOK1     0001667                       001651,   001654
RDS       0001353                       003514,   003570
RDSIN     0001352                       001415,   001723,   002420,   002432
RDSK      0002040                       003444
RDSKIP    0002151                       001504,   001506,   002142,   003440
RDSKP     0001376                       003442
RDSKP1    0001400                       003557
RDSX      0001351                       001361
RD1       0001356                       001310,   003547
RD2       0001360                       001311
RD3       0001463                       001312,   003576
RD4       0001472                       001313,   001466,   001477
RD44      0001474                       001475,   001525,   001527
RD44A     0001475                       002664,   003025
RD5       0002423                       001314,   002425
RD6       0002435                       001315,   002437
READ      0001703                       001670
READOK    0001721                       001716
READY     0000001                       001003
RECYC     0002267                       002235
RECYTF    0002300                       002170
RELEQ     0001076                       001045,   001443,   002136,   002342,   002463,   003325
RELEQX    0001075                       001111
REL1      0001100                       001277
REL2      0001103                       001300
RES       0001131                       001060,   002344,   002527,   003236
RESA      0001271                       001145
```

10-84

```
VER 1.1        1RT- READ X AND I TAPE DRIVER                                                       12/10/70      PAGE NO      42
                    SYMBOLIC REFERENCE TABLE

    RES1      0001132                    001123
    RES3      0001114                    001144,   001151
    RES4      0001116                    001276
    RETURN    0002136                    001413,   001523,   002115,   002126
    RLTH      0000072                    001540,   001544,   001546,   001550,   001551,   001555,   001556,   001560,   001562,
                                         001566,   001571,   001733,   001740,   001742,   001751,   001771
    RPHRFIX   0001716                    003516,   003520
    RPHRSW1   0001552                    001545,   001547,   003475
    RPHPSW3   0001510                    003505,   003507
    RTO       0002236                    001112,   002514
    RTDX      0002265                    001663,   002134,   002146,   002162,   002233
    R.DCH     0000577     SCPTEXT        001105
    R.DFM     0000671     SCPTEXT        002473
    R.IDLE    0000103     SCPTEXT        002263,   002331,   003357
    R.MTR     0000516     SCPTEXT        002362,   003352,   003355
    R.OVL     0000135     SCPTEXT        001455
    R.OVLJ    0000125     SCPTEXT        002276
    R.PROCES  0000516     SCPTEXT        002261
    R.RAFL    0000446     SCPTEXT        001154
    R.RCH     0000561     SCPTEXT        001136
    R.STB     0000663     SCPTEXT        001147
    SEG2      0002014                    002003
    SET       0003430                    003421,   003426
    SETA      0003431                    003425
    SKPSIZ    0001350                    001405,   001541
    SPACECK   0001372                    003607,   003614
    ST        0000006                    001002,   001017,   001176,   001607,   001652,   002124,   002140,   002346
    STAT      0001200                    001162
    STATUS    0000037                    001411,   001655,   001662,   001721,   002107,   002127,   002145,   002175,   002200,
                                         002213,   002214,   002231,   002451,   003167
    STAT1     0002230                    002217
    STAT2     0002226                    002222
    STCK      0003314                    003322
    STS       0001161                    001011,   001511,   001572,   002426,   002440,   002457,   003314
    STSA      0001220                    001026
    STSB      0001226                    001031
    STSX      0001160                    001177
    STS1      0001162                    001272
    STS2      0001164                    001273
    STS3      0001165                    001274
    STS4      0001166                    001275
    STS5      0001171                    001302
    STS6      0001173                    001303
    STS7      0001174                    001304
    STS8      0001175                    001140,   001305
    TABL      0002540                    002547,   002553
    TAPMARK   0000010                    002141
    TORET     0001413                    001376,   001401
    TTAP1     0003244                    003263
    TTAP2     0003267                    003257
    TURNEOF   0002124                    001620
    UPPAS1    0001664                    003226
    UPPAS1A   0001651                    003234
    WC        0000005                    001465,   001536,   001604,   001667,   002113,   002152,   002172,   002663,   002670,
                                         002671,   002704,   002706,   002721,   002723,   002743,   002746,   002755,   002757,
                                         002766,   003024,   003037,   003313
    WTCM      0001534                    001467,   001531,   002772,   003505
```

10-85

```
1.1        1RT- READ X AND I TAPE DRIVER
                SYMBOLIC REFERENCE TABLE

                                            001474,   001525
WTCMA      0001532                          001665,   001700,   001720,   001722,   002565,   003516
WTDATA     0001725                          002025
WT0        0002046                          001734
WT1        0001742                          001766
WT2        0001776                          001770
WT3        0002016                          001753,   001775
WT4        0002023                          002114
WT5        0002116                          002112
WT6        0002120                          001126
W.CPDFM    0000030       SCPTEXT            003267
W.TFLGS    0000005       SCPTEXT            003437
XBX        0003446
```

10-86

```
                                                                                                  ZOS053     457
                                    ********************************************************************** ZOS053     458
                                    *                                                             ZOS053     459
                                    *          OVERLAY 3SY  -  DRIVER FOR 844  MCD  AY  13        01SP400     54
                                    *                                                             ZOS053     461
                                    ********************************************************************** ZOS053     462
                                                                                                  ZOS053     463
                                                                                                  ZOS053     464
                                                                                                  ZOS053     465
                                    3SY       SEGMENT  DRIVORG                                    ZOS053     466
 6223                                         ORG      DRIVORG                                    ZOS053     467
                                                                                                  ZOS053     468
                                 10 MAXRRCT   SET     8                                           01SP400     55
                                                                                                  ZOS053     470
                                                                                                  RMSY         2
                                                                                                  RMSY         3
                                    *         COMMON DECK RMSY  -  DRIVER FOR 844 MCD  AY  13     RMSY         4
                                                                                                  RMSY         5
                                                                                                  RMSY         6
                                                                                                  RMSY         7
 6223        7203                             VFD     12/YPARAM                                   RMSY         8
 6224        7114                             VFD     12/YCTLR                                    RMSY         9
                                                                                                  RMSY        10
                                                                                                  RMSY        11
                                  0 YDIGIT    EQU     0                                           RMSY        12
                                                                                                  RMSY        13
                                                                                                  RMSY        14
                                    **        ACTIVATE CHANNEL                                    RMSY        15
                                                                                                  RMSY        16
 6225        0100 0000              YACTIVC   SUBR                                                RMSY        17
                                                                                                  RMSY        18
                                              IF      -DEF,RMSC                                   RMSY        19
 6227        1400                             LDN     P.ZERO            CLEAR REQUEST AREA        RMSY        20
 6230        6010                             CRD     D.T0 - D.T4                                 RMSY        21
 6231        1400                   YCHPOA    LDN     **                                          RMSY        22
 6232        0200 0455                        RJM  R.RCH                RESERVE A CHANNEL         RMSY        23
                                              ELSE                                                RMSY        24
                                              RJM  YLOAD                LOAD 7054 SOFTWARE        RMSY        25
                                              ENDIF                                               RMSY        26
 6234        5600 6241                        ADM     YDROPCA           ENABLE DROP SWITCH        RMSY        27
 6236        0366                             RETURN                                              RMSY        28
                                                                                                  RMSY        29
                                                                                                  RMSY        30
                                                                                                  RMSY        31
                                                                                                  RMSY        32
                                    **        RELEASE CHANNEL AND CLEAR UNIT RESERVE              RMSY        33
                                                                                                  RMSY        34
 6237        0100 0000              YDROPC    SUBR                                                RMSY        35
                                                                                                  RMSY        36
 6241        0301                   YDROPCA   UJN  *+1   OR   *+2       SWITCH                    RMSY        37
                                                                                                  RMSY        38
 6242        0374                             RETURN                    CHANNEL IS OFF - EXIT     RMSY        39
 6243        5700 6241                        SOM     YDROPCA           DEACTIVATE CHANNEL DROP   RMSY        40
 6245        2000 3010                        LDC     YDIGIT+10B        7054 FUNCTION COMPLETE    MSC00321     1
                             6246     YDROPCB EQU     *-1                                         MSC00322     1
 6247        7600                     YDROPCC FAN     **                                          MSC00321     2
 6250        1500                             LCN     0                                           MSC00322     2
```

10-87

10-88

```
6251          6500 6256          YDROPCC   IJM    YDROPCF,**                                        MSC00321    4
6253          1701                         SBN    1                                                 MSC00321    5
6254          0574                         NJN    YDROPCC              WAIT FOR TIME-OUT OR INACTIVE MSC00321    6
6255          7500                YDROPCE  DCN    **                   DEACTIVATE CHANNEL           MSC00321    7
6256                              YDROPCF  BSS    0                                                 MSC00321    8
                                           IF     -DEF,RMSC,2                                       RMSY       41
6256          1400               YDROPCG   LDN    **                                                RMSY       42
6257          0200 0465                    RJM  R.DCH                  DROP CHANNEL                 RMSY       43
6261          0355                         RETURN                                                   RMSY       44
                                                                                                    RMSY       45
                                                                                                    RMSY       46
                                                                                                    RMSY       47
                                           IF     -DEF,RMSC                                         RMSY       48
                                                                                                    RMSY       49
                                 **        COMPUTE ACCESS TIME                                      RMSY       50
                                 *         TARGET RB NUMBER IN A-REGISTER                           RMSY       51
                                                                                                    RMSY       52
6262          0100 0000          YHEAD     SUPR                                                     RMSY       53
                                                                                                    RMSY       54
6264          3404                         STD    D.Z4                 SAVE RB NUMBER               RMSY       55
6265          5041 1014                    LDM    SUPP,RPNR                                         RMSY       56
6267          1207                         LPN    7                                                 RMSY       57
6270          3401                         STD    D.Z1                 SET D.Z1 = RER NUMBER        RMSY       58
6271          3004                         LDD    D.Z4                                              RMSY       59
6272          0200 7004                    RJM  YGETACC                GET TARGET CYLINDER NO.      RMSY       60
6274          3243                         SBD    HPCS1                COMPARE WITH OLD             RMSY       61
6275          0664                         PJN  RETURN1                A-REG = ABSOLUTE VALUE OF    RMSY       62
6276          2377 7777                    LMC    -0                    DIFFERENCE OF CYLINDER NUMBERS RMSY    63
6300          0361                         RETURN                                                   RMSY       64
                                                                                                    RMSY       65
                                           ENDIF                                                    RMSY       66
                                                                                                    RMSY       67
                                                                                                    RMSY       68
                                                                                                    RMSY       69
                                 **        INITIATE HEAD MOTION IF NECESSARY                        RMSY       70
                                 *         HPCS1 HAS CURRENT POSITION (LAST BIT SET)                RMSY       71
                                 *                OR MEANINGLESS (LAST BIT UNSET)                   RMSY       72
                                                                                                    RMSY       73
6301          0100 0000          YSHIFT    SUPR                                                     RMSY       74
                                                                                                    RMSY       75
6303          3404                         STD    D.Z4                 SAVE RB NUMBER               RMSY       76
6304          5012 1014                    LDM    SUPP,UPOCM                                        RMSY       77
6306          1207                         LPN    7                                                 RMSY       78
6307          3401                         STD    D.Z1                 SET D.Z1 = RER NO.           RMSY       79
6310          2300 0603                    LDC    .PJN.+YPOSITF-YPOSINST                            RMSY       80
6312          5400 6370                    STM    YPOSINST             SET FIRST TIME SWITCH IN YPOSIT RMSY    81
                                           IFLT   MAXRECT,8,2                                       RMSY       82
                                           LDN    MAXRECT                                           RMSY       83
                                           STM    RBCCUNT              INITIALIZE RECCUNT           RMSY       84
6314          3004                         LDD    D.Z4                                              RMSY       85
6315          0200 7004                    RJM  YGETACC                GET NEW CYLINDER NUMBER AND  RMSY       86
                                                                        SET UP ADDREG               RMSY       87
                                           IF     -DEF,RMSC,2                                       RMSY       88
6317          5444 6203                    STM    UNIT,CUR             SAVE NEW CYLINDER NUMBER     RMSY       89
                                           ELSE   1                                                 RMSY       90
                                           STM    RMDEVF+1                                          RMSY       91
6321          3243                         SBD    H[illegible]1        COMPARE WITH OLD CYLINDER NUMBER RMSY   92
```

10-89

```
6322          0456                       ZJN  RETURN1              EXIT IF SAME CYLINDER              RMSY        93
6323          0200 6737                  RJM  YPUTADD              PERFORM SEEK                       RMSY        94
6325          0363                       RETURN                                                       RMSY        95
                                                                                                      RMSY        96
                                                                                                      RMSY        97
                                                                                                      RMSY        98
                              *          MAKE INSTRUCTION REVISIONS FOR READS                         RMSY        99
                                                                                                      RMSY       100
6326          0100 0000       YRDINST    SUBR                                                         RMSY       101
                                                                                                      RMSY       102
6330          2300 6335                  LDC     YRDLIST                                              RMSY       103
6332          0200 4623                  RJM  MODIFY                                                  RMSY       104
6334          0371                       RETURN                                                       RMSY       105
                                                                                                      RMSY       106
                                                                                                      RMSY       107
6335          6354            YRDLIST    VFD     12/YRWLIST                                           RMSY       108
6336          0004            YRDLIST1   VFD     12/YDIGIT+4       READ FUNCTION                      RMSY       109
6337          7100            YRDLIST2   VFD     12/.IAM.                                             RMSY       110
6340          0302                       UJN  YRDISK9-YRDISK8                                         RMSY       111
                                                                                                      RMSY       112
                                                                                                      RMSY       113
                              *          MAKE INSTRUCTION REVISIONS FOR WRITES                        RMSY       114
                                                                                                      RMSY       115
6341          0100 0000       YWTINST    SUBR                                                         RMSY       116
                                                                                                      RMSY       117
6343          2300 6350                  LDC     YWTLIST                                              RMSY       118
6345          0200 4623                  RJM  MODIFY                                                  RMSY       119
6347          0371                       RETURN                                                       RMSY       120
                                                                                                      RMSY       121
                                                                                                      RMSY       122
6350          6354            YWTLIST    VFD     12/YRWLIST                                           RMSY       123
6351          0005            YWTLIST1   VFD     12/YDIGIT+5       WRITE FUNCTION                     RMSY       124
6352          7300            YWTLIST2   VFD     12/.OAM.                                             RMSY       125
6353          6600            YWTLIST3   VFD     12/.FJM.                                             RMSY       126
                                                                                                      RMSY       127
                                                                                                      RMSY       128
                                                                                                      RMSY       129
                                                                                                      RMSY       130
                              **         READ/WRITE MODIFY ADDRESS LIST                               RMSY       131
                                                                                                      RMSY       132
6354          6466            YRWLIST    VFD     12/YRDISK2                                           RMSY       133
6355          6505                       VFD     12/YRDISK7                                           RMSY       134
6356          6510                       VFD     12/YRDISK8                                           RMSY       135
6357          0000                       DATA    0                                                    RMSY       136
                                                                                                      RMSY       137
                                                                                                      RMSY       138
                                                                                                      RMSY       139
                              **         COMPUTE NORMAL PRU COUNT                                     RMSY       140
                                                                                                      RMSY       141
6360          0100 0000       YSECOMP    SUBR                                                         RMSY       142
                                                                                                      RMSY       143
6362          1467                       LCN     55            PRU-S/RB - 1                           RMSY       144
6363          0374                       RETURN                                                       RMSY       145
                                                                                                      RMSY       146
                                                                                                      RMSY       147
                                                                                                      RMSY       148
                              **         POSITION UNIT FOR NEXT PRU                                   RMSY       149
```

10-90

```
STACK PROCESSOR - SCOPE 3 MASS STORAGE INPUT/OUTPUT PACKAGE   COMPASS - VER 2.   09/17/73  17.37.21.     PAGE   151
                    OVERLAY 3SY - DRIVER FOR 844 MDC  AY  13

                                       *          SWITCH HEAD OR EXIT TO REISSUE THE REQUEST               RMSY       150
                                       *           FOR HEAD REPOSITIONING                                  RMSY       151
                                                                                                           RMSY       152
6364        0100 0000          YPOSIT    SUBR                                                              RMSY       153
                                                                                                           RMSY       154
6366        3073                         LDD     LASTPRU                                                   RMSY       155
6367        3254                         SBD     PRU                    JUMP IF NOT END OF RB              RMSY       156
6370        0603               YPOSINST  PJN   YPOSITF OR RETURN$                                          RMSY       157
6371        0200 0000          YPRUINST  RJM   **                       ADVANCE RECORD BLOCK - PRURD/PRUWT RMSY       158
6373        0200 5163          YPOSITF   RJM   SECCMP                   SET LASTPRU FOR NEW RB             RMSY       159
6375        3033                         LDD     WORKA                                                     RMSY       160
6376        0472                         ZJN   YPRUINST                 JUMP IF NO RB BYTE RETURNED        RMSY       161
                                         IFLT    MAXRECT,8,2                                               RMSY       162
                                         SOM     RBCOUNT                JUMP IF RBCOUNT = 0                RMSY       163
                                         MJN   YPOSITC                                                     RMSY       164
6377        2000 0573                    LDC     .PJN.+77B+RETURN$-YPOSINST.                               RMSY       165
6401        5400 6370                    STM     YPOSINST               CLEAR FIRST TIME SWITCH            RMSY       166
6403        5041 1014                    LDM     SUBF,RPNR                                                 RMSY       167
6405        1207                         LPN     7                                                         RMSY       168
6406        3401                         STD     D.Z1                   SET UP RBR NO. FOR YGETACC         RMSY       169
6407        3033                         LDD     WORKA                  NEW RB NO.                         RMSY       170
6410        0200 7004                    RJM   YGETACC                  GET NEW CYLINDER NUMBER            RMSY       171
6412        3243                         SBD     HPOS1                  COMPARE WITH CURRENT CYLINDER      RMSY       172
6413        0403                         ZJN   YPOSITI                  JUMP IF SAME CYLINDER              0134MSY      1
6414        0100 3674          YPOSITC   LJM   NOTDONE                  GO REISSUE REQUEST                 0134MSY      2
6416        0200 6737          YPOSITI   RJM   YPUTACC                  PERFORM THE SEEK FOR THIS RB       RMSY       184
6420        0443                         ZJN     RETURN$                                                   MSC00338     1
6421        1302                         SCN     2                                                         MSC00338     2
6422        0473                         ZJN     YPOSITI                WAIT FOR NOT BUSY                  MSC00338     3
6423        1006                         SHN     17-11           POSITION ABNORMAL STATUS BIT              MSC00338     4
6424        0703                         MJN     YPOSITG                JUMP IF DRIVE ERROR                MSC00338     5
                                         IF      -DEF,RMSC                                                 0134MSY      4
                                       *                         REISSUE STACK REQUEST IF UNIT RESERVED OR MSC00338     6
6425        0100 3665                    LJM   BYPASS                   ON TIMEOUT, 1SX WAS ALREADY        0134MSY      6
                                                                          CALLED BY YPAN                   0134MSY      7
                                         ELSE                                                              0134MSY      8
                                         LJM   ERRCALL                                                     0134MSY      9
                                         ENDIF                                                             0134MSY     10
                                                                                                           MSC00338     7
6427        0200 6672          YPOSITG   RJM     YSTSC           GET DETAIL STATUS                         MSC00338     8
6431        0200 7056                    RJM     YERRINF                                                   MSC00338     9
6433        1400                         LDN     0                                                         RMSY       192
6434        2410                         STD     D.T0                   CLEAR ABORT FLAG FOR ROUTINE -CALL- RMSY      193
6435        1473                         LDN     DEROR                                                     RMSY       194
6436        0100 3663                    LJM   ERRCALL                                                     RMSY       195
                                                                                                           RMSY       196
                                                                                                           RMSY       197
                                                                                                           RMSY       198
                                       **         READ OR WRITE DISK - INCLUDES PARITY ERROR TESTS AND     RMSY       199
                                       *           ERROR CORRECTION                                        RMSY       200
                                                                                                           RMSY       201
6440        3654               YRDISKX   AOD     PRU                    BUMP PRU                           RMSY       202
6441        1402                         LDN     2                                                         RMSY       203
6442        5500 7164                    RAM     ADDREG+1               BUMP SECTOR NUMBER                 RMSY       204
6444        1237                         LPN     37B                                                       RMSY       205
6445        1730                         SBN     24                                                        RMSY       206
6446        0704                         MJN   YR[illegible]K1          EXIT IF STILL ON SAME TRACK        [illegible] 207
```

```
6447         1450                         LDN    100B-24D          BUMP TRACK BY 1, RESET SECTOR        0413MSY    1
6450         5500 7164                    RAM    ADDREG+1                                               RMSY     209
6452         5000 7175        YRDISK1     LDM    WCTA                                                   RMSY     210
6454         3474                         STD    WDCT                                                   RMSY     211
                                                                                                        RMSY     212
6455         0100 0000        YRDISK      SUBR                                                          RMSY     213
                                                                                                        RMSY     214
6457         3074                         LDD    WDCT                                                   RMSY     215
6460         5400 7175                    STM    WCTA              MOVE BYTE COUNT FROM DIRECT CELL     RMSY     216
6462         1400                         LDN    0                                                      RMSY     217
6463         5400 6523                    STM    YSWITCH                                                RMSY     218
6465         2000 0304                    LDC    YDIGIT+4          7054 READ/WRITE FUNCTION             RMSY     219
                         6466   YRDISK2   EQU    *-1                                                    RMSY     220
6467         7600               YRDISK3   FAN    **                                                     RMSY     221
6470         5400 7162                    STM    YLAFN                                                  RMSY     222
6472         1500                         LCN    0                                                      0134MSY   12
6473         6500 6502          YRDISK4   IJM    YRDISK6,**        TIMEOUT - 1 SECOND                   0134MSY   13
6475         1701                         SBN    1                                                      RMSY     225
6476         0574                         NJN    YRDISK4                                                0134MSY   14
6477         7500               YRDISK4C  DCN    **                                                     MSC00322   3
6500         0100 6427          YRDISK5   LJM    YPOSITG           JUMP ON ERROR                        RMSY     227
                                                                                                        RMSY     228
6502         7400               YRDISK6   ACN    **                                                     RMSY     229
6503         2000 0502                    LDC    PRULENB+2         BYTE COUNT TO -A- REGISTER           RMSY     230
6505         7100 7174          YRDISK7   IAM    IOBUF,**          READ/WRITE PRU                       RMSY     231
6507         3412                         STD    D.T2              SAVE REMAINING WORD COUNT            RMSY     232
6510         6600 6510          YRDISK8   FJM    YRDISK8,**        (UJN YRDISK9 IF READING )            RMSY     233
6512         6500 6515          YRDISK9   IJM    YRDISK11,**                                            RMSY     234
6514         7500               YRDISK10  DCN    **                                                     RMSY     235
6515         0200 6641           YRDISK11 RJM    YSTSG      GET GENERAL STATUS                          MSC00338  10
6517         0516               YRDISK13  NJN    YRDISK17          JUMP IF ERROR                        RMSY     244
                                                                                                        RMSY     245
6520         3012                         LDD    D.T2                                                   RMSY     246
6521         0514                         NJN    YRDISK17          IF I/O OPERATION DID NOT COMPLETE    RMSY     247
                                                                    JUMP TO ERROR PROCESSING EVEN WITH  RMSY     248
                                                                    GOOD GENERAL STATUS                 RMSY     249
                                                                                                        RMSY     250
6522         2000 0000                    LDC    **                                                     RMSY     251
                         6523   YSWITCH   EQU    *-1                                                    RMSY     252
6524         0407                         ZJN    YRDISK16          JUMP IF NO RETRIES WERE MADE         RMSY     253
                                                                                                        RMSY     254
6525         5000 3706          YRDISK14  LDM    OPDEX                                                  RMSY     255
6527         0404                         ZJN    YRDISK16          EXIT IF PRECEDED BY UNCORRECTABLE    RMSY     256
                                                                    ERROR                               RMSY     257
6530         1403                         LDN    3                 FLAG RECOVERED ERROR                 RMSY     258
6531         5400 3706          YRDISK15  STM    OPDEX                                                  RMSY     259
6533         0100 6440          YRDISK16  LJM    YRDISKX           EXIT                                 RMSY     260
                                                                                                        RMSY     261
6535         0200 6672           YRDISK17 RJM    YSTSD      GET DETAIL STATUS FOR ERROR MESSAGE         MSC00338  11
6537         0200 7055                    RJM    YERRINF           PREPARE ERROR INFORMATION            RMSY     263
6541         3007                         LDD    ST                                                     RMSY     264
6542         1010                         SHN    17-9                                                   RMSY     265
6543         0603                         PJN    YRDISK18          JUMP IF NO CATASTROPHIC ERROR        RMSY     266
6544         1400               YRDIS17A  LDN    0                 FLAG UNCORRECTABLE ERROR             RMSY     267
6545         0364                         UJN    YRDISK15                                               RMSY     268
                                                                                                        RMSY     269
6546         1004               YRDISK18  SHN    9-5                                                    RMSY     270
```

10-91

```
STACK PROCESSOR - SCOPE 3 MASS STORAGE INPUT/OUTPUT PACKAGE   COMPASS - VER 2.     09/17/73  17.37.21.     PAGE   153
                  OVERLAY 3SY - DRIVER FOR 844 MCC  AY  13

6547      0711                          MJN  YRDISK21          JUMP IF CORRECTABLE DATA ERROR     RMSY      271
                                                                                                  RMSY      272
6550      1017                          SHN    5-8+18                                             0134MSY    15
6551      0672                          PJN  YRDIS17A          ASSUME UNCORRECTABLE ERROR IF NEITH RMSY     274
                                                                RECOVERY IN PROGRESS NOR          RMSY      275
                                                                CATASTROPHIC BITS ARE SET         RMSY      276
                                                                                                  RMSY      277
6552      2000 0014                     LDC    YDIGIT+14B      7054 CONTINUE FUNCTION             RMSY      278
                    6553    YRDISK19    EQU    *-1                                                RMSY      279
6554      5400 6523                     STM    YSWITCH         SET RETRY SWITCH                   RMSY      280
6556      0100 6467                     LJM  YRDISK3           GO REPEAT OPERATION                RMSY      281
                                                                                                  RMSY      282
                                                                                                  RMSY      283
                                                                                                  RMSY      284
                                                                                                  RMSY      285
6560      1410              YRDISK21    LDN    8                                                  RMSY      286
6561      3412                          STD    D.T2            COUNT FOR DIVIDE                   RMSY      287
6562      5000 7110                     LDM    YDETSTS+11                                         RMSY      288
6564      1004                          SHN    4                                                  RMSY      289
6565      0305                          UJN  YRDISK23                                             RMSY      290
                                                                                                  RMSY      291
                            *           DIVIDE BIT ADDRESS BY 12 TO FIND PPU BYTE ADDRESS         RMSY      292
                                                                                                  RMSY      293
6566      3010              YRDISK22    LDD    D.T0                                               RMSY      294
6567      1014                          SHN    12                                                 RMSY      295
6570      3111                          ADD    D.T1                                               RMSY      296
6571      1001                          SHN    1                                                  RMSY      297
6572      2164 0000         YRDISK23    ADC    -12S12+1        DECREMENT DIVIDEND,INCREMENT QUOTIE RMSY     298
6574      0675                          PJN  YRDISK23                                             RMSY      299
6575      2113 7777                     ADC    12S12-1         CORRECT LAST SUBTRACT              RMSY      300
6577      3411                          STD    D.T1                                               RMSY      301
6600      1063                          SHN    -12                                                RMSY      302
6601      3410                          STD    D.T0                                               RMSY      303
6602      3712                          SOD    D.T2                                               RMSY      304
6603      0662                          PJN  YRDISK22          JUMP IF NOT FINISHED               RMSY      305
                                                                                                  RMSY      306
                            *           REMAINDER IS IN D.T0, QUOTIENT IN D.T1                    RMSY      307
                            *           SET UP SHIFT INSTRUCTIONS AND MASKS TO PERFORM THE        RMSY      308
                            *            INCLUSIVE OR OPERATIONS                                  RMSY      309
                                                                                                  RMSY      310
6604      2000 1023                     LDC    .SHN.+19                                           RMSY      311
6606      3210                          SBD    D.T0                                               RMSY      312
6607      5400 6627                     STM    YRDISK25        BYTE2 SHIFT IS 19-REMAINDER        RMSY      313
6611      1654                          ADN    77B-19          BYTE1 SHIFT IS -REMAINDER          RMSY      314
                                                                (= 77B - REMAINDER)               RMSY      315
6612      5400 6617                     STM    YRDISK24                                           RMSY      316
6614      5000 7104                     LDM    YDETSTS+7       CORRECTION BYTE                    RMSY      317
6616      1001                          SHN    1               LEFT JUSTIFY                       RMSY      318
6617      1000              YRDISK24    SHN    **              SHN -REMAINDER                     RMSY      319
6620      3412                          STD    D.T2            SAVE PART 1                        RMSY      320
6621      5311 7174                     LMM    IOBUF,D.T1      CORRECT BYTE1                      RMSY      321
6623      5411 7174                     STM    IOBUF,D.T1                                         RMSY      322
6625      5000 7104                     LDM    YDETSTS+7                                          RMSY      323
6627      1023              YRDISK25    SHN    19-**           SHN 19-REMAINDER                   RMSY      324
6630      3312                          LMD    D.T2            CLEAR PART 1 OF CORRECTION BYTE    RMSY      325
6631      1071                          SHN    -6              POSITION                           RMSY      326
6632      5311 7175                     LMM    IOBUF+1,D.T1    CORRECT BYTE2                      RMSY      327
6634      5411 7175                     STM    IOBUF+1,D.T1
```

10-92

```
6636         0100 6525                   LJM  YRDISK14                                    RMSY        328
                                                                                          RMSY        329
                                                                                          RMSY        330
                                                                                          RMSY        331
                                *        GET STATUS                                       RMSY        332
                                *                                                         0134MSY      16
                                *        EXIT   A = GENERAL STATUS (WITHOUT BUSY BIT)     0134MSY      17
                                *                ( IF A-REG IS NON-ZERO WITHOUT BIT 11 (ABNORMAL  0134MSY  18
                                *                  TERMINATION) SET, THEN TIMEOUT OCCURRED IN YFAN  0134MSY  19
                                                                                          RMSY        333
6640         0100 0000        YSTSG      SUBR                                             MSC00338     12
                                                                                          RMSY        335
6642         1412                        LCN    128            7054 GENERAL STATUS FUNCTION  RMSY     336
6643         0200 6711                   RJM  YFAN                                        RMSY        337
6645         0572                        NJN  RETURN$                                     0134MSY      20
6646         7400             YSTS1      ACN    **                                        RMSY        339
6647         1500                        LCN    0                                         MSC00322      4
6650         6600 6661        YSTS1A     FJM    YSTS2,**       STATUS IS THERE            MSC00322      5
6652         1701                        SBN    1                                         MSC00322      6
6653         0574                        NJN    YSTS1A         WAIT TILL STATUS OR TIME-OUT  MSC00322   7
6654         1401                        LCN    1                                         MSC00322      8
6655         6500 6640        YSTS1B     IJM    RETURN$,**                                MSC00322      9
6657         7500             YSTS1C     DCN    **                                        MSC00322     10
6660         0357                        RETURN                                           MSC00322     11
                                                                                          MSC00322     12
6661         7000             YSTS2      IAN    **             INPUT STATUS BYTE          RMSY        340
6662         6500 6665        YSTS2A     IJM  YSTS3A,**                                   RMSY        341
6664         7503             YSTS3      DCN    **                                        RMSY        342
6665         3467             YSTS3A     STD    ST                                        RMSY        343
6666         5400 7160                   STM    STAT1                                     RMSY        344
6670         0347                        RETURN                                           MSC00338     13


                                **       GET DETAIL STATUS FROM THE BUFFERED CONTROLLER.  MSC00338     15
                                *                                                         MSC00338     16
                                *        EXIT   TWELVE WORDS OF DETAIL STATUS IN YDETSTS  MSC00338     17
                                                                                          MSC00338     18
6671         0100 0000        YSTSD      SUBR                                             MSC00338     19
6673         1413                        LDN    13B            7054 DETAIL STATUS FUNCTION  RMSY      347
6674         0200 6711                   RJM  YFAN                                        RMSY        349
6676         0572                        NJN  RETURN$                                     0134MSY      21
6677         1414                        LDN    12             LENGTH OF DETAIL STATUS    RMSY        350
6700         7400             YSTS4      ACN    **                                        RMSY        351
6701         7100 7075        YSTS5      IAM    YDETSTS,**     INPUT DETAIL STATUS        RMSY        352
6703         6500 6706        YSTS5A     IJM  YSTS7,**                                    RMSY        353
6705         7500             YSTS6      DCN    **                                        RMSY        354
6706         0362              YSTS7     RETURN                                           MSC00338     20
                                                                                          RMSY        357
                                                                                          RMSY        362
                                                                                          RMSY        363
                                *        FUNCTION AND TIME OUT A FULL CHANNEL CONDITION   RMSY        364
                                *                                                         0134MSY      22
                                *        EXIT   A .EQ. 0 IF FUNCTION ACCEPTED             0134MSY      23
                                *               A .NE. 0 IF NOT ACCEPTED IN 1 SECOND      0134MSY      24
                                                                                          0134MSY      25
```

10-93

10-94

```
 6707       1400                  YFANX     LDN    0               NORMAL EXIT                          0134MSY     26
                                                                                                        RMSY       365
 6710       0100 0000             YFAN      SUBR                                                        RMSY       366
                                                                                                        RMSY       367
 6712       2100 0000                       ADC    YDIGIT          ADD IN EQUIPMENT NUMBER              RMSY       368
                        6713      YFAN1     EQU    *-1                                                  RMSY       369
 6714       6400 6724             YFAN1A    AJM   YFAN4,**         JUMP IF CHANNEL ALREADY ACTIVE       0134MSY     27
 6716       7600                  YFAN2     FAN    **                                                   RMSY       371
 6717       1500                            LCN    0                                                    0134MSY     28
 6720       6500 6707             YFAN3     IJM   YFANX,**         TIMEOUT - 1 SECOND                   0134MSY     29
 6722       1701                            SBN    1                                                    RMSY       374
 6723       0574                            NJN   YFAN3                                                 RMSY       375
 6724       7500                   YFAN4    DCN    **                                                   MSC00322    13
 6725       1400                            LDN    0                                                    MSC00322    14
 6726       3410                            STD    D.T0            CLEAR ABORT FLAG FOR SUBROUTINE CAL  RMSY       376
 6727       0200 7356                       RJM   YERRINF                                               RMSY       377
 6731       1473                            LDN    DERCR           TEMPORARILY USE REJECT FOR TIME OUT  RMSY       378
                                            IF     DEF,RMSC,2                                           RMSY       379
                                            LJM   ERRCALL          ERROR EXIT IF TIMEOUT IN DEADSTART   RMSY       380
                                            ELSE   1                                                    RMSY       381
 6732       0200 6036                       RJM   CALL                                                  RMSY       382
 6734       1401                            LDN    1               EXIT WITH A-REG POSITIVE NON-ZERO    0134MSY     31
 6735       0352                            RETURN                                                      RMSY       384
                                                                                                        RMSY       385
                                                                                                        RMSY       386
                                                                                                        RMSY       387
                                  *         LOAD THE ADDRESS REGISTER                                   RMSY       388
                                  *                                                                     0134MSY     32
                                  *         ENTRY  ADDREG+0, +1 = DISK ADDRESS                          0134MSY     33
                                  *                                                                     0134MSY     34
                                  *         EXIT   A .EQ. 0 IF NO ERROR                                 0134MSY     35
                                  *                A .GT. 0 IF TIMEOUT OR ACTIVE CHANNEL WAS ENCOUNTERED 0134MSY     36
                                                                                                        RMSY       389
 6736       0100 0000             YPUTADC   SUBR                                                        RMSY       390
                                                                                                        RMSY       391
 6740       0200 6641              YPUTAD1  RJM    YSTSG       GET GENERAL STATUS                       MSC00338    22
 6742       1007                            SHN    17-10                                                MSC00338    23
 6743       0774                            MJN   YPUTAD1          JUMP IF OPPOSITE ACCESS RESERVED     0134MSY     46
 6744       1402                            LDN    2               7054 SELECT 2/1 INTERLACE FUNCTION   RMSY       392
 6745       5400 7162                       STM    YLAFN           SAVE FUNCTION CODE                   0134MSY     47
 6747       0200 6711                       RJM   YFAN                                                  RMSY       393
 6751       0564                            NJN   RETURN1          EXIT WITH A-REG NON-ZERO ON TIMEOUT  0134MSY     48
 6752       7400                  YPUTAD7   ACN    **                                                   0134MSY     49
                                  *         SET UP 4 PARAMETER PACKAGE FOR SEEK                         0413MSY      2
 6753       3044                            LDD    CUR                                                  0413MSY      3
 6754       3401                            STD    D.Z1            PARAM 1 = UNIT NO.                   0413MSY      4
 6755       5000 7163                       LDM    ADDREG                                               0413MSY      5
 6757       3402                            STD    D.Z2            PARAM 2 = CYLINDER NO.               0413MSY      6
 6760       5000 7164                       LDM    ADDREG+1                                             0413MSY      7
 6762       1014                            SHN    12                                                   0413MSY      8
 6763       3403                            STD    D.Z3            PARAM 3 = TRACK NO.                  0413MSY      9
 6764       1063                            SHN    -12                                                  0413MSY     10
 6765       3404                            STD    D.Z4            PARAM 4 = SECTOR NO.                 0413MSY     11
 6766       1404                            LDN    4                                                    0413MSY     12
 6767       7300 0001             YPUTAD8   OAM    D.Z1,**         SEND ADDRESS                         0413MSY     13
 6771       6500 6774             YPUTAD9   IJM   YPUTAD11,**                                           0134MSY     51
 6773       7500                  YPUTAD10  DCN                                                         01[illegible]SY     52
```

```
6774        0403                 YPUTAD11  ZJN  YPUTAD12            JUMP IF OUTPUT COMPLETE             0413MSY    14
6775        1401                           LDN    1                 EXIT WITH A = 1 IF INCOMPLETE OUTPU 0413MSY    15
6776        0303                           UJN  YPUTAD13                                                0413MSY    16
6777        0203 6641            YPUTAD12  RJM  YSTSG               GET GENERAL STATUS                  0413MSY    17
7001        0100 6736            YPUTAD13  RETURN                                                       0413MSY    18
                                                                                                        RMSY      403
                                                                                                        RMSY      404
                                                                                                        RMSY      405
                                 *         COMPUTE PHYSICAL ADDRESS                                     RMSY      406
                                 *         ON ENTRY A = RB NUMBER                                       RMSY      407
                                 *                  C.21 = RBR NO.                                      RMSY      408
                                                                                                        RMSY      409
7003        0100 0000            YGETADD   SUBR                                                         RMSY      410
                                                                                                        RMSY      411
7005        1016                           SHN    18-4                                                  0413MSY    19
7006        3404                           STD    D.74              SAVE CYLINDER NO. WITHIN RBR        0413MSY    20
7007        1001                           SHN    1                                                     0413MSY    21
7010        3400                           STD    D.70              SAVE ODD / EVEN PRU BIT             0413MSY    22
7011        1057                           SHN    1-17                                                  0413MSY    23
7012        3406                           STD    D.Z6              RB MOD 4                            0413MSY    24
7013        1002                           SHN    2                 *4                                  RMSY      416
7014        3106                           ADD    D.Z6                                                  RMSY      417
7015        3405                           STD    D.Z5              *5                                  RMSY      418
7016        3506                           RAD    D.Z6              *6                                  RMSY      419
7017        3000                           LDD    D.70                                                  0413MSY    25
7020        1201                           LPN    1                 ISOLATE ODD / EVEN PRU BIT          0413MSY    26
7021        3154                           ADD    PRU               SECTOR FROM START OF RB =           0413MSY    27
7022        3154                           ADD    PRU                2*PRU + ODD/EVEN BIT + OFFSET      0413MSY    28
7023        2100 0000                      ADC    **                ADD STARTING PRU FOR                RMSY      425
                         7024    YALOPAD   EQU    *-1                NON-STANDARD ALLOCATION            RMSY      426
7025        1630                           ADN    24                                                    RMSY      427
7026        3206                           SBD    D.Z6              A REG = STARTING SECTOR             RMSY      428
7027        1006                           SHN    6                 POSITION FOR DIVIDE BY 24           0413MSY    29
7030        2177 5300            YGETADD2  ADC    -24*100B+1        DECREMENT SECTOR, INCREMENT TRACK   0413MSY    30
7032        0675                           PJN  YGETADD2                                                0413MSY    31
7033        2100 2776                      ADC    +24*100B-2        CORRECT LAST SUBTRACT               0413MSY    32
7035        3505                           RAD    D.Z5              BITS 11-6 = SECTOR NO.,             0413MSY    33
                                                                    BITS  5-0 = TRACK NO.               0413MSY    34
7036        1377                           SCN    77B               CLEAR TRACK NO.                     0413MSY    35
7037        1006                           SHN    17-11             SECTOR NO. TO UPPER 6 BITS          0413MSY    36
7040        3305                           LMD    D.Z5              RE-INSERT TRACK NO. IN BITS 5-0     0413MSY    37
7041        1006                           SHN    6                 REPOSITION, BITS 11-6 = TRACK NO.   0413MSY    38
                                                                    BITS 5-0 = SECTOR NO.               0413MSY    39
7042        5400 7164                      STM    ADDREG+1                                              0413MSY    40
7044        3004                           LDD    D.Z4              CYLINDER NO. WITHIN RBR             0413MSY    41
7045        5101 7111                      ADM    YCYL,D.Z1         ADD STARTING CYLINDER OF RBR        0413MSY    42
7047        5400 7163                      STM    ADDREG            STORE PHYSICAL CYLINDER NO.         0413MSY    43
7051        1001                           SHN    1                 EXIT WITH A-REG BITS 11-1 = CYL NO. RMSY      453
7052        1601                           ADN    1                              BIT  0 = 1             RMSY      454
7053        0100 7003                      RETURN                                                       RMSY      455
                                                                                                        RMSY      456
                                                                                                        RMSY      457
                                                                                                        RMSY      458
                                 *         SET UP ERROR INFORMATION FOR 1SX                             RMSY      459
                                                                                                        RMSY      460
7055        0100 0000            YERRINF   SUBR                                                         RMSY      461
                                                                                                        RMSY      462
```

10-95

```
7057          5000 7075                    LDM    YDETSTS          MOVE FIRST TWO BYTES OF DETAILED    RMSY     463
7061          5400 7165                    STM    RETRYS            STATUS TO RETRYS AND STAT2         RMSY     464
7063          5000 7076                    LDM    YDETSTS+1                                            RMSY     465
7065          5400 7165                    STM    STAT2                                                RMSY     466
7067          0200 6013                    RJM  ERRINF                                                 RMSY     467
                                           IF     -DEF,RMSC                                            RMSY     468
7071          0463                         ZJN  RETURNS         EXIT IF PRECEDED BY PERMANENT ERROR   RMSY     469
7072          6371 7077                    CWM    YDETSTS+2,TWO    WRITE LAST TEN BYTES OF DETAILED    RMSY     470
                                                                    STATUS TO MESSAGE BUFFER           RMSY     471
                                           ENDIF                                                       RMSY     472
                                                                                                       RMSY     473
7074          0360                         RETURN                                                      RMSY     474
                                                                                                       RMSY     475
7075                           YDETSTS     BSSZ   12                                                   RMSY     476
                                           IF     -DEF,RMSC,2                                          RMSY     477
                        7162   YLAFN  -    EQU    LFUNC                                                RMSY     478
                                           ELSE   1                                                    RMSY     479
                               YLAFN       BSSZ   1                                                    RMSY     480
                                                                                                       RMSY     481
                                                                                                       RMSY     482
                                                                                                       RMSY     483
                               *           TABLE FOR MULTIPLY BY 137.  USED FOR FINDING STARTING CYLIND  RMSY   484
                               *            NUMBER FOR AN RBR                                          RMSY     485
                                                                                                       RMSY     486
7111          0000             YCYL        DATA   0                   *0                               RMSY     487
7112          0211                         DATA   137                 *1                               RMSY     488
7113          0422                         DATA   274                 *2                               RMSY     489
                                                                                                       RMSY     490
                                                                                                       RMSY     491
                        7114   YNSAORG     EQU    *                                                    RMSY     492
                                                                                                       RMSY     493
                                                                                                       RMSY     494
                                                                                                       RMSY     495
                               *           ADDRESS LIST  FOR CONTROLLER NUMBER INSERTION               RMSY     496
                                                                                                       RMSY     497
7114          0011             YCTLR       VFD    12/C.T1                                              RMSY     498
7115          6245                         VFD    12/YDRCPCE                                           MSC00322  16
7116          6351                         VFD    12/YWTLIST1                                          RMSY     499
7117          6335                         VFD    12/YRDLIST1                                          RMSY     500
7120          6465                         VFD    12/YRDISK2                                           RMSY     501
7121          6553                         VFD    12/YRDISK19                                          RMSY     503
7122          6713                         VFD    12/YFAN1                                             RMSY     504
7123          0000                         DATA   0                                                    RMSY     505
                                                                                                       RMSY     506
                                                                                                       RMSY     507
                                                                                                       RMSY     508
                               *           ADDRESS LIST FOR CHANNEL NUMBER INSERTION                   RMSY     509
                                                                                                       RMSY     510
7124          0311             YCHANT      VFD    12/D.T1                                              RMSY     511
7125          2046                         CHLIST                                                      RMSY     512
                                           IF     -DEF,RMSC,2                                          RMSY     513
7134          6231                         VFD    12/YCHECA                                            RMSY     514
7135          6256                         VFD    12/YDRCPCG                                           RMSY     515
7136          6247                         VFD    12/YDRCPCC                                           MSC00321   9
7137          6251                         VFD    12/YDRCPCD                                           MSC00321  10
7140          6255                         VFD    12/YDRCPCE                                           MSC00321  11
7141          6467                         VFD       RDISK3                                            RMS      520
```

350

```
7142      6473                       VFD    12/YRDISK4                          RMSY       521
7143      6477                       VFD    12/YRDISK4C                         MSC00322    17
7144      6502                       VFD    12/YRDISK6                          RMSY       522
7145      6505                       VFD    12/YRDISK7                          RMSY       523
7146      6510                       VFD    12/YRDISK8                          RMSY       524
7147      6512                       VFD    12/YRDISK9                          RMSY       525
7150      6514                       VFD    12/YRDISK10                         RMSY       526
7151      6352                       VFD    12/YWTLIST2                         RMSY       532
7152      6337                       VFD    12/YRDLIST2                         RMSY       533
7153      6353                       VFD    12/YWTLIST3                         RMSY       534
7154      6714                       VFD    12/YFAN1A                           0134MSY     54
7155      6715                       VFD    12/YFAN2                            RMSY       535
7156      6720                       VFD    12/YFAN3                            RMSY       536
7157      6724                       VFD    12/YFAN4                            MSC00322    18
7160      6646                       VFD    12/YSTS1                            RMSY       537
7161      6650                       VFD    12/YSTS1A                           MSC00322    19
7162      6655                       VFD    12/YSTS1B                           MSC00322    20
7163      6657                       VFD    12/YSTS1C                           MSC00322    21
7164      6661                       VFD    12/YSTS2                            RMSY       538
7165      6662                       VFD    12/YSTS2A                           RMSY       539
7166      6664                       VFD    12/YSTS3                            RMSY       540
7167      6700                       VFD    12/YSTS4                            RMSY       541
7170      6701                       VFD    12/YSTS5                            RMSY       542
7171      6703                       VFD    12/YSTS5A                           RMSY       543
7172      6705                       VFD    12/YSTS6                            RMSY       544
7173      6752                       VFD    12/YFLTAC7                          0134MSY     59
7174      6767                       VFD    12/YFUTAC8                          0134MSY     60
7175      6771                       VFD    12/YFUTAC9                          0134MSY     61
7176      6773                       VFD    12/YFLTAC10                         0134MSY     62
                                     IF     DEF,RMSC                            RMSY       549
                                     VFD    12/YLCAD4                           RMSY       550
                                     VFD    12/YLCAD5A                          RMSY       551
                                     VFD    12/YLCAD6                           RMSY       552
                                     VFD    12/YLCAD9                           RMSY       553
                                     VFD    12/YLCAD8C                          MSC00337     1
                                     VFD    12/YLCAD9A                          MSC00337     2
                                     VFD    12/YLCAD9C                          MSC00337     3
                                     VFD    12/YCUT1                            RMSY       554
                                     VFD    12/YCUT2                            RMSY       555
                                     ENDIF                                      RMSY       556
7177      0000                       DATA   0                                   RMSY       557
                                                                                RMSY       558
                                                                                RMSY       559
                         *           LOCATIONS FOR LINKAGE .                    RMSY       560
                                                                                RMSY       561
                                                                                RMSY       562
7200      7700                       PARAMS Y                                   RMSY       563
                                                                                RMSY       564
                                                                                ZOS053     472
                                                                                ZOS053     473
                    41    YPATCH     EQU    DRIVEND-YNSACRG                     ZOS053     474
                                     IFLT   YPATCH,0,1                          ZOS053     475
                                     ERR    3SY IS TOO BIG                      ZOS053     476
                                                                                ZOS053     477
```

10-97

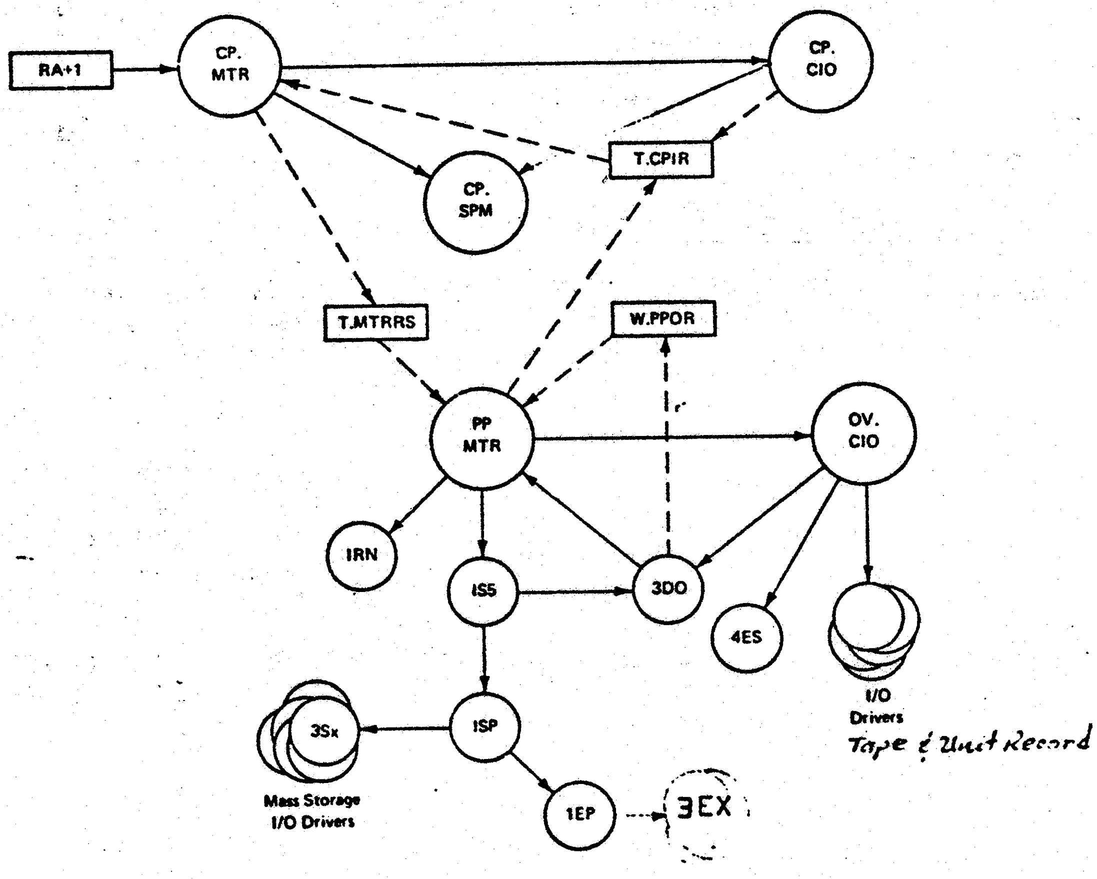
SCOPE
3.4

INPUT/OUTPUT
SOFTWARE SUBSYSTEMS

INTRODUCTION

Input and output request processing depends upon the source of each request. Active user CP programs issue RA+1 requests for I/O which are cycled through CPMTR. PP programs request I/O by placing a monitor request into their PP output register. System programs, which run at control point N.CP+1 cannot make monitor requests through RA+1. Since they run as CP service functions for PP programs, they make such requests through the output register of the PP servicing the program.

CPMTR assigns the I/O request to CP.CIO for I/O buffered requests or has MTR assign a PPU for CIO (circular input/output), who processes requests for magnetic tape, teletype, and unit record I/O. When actual disk I/O is required CIO/CP.CIO cause a copy of 1SP/1EP to be loaded to actually access the disk.



Another I/O processor, JANUS, exists in SCOPE, but its function is limited to processing unit record I/O for the system input and output queues. The queues contain job input and output files and are related to the job processing activities of SCOPE.

CIO

The circular input/output processor consists of the central memory program CP.CIO, the PP program OV.CIO and several PP I/O drivers. A system programmer can write his own input/output software, or he can have his program generate a call to CIO. Before calling CIO, the program must set up circular buffer parameters and the CIO operation code in the file environment table {FET} for the file. The relative address of the FET is placed in the CIO call.

A PP routine places a CIO call in its PP output register; PPMTR passes it through the CP input register for the CP.MTR. A CP program places a CIO call in the CP request register {RA+1}. When PPMTR accepts the CIO call, it assigns a PP and clears byte 0 of the PP output register.

When CP.MTR detects a CIO call, it passes it to CP.CIO if the request is for a buffered file or to CIO, for validation and selection of the proper routine to supervise execution of the function. The CIO is then reissued via the request stack and CP.MTR to be processed by the required PPCIO driver; byte zero of the RA+1 register is cleared. When the I/O operation is completed CP.CIO adds one to the code/status field of FET word one. As all CIO codes placed in the FET code/status field are even numbers, an odd number in that field signals completion of the operation {or that the file is not busy}.

SCOPE CIO CODES {3.4}

All codes indicated by * are illegal; all reserved codes are illegal. All codes are octal for coded mode operations; add 2 for binary mode. Example: 010 is coded READ, 012 is binary READ.

000	RPHR	054	*	130	CLOSE,NR
004	WPHR	060	UNLOAD	134	*
010	READ	064	*	140	OPEN
014	WRITE	070	RETURN	144	OPEN,WRITE
020	READSKP	074	*	150	CLOSE
024	WRITER	100	OPEN,NR	154	*
030	*	104	OPEN,WRITE NR	160	OPEN
034	WRITEF	110	POSMF	164	*
040	BKSP	114	EVICT	170	CLOSE,UNLOAD
044	BKSPRU	120	OPEN,NR	174	CLOSE,RETURN
050	REWIND	124	*		

200 Series for Special Read or Write (reverse, skip, non-stop, rewrite, etc.)

200 READC
204 WRITEC
210 READLS
214 REWRITE
220 *
224 REWRITER
230 *
234 REWRITEF
240 SKIPF
244 *
250 READNS
254 *
260 READN
264 WRITEN
270 *
274 *

300 Series for Tape OPEN and CLOSE

300 OPEN,NR
304 *
310 *
314 *
320 *
324 *
330 CLOSER
334 *
340 OPEN
350 CLOSER
354 *
360 *
364 *
370 CLOSER,UNLOAD
374 *

400 Series (Reserved for CDC)

500 Series (Reserved for Installations)

600 Series

600 *
604 *
610 *
614 *
620 *
624 *
630 *
634 *
640 SKIPB
644 *
650 *
654 *
660 *
664 *
670 *
674 *

7000 Series (Reserved for CDC)

CIRCULAR BUFFER

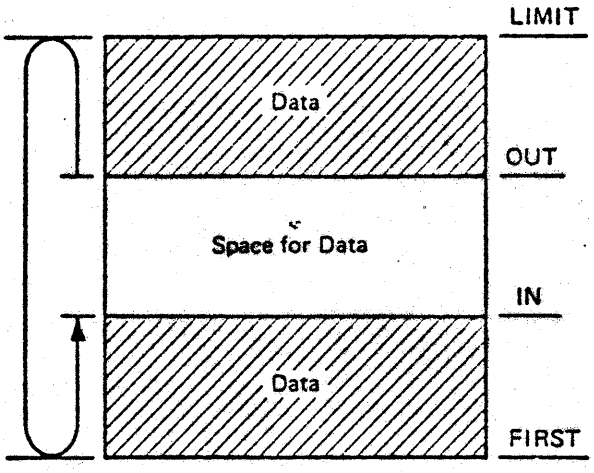
A circular buffer is a temporary storage area in central memory through which data passes during I/O operations. It is termed circular because I/O processing routines treat the last word and the first word of the buffer area as contiguous.

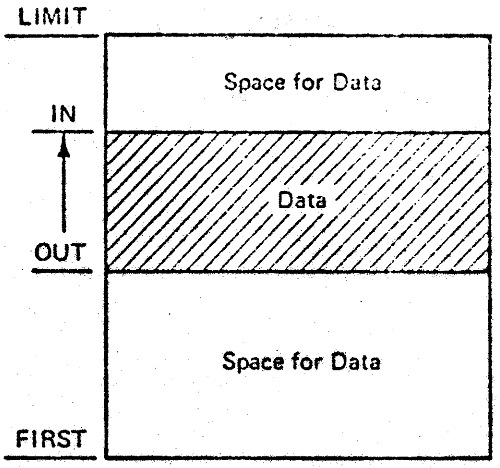
FIRST is the first word address of the circular buffer. Routines that process I/O never change the value of FIRST.

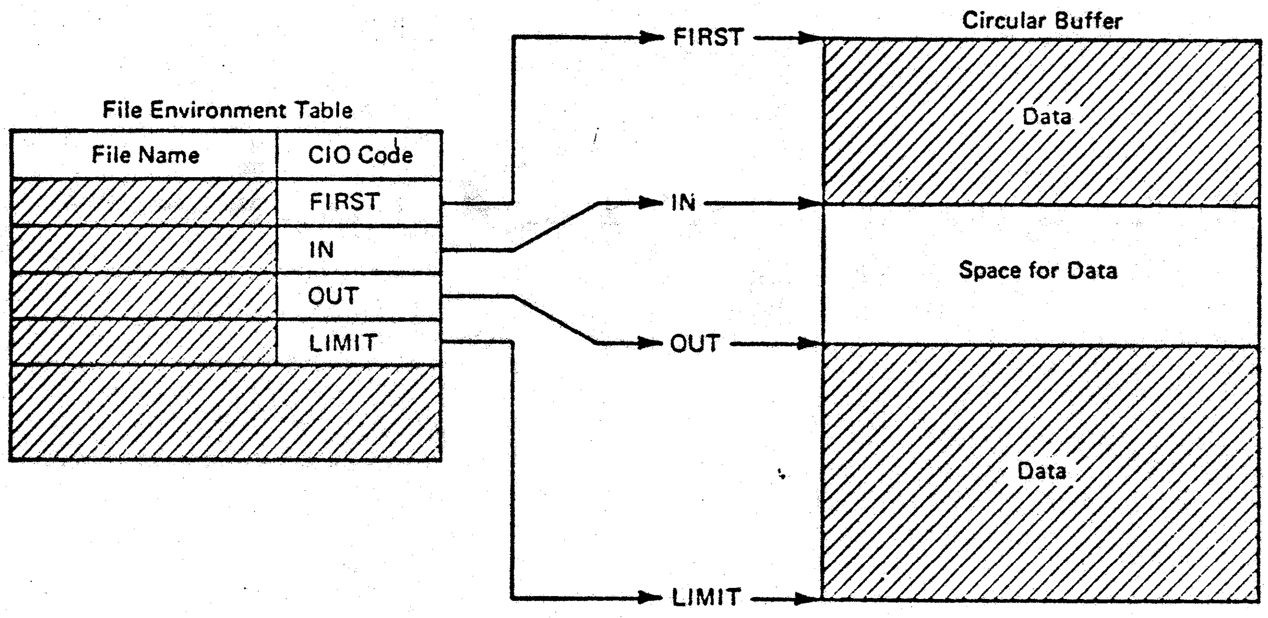
LIMIT is the last word address +1 of the buffer area. No data is stored in this word. When LIMIT is reached, the next address accessed is FIRST. Routines that process I/O never change the value of LIMIT.

OUT is the next location from which data is removed from the circular buffer. CIO or the calling program changes OUT depending on whether the operation is read or write.

IN is the next location into which data is written. CIO or the calling program changes IN depending on whether the operation is read or write. When IN=OUT-1, the buffer is full. A partly filled buffer extends from OUT to IN-1.







The circular buffer must be at least one word larger than the length of one PRU. For a write operation at least one PRU of data should be in the buffer. For a read operation the buffer must have room to receive one PRU of data. Less than one PRU may be transmitted only if an end-of-record is read or written.

CIO OPERATION

When MTR initiates CIO, either version, to perform file I/O, CIO locates the FNT for the file. If the FNT pointer in the FET is non-zero, CIO checks the FNT entry indicated by the pointer to determine if the file name in the FNT entry is the same as the file name in the FET; it will also check that the file is assigned to the job control point. If the names do not match or if the FNT pointer is zero, CIO will search the entire FNT for a file assigned to that job control point with a matching name. If the file is not found, CP.CIO will create a FNT entry for the file. Such files are always local and assigned to allocatable devices. Once the FNT entry is found or created, CIO stores the address of the FNT entry in the FET. The FNT pointer in the FET facilitates the FNT search.

If file status is busy, CIO posts the request for rescheduling and exits. Otherwise, CIO checks the code field in the FET against the last code/status field in the FNT to ensure the requested operation can legally follow the preceding operation. If not, CIO replaces the RA+1 call with a request for the PP program CEM which handles error messages, then reissues the RA+1 call to be processed again by CP.MTR. If the operation is legal, CIO transfers the code/status field in the FET to the last code/status field in the FNT. The proper CIO routine is selected to supervise function execution.

When the file is opened, CIO determines if the file is on an allocatable or non-allocatable device or is ECS resident by checking the device code in the second word of the FNT. If the file is on an allocatable device, CIO puts the request in the I/O request stack in CMR. The stack processor CP.SPM schedules I/O on allocatable devices; it will perform the I/O and set the completion bit. PP.CIO and its overlays process I/O requests for non-allocatable.

When PP.CIO is required, PPMTR assigns an available PP and causes CIO to be loaded and initialized. Depending upon the operation, CIO will call one or more of the following overlays.

Function Routines:

1CL	File Close
1OP	File Open
1MF	Multifile Positioning
1RP	Reel Close
3DO	Mass Storage Device File Open
4ES	Enter Stack Request {mass storage I/O}
6WM	Write Error Message

Tape Drivers:

1RS	Read 7-track stranger {S} tapes
1RT	Read 7-track SCOPE standard labeled tapes
1MT	Read/write L {Long Stranger} tapes {7-track}
1NR	Read 9-track stranger {S} tapes
1NW	Write 9-track stranger {S} tapes
1WS	Write 7-track stranger {S} tapes
1WI	Write 7-track SCOPE standard labeled tapes
1TF	Move tape forward {except long record {L} tapes}
2TB	Move tape backward {except long record {L} tapes}
1R9	Read 9-track SCOPE standard labeled tapes
1W9	Write 9-track SCOPE standard labeled tapes

Unit Record Drivers:

2PC	On line card punch
2RC	On line card reader
2LP	On line printer

Tape Error Recovery Drivers:

1P1	Write error recovery - tape positioning
1P2	Write error recovery - erase/rewrite
1P3	Write error recovery - verification driver
1P4	Write error recovery - final driver
1RV	Initialize/terminate read error recovery
1R2	Read parity error recovery
1R3	Read error recovery - position/reread
1N0	Noise error recovery - read error processing
1N2	Noise error recovery - read recovery driver
1N3	Noise error recovery - skip noise record
1CS	Write CM data - 7/9 track stranger tape read recovery
1CT	Write CM data - 7/9 track SCOPE standard tape read recovery
1CR	Write CM data - 7-track other tape read recovery

If the file device code is for a non-allocatable device, PPCIO loads an I/O driver into its PP to perform the actual I/O. The overlay selected is determined by the operation requested. For example, if a user issues a request to read data from a file on a SCOPE standard format 7-track tape, CIO will call the overlay 1RT into its PP. 1RT will reserve one of the hardware channels connected to the equipment. It then issues the function codes to connect the controller and tape drive. 1RT issues functions to transmit one PRU of data from the tape driver over the data channel.

1RT accumulates the PRU of data in a PP buffer. When the entire PRU is transmitted or an end-of-record {short PRU} is encountered, 1RT picks up the pointers to the circular buffer in central memroy from the FET. 1RT continues to transfer PRUs of data from the tape through the PP buffer to the circular buffer until the buffer is full or an end-of-record is encountered. 1RT updates the PRU count in the file FNT, releases the channel, sets completion bits in the FNT and FET, and drops out.

The following charts depict the logical sequence of events during various CIO tape operations.

READ	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
1. Exit if not enough room in buffer for one maximum size physical record.	x	x				
2. Exit if not enough room in buffer for MLRS words.			x	x	x	x
3. Read one physical record into PP.	x	x	x	x		
4. Read one physical record into CM.					x	x
5. If physical record exceeds maximum allowable, return error status DEVICE CAPACITY EXCEEDED and perform error procedures.	x	x				
6. If physical record exceeds maximum logical record size, return error status DEVICE CAPACITY EXCEEDED and perform error procedures. If a long record is encountered, excess information is discarded without notification to user.			x	x	x	x
7. If end-of-file mark was read, perform end-of-file mark procedures.	x	x	x	x	x	x
8. If noise records encountered, go to 3.	x	x	x	x	x	x
9. If parity error, perform parity procedures.	x	x	x	x	x	x
10. If end-of-tape reflective spot was encountered and tape is unlabeled, perform end-of-reel procedures.			x	x	x	x
11. If short PRU was read, strip level number.	x	x				
12. If zero length PRU was read, go to 21.	x	x				
13. When 6681 is present, convert data in PP from BCD to display code.		x		x		
14. When 6681 is present, convert data in CM from External BCD to display code.						x

READ {continued}	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
15. Convert 1632 line terminator to 0000.		x				
16. Transmit data to CM.	x	x	x	x		
17. Update IN.	x	x	x	x	x	x
18. Fetch OUT from CM.	x	x				
19. Place in word 7 of FET the number of unused bits in the last data word.			x	x	x	x
20. If full PRU, go to 1.	x	x				
21. If last record was level 17 of tape mark, set end-of-file status.	x	x	x	x	x	x
22. Set end of record in status field of FET and exit.	x	x	x	x	x	x

WRITE	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
1. Exit if not full PRU.	x	x				
2. If data from OUT to IN exceeds maximum logical record size from FET, return DEVICE CAPACITY EXCEEDED and perform error procedures.			x	x	x	x
3. Fetch number of unused bits in last data word from FET and adjust record length. If record length constitutes a noise record, return DEVICE CAPACITY EXCEEDED and perform error procedures.			x	x	x	x
4. Read one PRU of data starting at OUT from CM to PP.	x	x				
5. Read data contained between OUT and IN from CM to PP. Adjust by unused bit count.			x	x		
6. When 6681 present, convert display code to BCD in PP memory.		x		x		
7. When 6681 present, convert from display code to BCD in CM.						x
8. Convert zero byte line terminator to 1632.		x				
9. Write record to tape.	x	x	x	x		
10. Write from CM to tape, data contained between OUT and IN, adjusted by unused bit count.					x	x
11. When 6681 present, convert data in CM buffer back to display code.						x
12. If parity error, perform parity procedure.	x	x	x	x	x	x
13. If end-of-tape reflective spot, perform end-of-reel procedures.	x	x	x	x	x	x
14. Update OUT.	x	x	x	x	x	x
15. Exit.			x	x	x	x
16. Fetch IN from CM	x	x				
17. Go to 1.	x	x				

WRITER	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
1. If IN = OUT, exit.			×	×	×	×
2. If PRU not full, insert level number in PP buffer.	×	×				
3. If data from OUT to IN exceeds maximum logical record size from FET, return DEVICE CAPACITY EXCEEDED and perform error procedures.			×	×	×	×
4. Fetch number of unused bits in last data word from FET and adjust record length. If record length constitutes a noise record, return DEVICE CAPACITY EXCEEDED and perform error procedures.			×	×	×	×
5. Read one PRU starting at OUT or between OUT and IN, whichever is smaller, from CM to PP.	×	×				
6. Read data between OUT and IN from CM to PP. Adjust by unused bit count.			×	×		
7. When 6681 is present, convert display code to BCD in PP memory.		×		×		
8. When 6681 is present, convert display code to BCD in CM.						×
9. Convert zero byte line terminator to 1632.			×			
10. If IN = OUT, write zero length record. Go to 12.	×	×				
11. Write record to tape.	×	×	×	×		
12. Write data between OUT and IN from CM to tape, adjust by unused bit count.						
13. When 6681 is present, convert data in CM buffer to display code.						×
14. If parity error, perform parity procedures.	×	×	×	×	×	×
15. If end-of-tape reflective spot, perform end-of-reel procedures.	×	×	×	×	×	×
16. Update OUT.	×	×	×	×	×	×

WRITER {continued}	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
17. Exit.			x	x	x	x
18. If full PRU is not written, exit.	x	x				
19. Go to 1.	x	x				

WRITEF	Standard Binary.	Standard Coded	S Binary	S Coded	L Binary	L Coded
1. If no data from OUT to IN, go to 23.	x	x				
2. If no data from OUT to IN, go to 19.			x	x	x	x
3. If not full PRU, insert 0 level number.	x	x				
4. If data from OUT to IN exceeds maximum logical record size, return DEVICE CAPACITY EXCEEDED and perform error procedures.			x	x	x	x
5. Fetch number of unushed bits in last data word from FET and adjust record length. If record length constitutes a noise record, return DEVICE CAPACITY EXCEEDED and perform error procedures.			x	x	x	x
6. Fetch one PRU of data starting at OUT or data between OUT and IN, whichever is smaller, from CM to PP.	x	x				
7. Read data contained between OUT and IN from CM to PP. Adjust by unused bit count.			x	x		
8. When 6681 is present, convert display code to BCD in PP memory.		x		x		
9. When 6681 is present, convert display code to BCD in CM.						x
10. Convert zero byte line terminator to 1632.		x				
11. Write record to tape.	x	x	x	x		
12. Write data between OUT and IN from CM to tape, adjust by unused bit count.					x	x
13. When 6681 is present, convert data in CM buffer to display code.						x
14. If parity error, perform parity procedures.	x	x	x	x	x	x
15. If end-of-tape reflective spot, perform end-of-reel procedures.	x	x	x	x	x	x
16. Update OUT.	x	x	x	x	x	x

WRITE (continued)	Standard Binary	Standard Coded	S Binary	S Coded	L Binary	L Coded
17. Write end-of-file mark and exit.			x	x	x	x
18. If full PRU is not written, write zero length level 17 record and exit.	x	x				
19. Go to 3.	x	x				
20. If last operation was WRITE, write zero length PRU.	x	x				
21. Go to 17.	x	x				

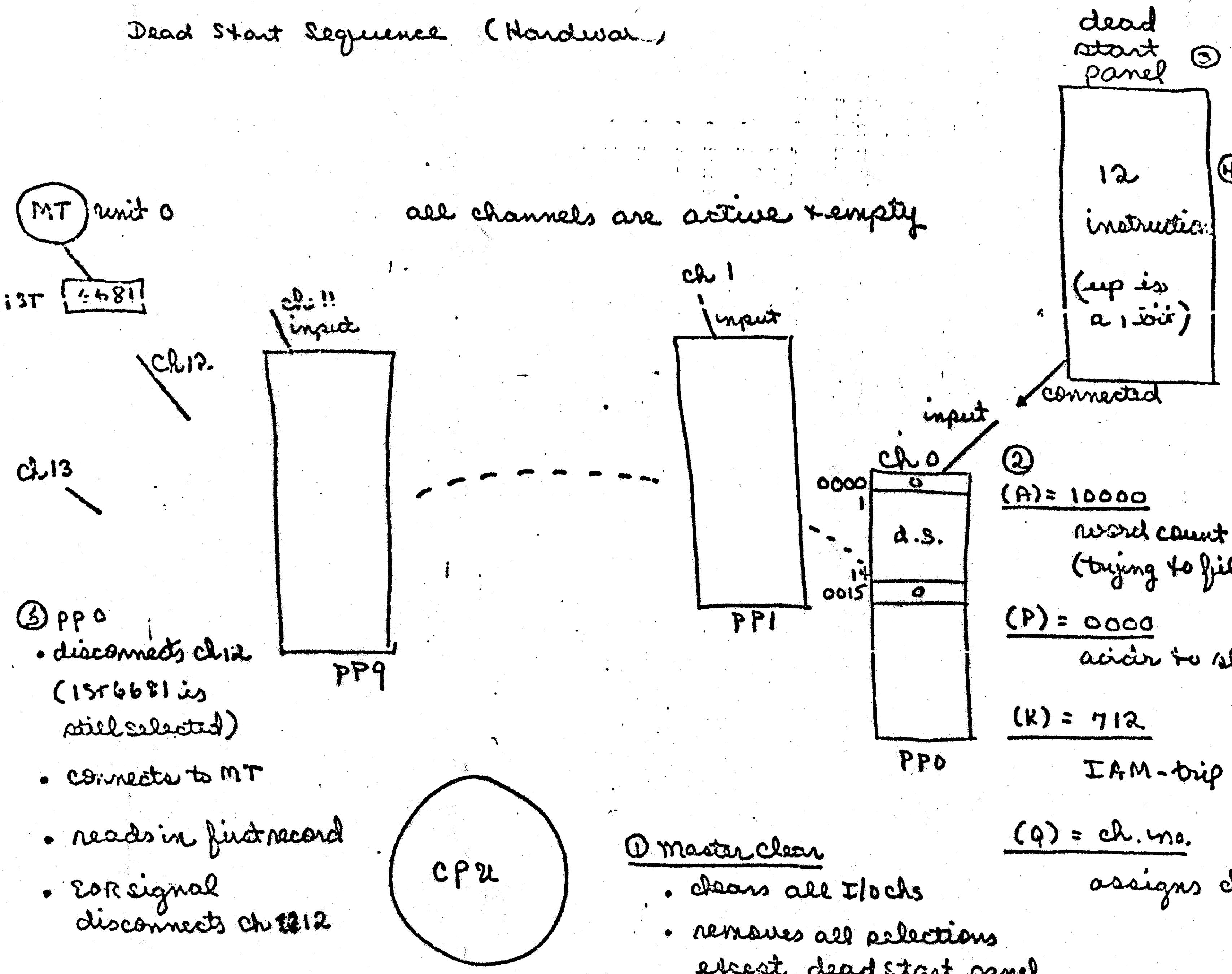
SECTION ELEVEN

DEADSTART
HARDWARE SEQUENCE

SECTION ELEVEN - DEADSTART

CONTENTS

Dead Start Sequence (Hardwar)



① Master clear
- clears all I/O chs
- removes all selections except dead start panel
- sets all channels active + empty
- selects all 6681's in system

②
(A) = 10000
word count to inp (trying to fill pp)

(P) = 0000
addr to store in

(K) = 712
IAM - trip 4^{2}

(Q) = ch. no.
assigns ch to pp

③ send to PP0

④ disconnect ch 0 (causes execution at 0001 in PP0)

⑤ PP0
- disconnects ch 12 (13T 6681 is still selected)
- connects to MT
- reads in first record
- EOR signal disconnects ch 12

[illegible] the Dead Start from the Customer Engineers'
6000 [illegible] PP Training Manual

221
no

DEAD START

Dead Start is a system used to initially start the computer, dump the contents of the peripheral and control processor memories to a printer or other output device, or sweep memory without executing instructions.

The dead start panel contains a 12 x 12 matrix of toggle switches, a Sweep-Load-Dump switch and a Dead Start switch. It also contains memory margin switches which are used for maintenance checks.

LOAD

To initially load programs and data into the 6400, the Sweep-Load-Dump switch is set to Load. The matrix of toggle switches is set to a 12-word program (up position equals one, down position equals zero). When the Dead Start switch is turned on, a 1 μsec dead start pulse initiates the following actions.

- Assigns to each peripheral and control processor the corresponding I/O channel
- Sets all I/O channels to active and empty
- Sets K for all processors to 712 (input).
- Sends a MC on all I/O channels
- Sets P for all processors to zero
- Sets A for all processors to a word count of 10,000

The dead start pulse is repeated every 4096 μsec while the Dead Start (DS) switch is on. To start the 6400, the DS switch is normally turned on momentarily, then turned off. Recycling of the dead start pulse is controlled by the Real-Time clock. The pulse is formed by ANDing DS switch On with the 12 bits of the Real-Time clock.

When the dead start synchronizer on channel 0 receives the MC sent by dead start, it sends a full pulse but no data. When processor 0 receives the full, it stores the content of the channel 0 Input register (all zeros) in location 0000 and sends an empty pulse to the dead start synchronizer. The dead start synchronizer then acts like an input device and sends 12 words from the switch matrix (as 12-bit words), and processor 0 stores them in locations 0001 through 0014_8. After the last word, the dead start synchronizer sends a Disconnect signal which causes processor 0 to exit from the 712 instruction and store zeros in address 0015. Processor 0 reads location 0000, adds one to its content and goes to 0001 for its next instruction. It then executes the 12-word (or less) program, which normally is a control program, which is used to load information and begin operation. The other processors are still set to 712 and may receive data from processor 0 via their assigned I/O channels using interchannel communications.

SWEEP

If the DS switch is operated with the Sweep-Load-Dump switch set to Sweep, all processors are set to a 505 instruction and P registers are set to 0000. Since the 50 instruction does not require five trips around the barrel, there is no logic to clear or advance K from 505. The 50X translation of K causes all processors to sweep through their memories, reading and restoring without executing instructions. This is a maintenance routine and can be used to check the operation of memory logic.

DUMP

The dead start pulse, with the Sweep-Load-Dump switch set to Dump, initiates the following actions.

- Sets all processors to 732
- Sends MC on all channels except channel 0
- Holds channel 0 active and empty
- Assigns each processor to its corresponding I/O channel
- Sets all A and P registers to zero

All processors sense the empty and active conditions of their assigned channels, output the content of their address 0000, set their I/O channels to full, and wait for an empty condition. All processors advance P by one and reduce A by one ($A = 7776_8$). Channel 0, which is assigned to processor 0, is held to empty by the Dump switch. Therefore, processor 0 cycles through the 732 instruction until A=1 and then goes to memory location 0001 for its next instruction. Thus processor 0 can send its entire memory contents on channel 0, although no I/O device has been selected to receive it. Processor 0 is then free to execute a dump program which must have been previously stored in memory 0 (beginning at location 0001).

11-2

Test your knowledge of I/0 !

Decode the deadstart programs

① if the deadstart tape is on channel 12 or 13

② if the tape is on channel 1-11

DEADSTARTING

Deadstart under SCOPE 3.3 is from tape; deadstart cards are not used. The system tape includes a series of programs which constitute the deadstart package.

The deadstart panel setting shown below is for a deadstart tape on channels 0, 12, or 13. The system tape is read directly by the panel; no cards are necessary. If the tape channel is 0, the only change is that word 1 is all zeros; because channel 0 is disconnected by the deadstart synchronizer and cannot be disconnected again.

The display channel number must be 10B. The synchronizer number must be 7. Otherwise, the common deck DSLCOM must be changed and CONTROL (CED) must be reassembled.

If the tape channel is 12B or 13B:

Word	Binary	Octal	Description
0001	111 101 00t ttt	75TT	Disconnect tape channel
0002	111 111 00t ttt	77TT	Select tape unit
0003	eee ccc 00u uuu	ECUU	
0004	111 111 00t ttt	77TT	Rewind tape
0005	000 000 001 000	0010	
0006	111 111 00t ttt	77TT	Select input to end-of-record
0007	001 100 000 s00	140S	
0010	111 100 00t ttt	74TT	Activate tape channel
0011	111 001 00t ttt	71TT	Input tape record
0012	000 000 001 011	0013	to location 13

Remainder of the panel is irrelevant

tttt = Tape channel

eee = Equipment number of tape controller

uuuu = Tape unit

ccc = CMR number (000 for first CMR)

s = PPO save switch (if zero, save PPO, nonzero, do not)

11-4

Installation Handbook

If the system tape is on channels 1-11, use the following setting:

Word	Binary	Octal	Description
0001	111 011 00t ttt	73TT	Free tape channel
0002	000 000 001 011	0013	
0003	111 101 00t ttt	75TT	Disconnect tape channel
0004	111 111 00t ttt	77TT	Select tape unit
0005	eee ccc 00u uuu	ECUU	
0006	111 111 00t ttt	77TT	Select input to end of record
0007	001 100 000 s00	140S	
0010	111 100 00t ttt	74TT	Activate channel
0011	111 001 00t ttt	71TT	Input to location 13
0012	000 000 001 011	0013	
0013	000 000 000 000	0000	Loop on input
0014	111 001 001 010	7112	from channel 12

tttt = Tape channel number

eee = Tape controller equipment number

uuuu = Tape unit number

ccc = CMR number (000 for first CMR)

s = PP0 save switch (if zero, save PP0 -- nonzero, do not)

When an installation prepares a system tape, it may place up to eight central memory resident configurations in the deadstart section. The CMR that will be loaded from the tape during a given deadstart is determined by the setting of bits 6, 7, 8 of word 3 on the Dead Start Panel. The nth CMR is selected by setting the value to n-1. Only the contents of CMR (primarily the EST) is varied; the CM directory and the resident library remain the same no matter which CMR is used.

The operator can select particular functions through the display console. Permanent messages, informing the operator of current activity, appear on the right screen; conversational messages, allowing passage of parameters, appear on the left screen.

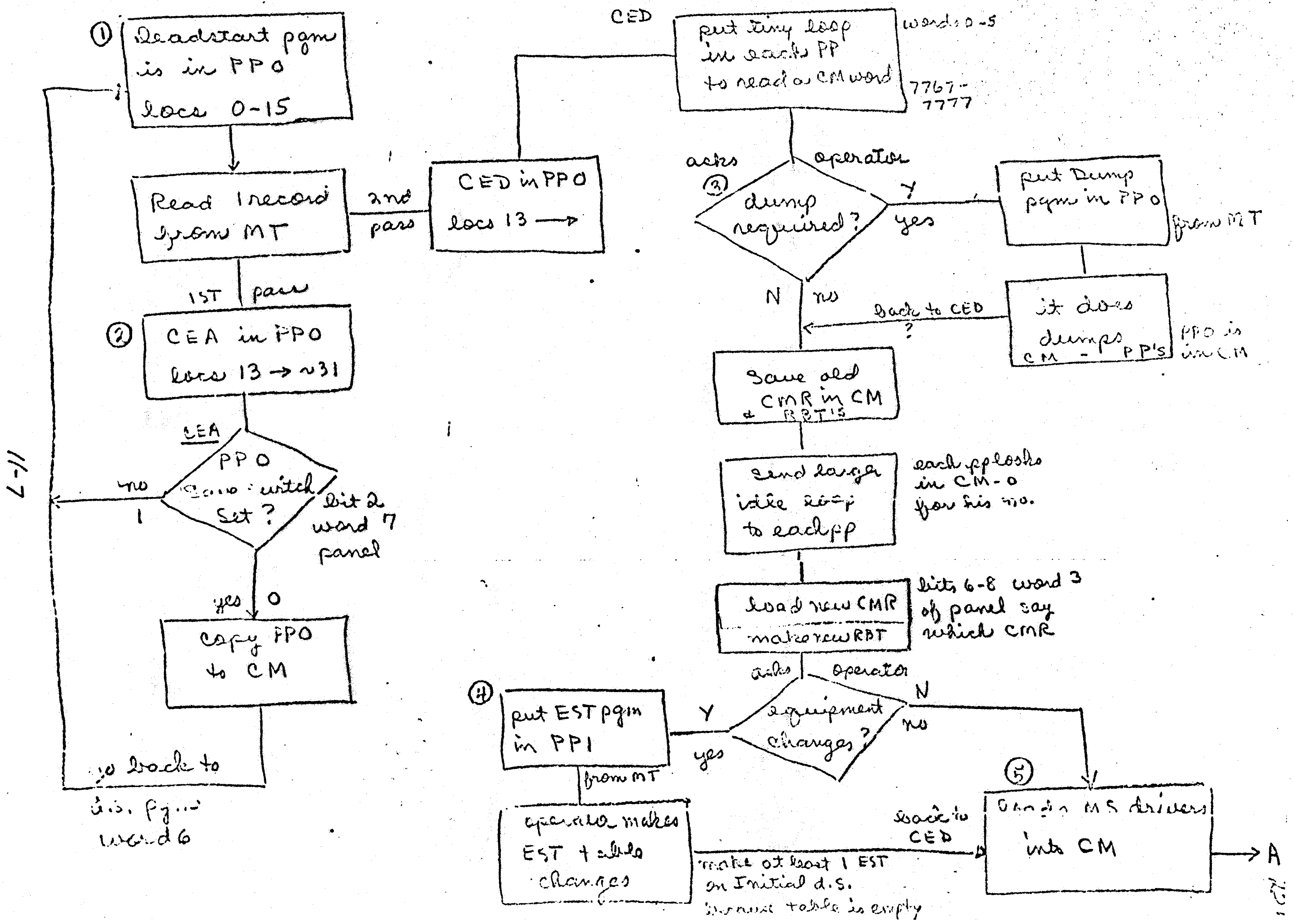
11-5

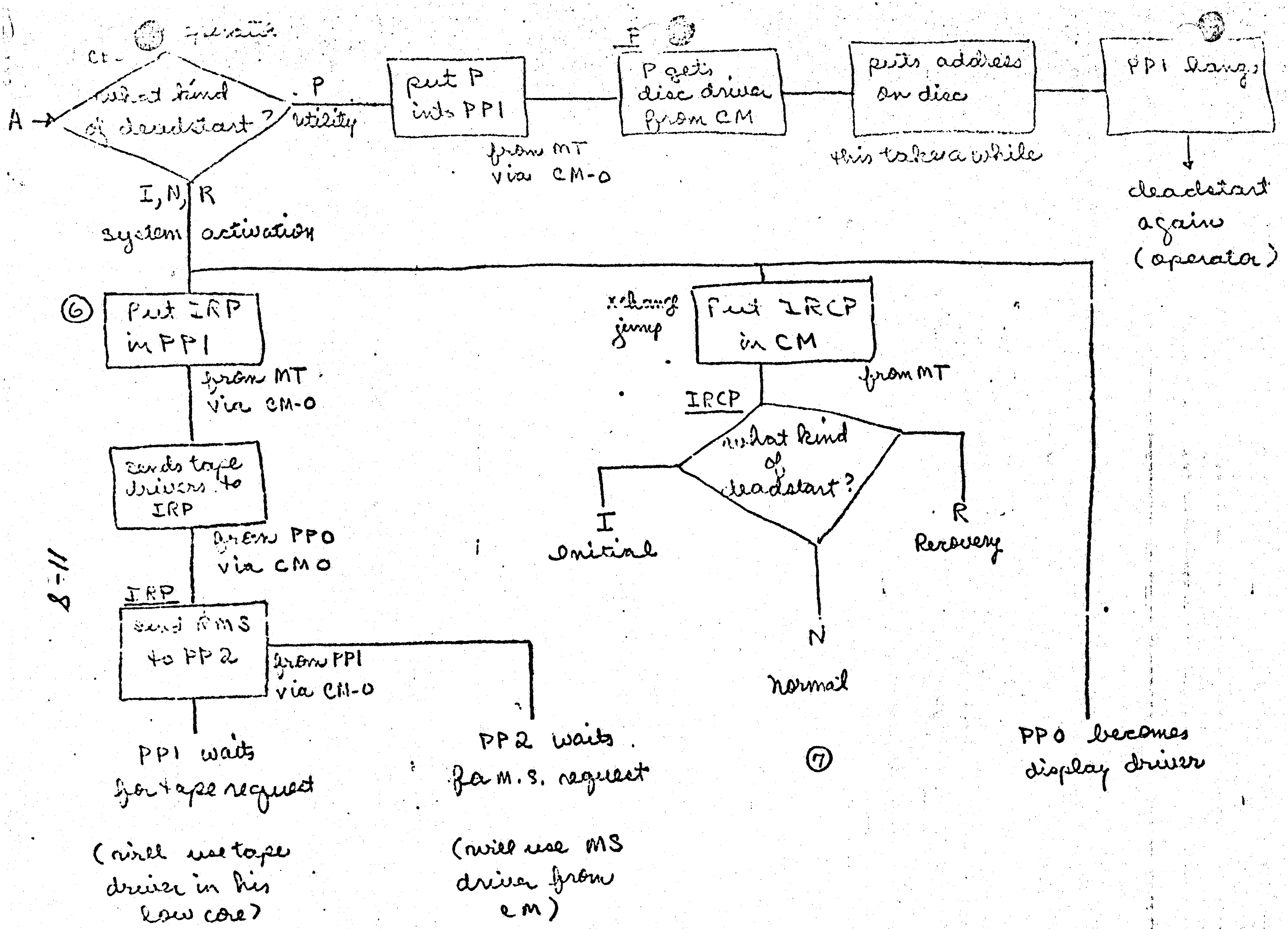
Installation Handbook

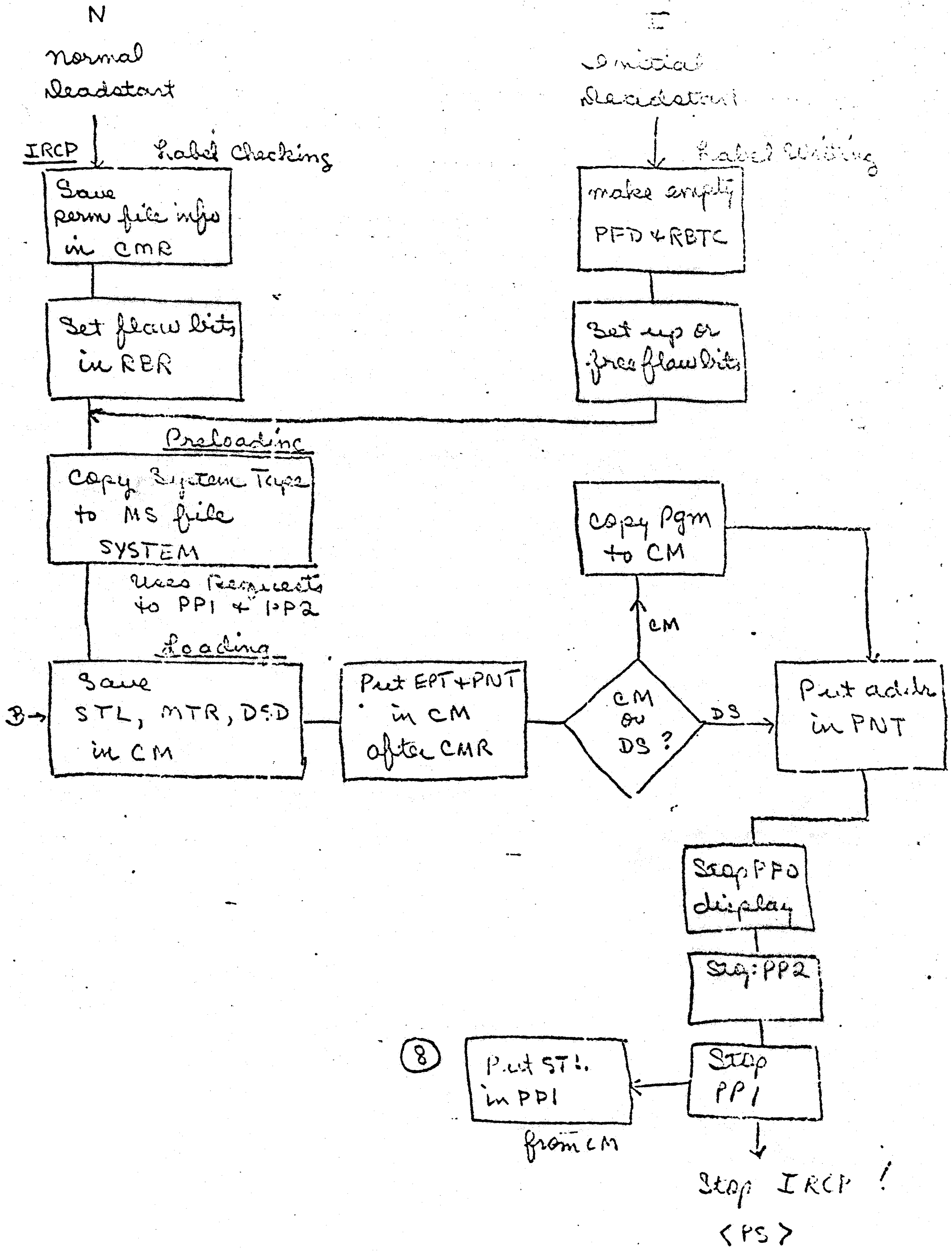
DEADSTART SOFTWARE

v. 3.2

Start Sequence (Software)

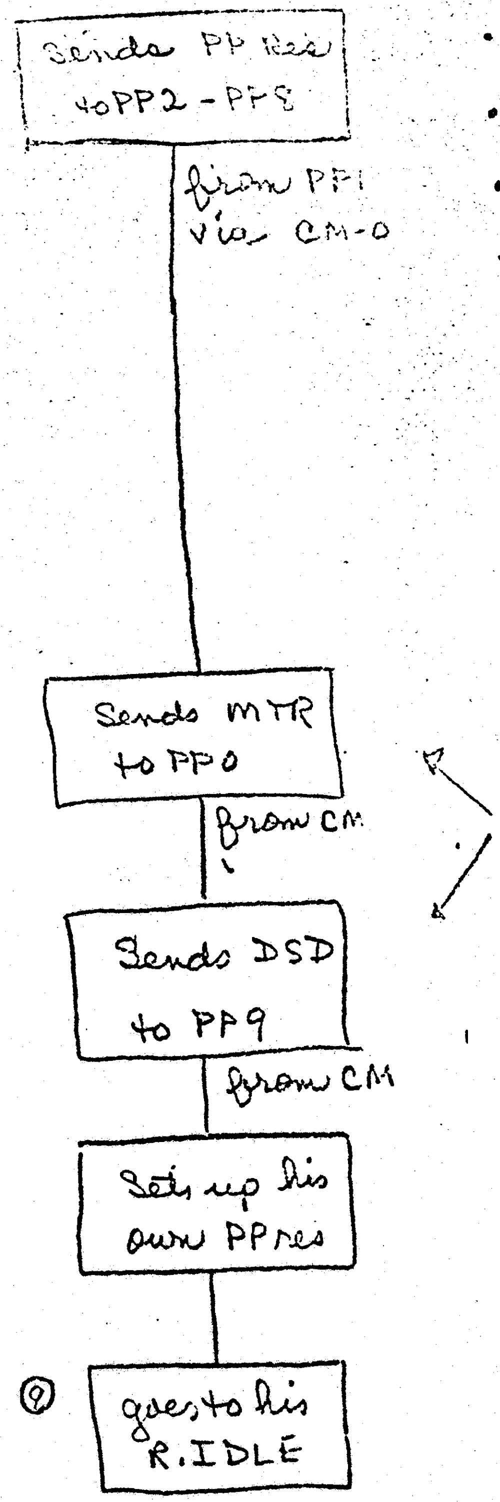



A
what kind of deadstart?
P
utility
put P into PP1
from MT via CM-0
P
P gets disc driver from CM
puts address on disc
this takes a while
PP1 hangs
deadstart again (operator)
I, N, R
system activation
⑥
Put IRP in PP1
from MT via CM-0
sends tape drivers to IRP
from PP0 via CM0
IRP
send RMS to PP2
from PP1 via CM-0
PP1 waits for tape request
(will use tape driver in his low core)
PP2 waits for M.S. request
(will use MS driver from CM)
xchange jump
Put IRCP in CM
from MT
IRCP
what kind of deadstart?
I
Initial
N
normal
R
Recovery
⑦
PP0 becomes display driver
8-11


N
normal
Deadstart
I
Initial
Deadstart
IRCP
Label Checking
Label Writing
Save
perm file info
in CMR
make empty
PFD + RBTC
Set flaw bits
in RBR
Set up or
free flaw bits
Preloading
Copy System Tape
to MS file
SYSTEM
Uses Requests
to PP1 + PP2
copy Pgm
to CM
CM
Loading
B
Save
STL, MTR, DSD
in CM
Put EPT+PNT
in CM
after CMR
CM
or
DS ?
DS
Put addr
in PNT
Stop PPO
display
Stop PP2
⑧
Put STL
in PP1
from CM
Stop
PP1
Stop IRCP !
<PS>

254

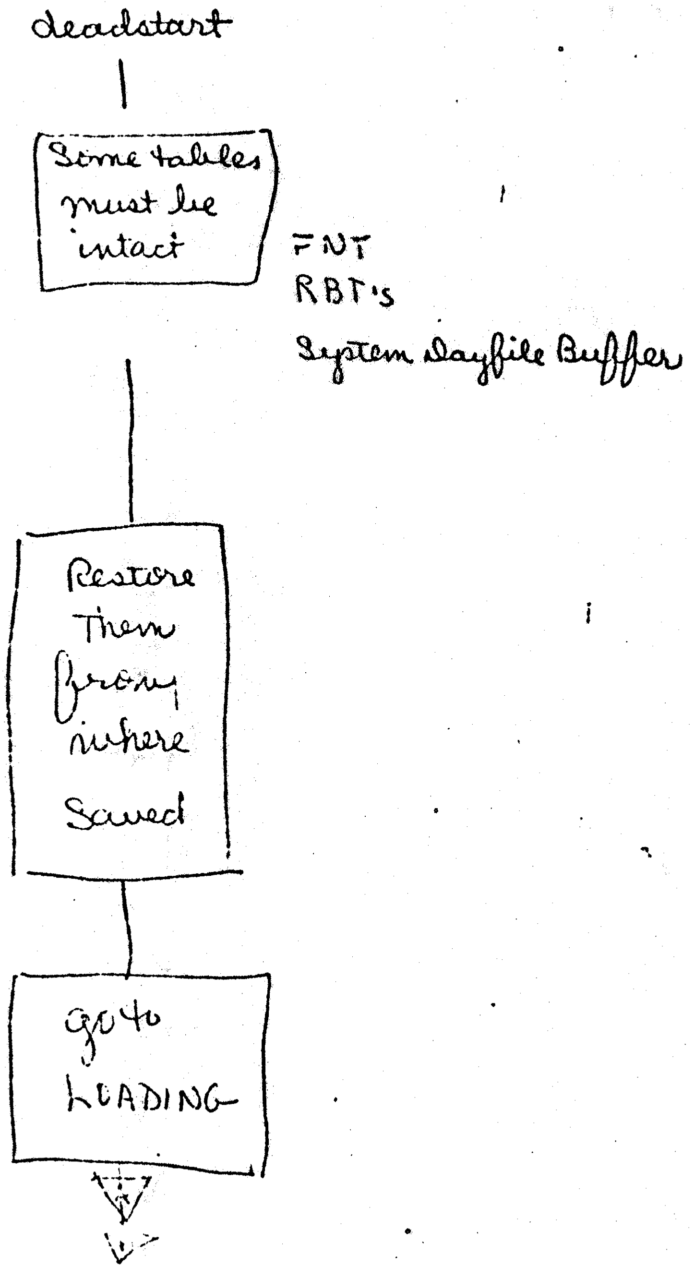
STL



- fixes D.PPIR, D.PP0R, D.PPMESI
- puts PP no. in CM-0
- PP sees his no.
- Sends R.IDLE-1 addr to PP loc 0
- Sends 777 pp words (PP res 1-777)
- disconnect ch 0 (pp goes to R.IDLE)

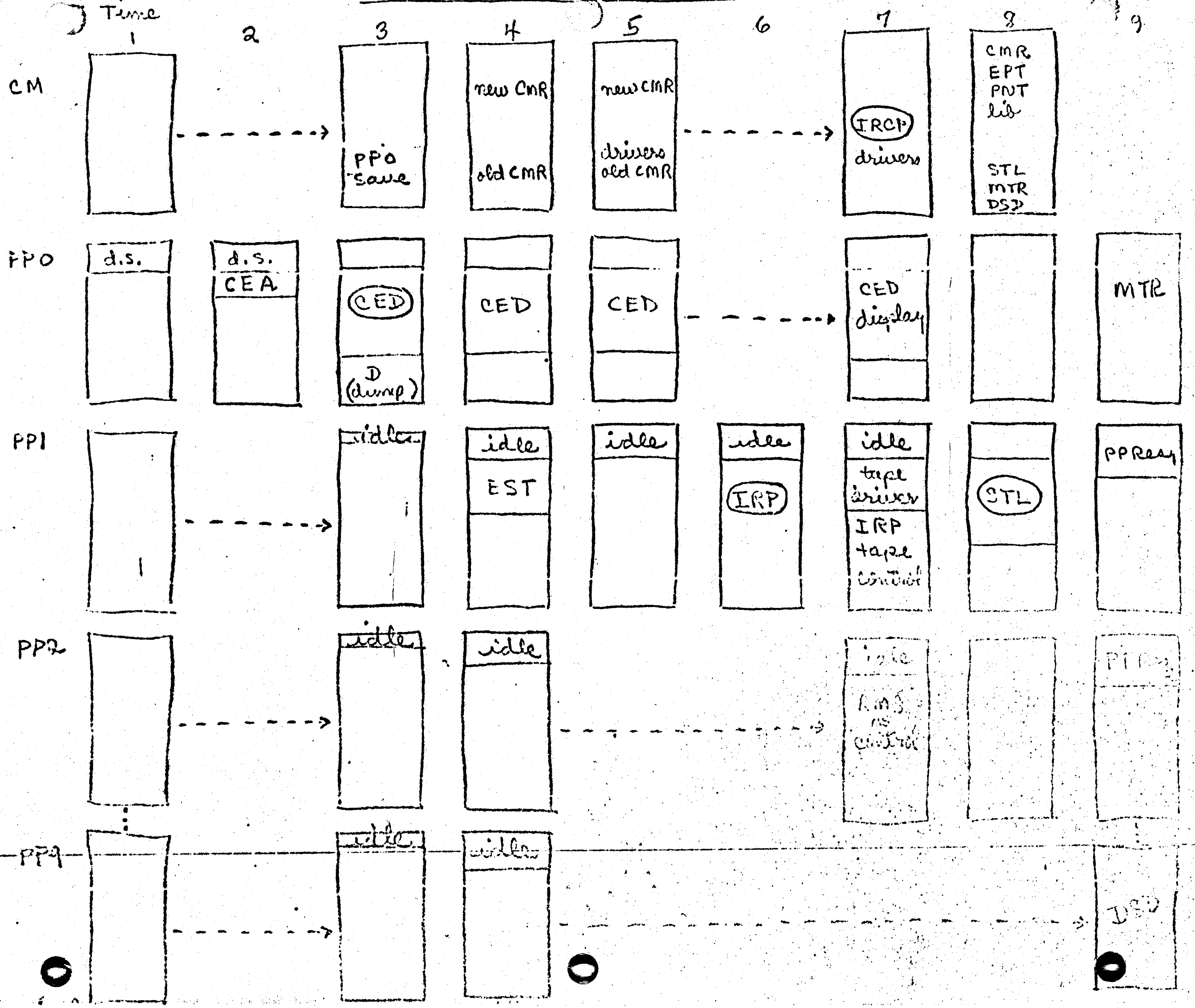
- Sends small pgm via CM-0
- .. which reads CM...

RECOVERY



11-11

Core Contents at Deadstart Times



SECTION TWELVE

RELOCATABLE OVERLAYS

SECTION TWELVE - RELOCATABLE OVERLAY

CONTENTS

RELOCATABLE OVERLAY CONCEPTS

more like IDENT

```
                              IDENT     PGM, C.PPFWA      } code of
                                {                         } transient
1000=7777?          TSYM      DATA      -0                } program
                                {
                    ABC       SEGMENT   77B
                              ORG       2000B              ORG anywhere
                              LOC       0
0000=0014                     VFD       12/RTAB ———— {distance to
                                                     {relocation
0001=0100  0000     XENTRY    LJM       0                      table
                    ENTRY     EQU       *-1          } entry
                                                     } and exit
0003=5000  0011               LDM       LOC          }
0005=5000  1000               LDM       TSYM         } instructions
0007=0100  0013  12           LJM       *+3          }   (do
0011                LOC       BSS       1            }   nothing)
0012=0100  0001               LJM       XENTRY       } exit
                    RTAB      BSS       0
                              VFD       12/0004B     }
                              VFD       12/0010B     } relocation
                              VFD       12/0013B     }   table
                              DATA      0            }
                                {

                    ABD       SEGMENT   77B
                                {

                              END
```

these must be relocated

next segment

Dick Tobin
cs   jb
2/72

```
PGM.C.PPFWA                                            COMPASS - VER 2.     02/10/72  11.57.39.      PAGE

                                      IDENT  PGM.C.PPFWA
                                      PERIPH
                                      SST
      1000                            ORG    C.PPFWA
                              * MACROS TO REDEFINE INSTRUCTIONS INVOLVING ADDRESSSES
                              LJM.    MACRO  M,D
                                      LOCAL  A
                                      VFD    6/01B,6/D           01 IS FUNCTION CODE FOR LJM
                              A       VFD    12/M
                                      RMT
                                      VFD    12/A
                                      RMT
                                      ENDM
                              LDM.    MACRO  M,D
                                      LOCAL  A
                                      VFD    6/50B,6/D           50B IS FUNCTION CODE FOR LDM
                              A       VFD    12/M
                                      RMT
                                      VFD    12/A
                                      RMT
                                      ENDM

                              * CODE FOR CALLING PROGRAM
      1000        7777        TSYM    DATA   -0
      1001                    CODE    BSS    100
      1145        0000        TSYM1   DATA   0

                              ABC     SEGMENT 77B
      2000                            ORG    2000B               CAN BE ANYWHERE
                 2000         ORG     SET    2000B               MUST AGREE WITH ORG STATEMENT
 L    0                               LOC    0

 L    0    0014                       VFD    12/RTAB             DISTANCE TO RTAB
 L
                                      LIST   G
 L    1    0100 0000          XENTRY  LJM    0
                         2    ENTRY   EQU    *-1                 ENTRY AND EXIT POINTS

 L    3    5000                       LDM.   LOC                 MUST BE RELOCATED
 L    4    0011            ↑↓000001           VFD    12/LOC                                      LDM.      .1
 L    5    5000 1000                  LDM    TSYM                LOAD A SYMBOL FROM TRANSIENT
 L    7    0100                       LJM.   *+3                 MUST BE RELOCATED
 L   10    0013            ↑↓000002           VFD    12/*+3                                      LJM.      .1
 L   11                       LOC     BSS    1
 L   12    0100                       LJM.   XENTRY              MUST BE RELOCATED
 L   13    0001            ↑↓000003           VFD    12/XENTRY                                   LJM.      .1
 L
 L   14                       RTAB    BSS    0                   BEGINNING OF RELOCATION TABLE
                                      HERE
 L   14    0004                       VFD    12/↑↓000001                                         *RMT*     .1
 L   15    0010                       VFD    12/↑↓000002                                         *RMT*     .1
 L   16    0013                       VFD    12/↑↓000003                                         *RMT*     .1
 L   17    0000                       DATA   0                   END OF RELOCATION TABLE

      20                              END
```

12-2

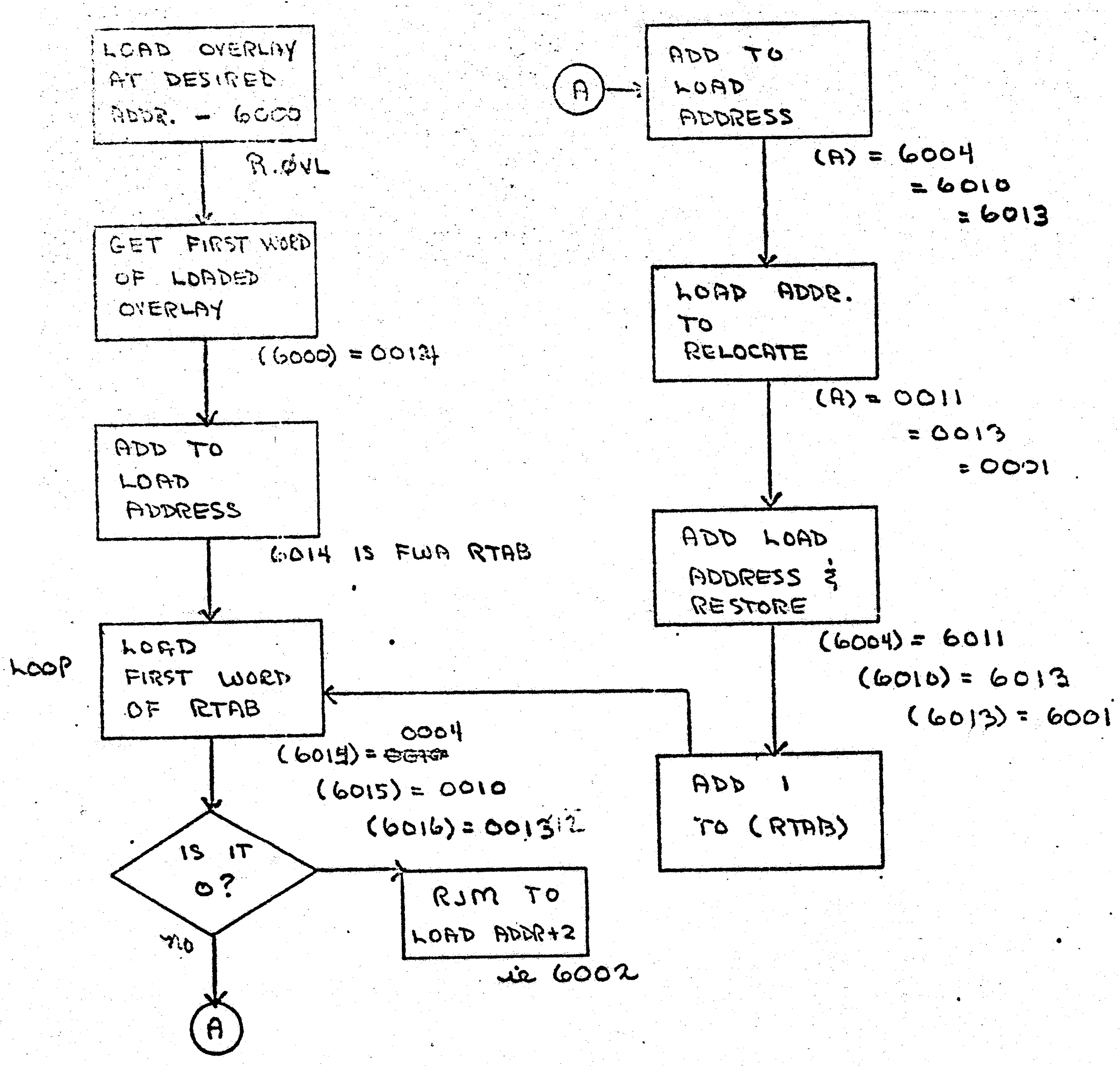
RELOCATABLE OVERLAY

EXAMPLE: LOAD THE OVERLAY AT 6000B

CONTENTS OF CORE:

```
(1000) = 7777
(1001) =
(1145) = 0000

(6000) = 0014        -DISTANCE TO TABLE
(6001) = 0100        } ENTRY/EXIT
(6002) = 0000
(6003) = 5000
(6004) = 0011
(6005) = 5000
(6006) = 1000
(6007) = 0100
(6010) = 0013 12          ADDRESSES TO
(6011) = G                    RELOCATE
(6012) = 0100
(6013) = 0001
(6014) = 0004        } RELOCATION TABLE
(6015) = 0010
(6016) = 0013 12
(6017) = 0000
```

THE CALLING PROGRAM MUST RELOCATE THE OVERLAY



THE LOOP RELOCATES EACH ADDRESS CONSECUTIVELY UNTIL IT REACHES A ZERO WORD - END OF RTAB

```
                                        IDENT  PGM.C.PPEWA
                                        PERIPH
                                        SST
     1000                               ORG    C.PPEWA
     1000       7777                    DATA   -0

                              * MACROS TO REDEFINE INSTRUCTIONS INVOLVING ADDRESSSES
                              LJM.      MACRO  M,D
                                        LOCAL  A
                                        VFD    6/01B,6/D            01 IS FUNCTION CODE FOR LJM
                              A         VFD    12/M-ORG
                                        RMT
                                        VFD    12/A-ORG
                                        RMT
                                        ENDM
                              LDM.      MACRO  M,D
                                        LOCAL  A
                                        VFD    6/50B,6/D            50B IS FUNCTION CODE FOR LDM
                              A         VFD    12/M-ORG
                                        RMT
                                        VFD    12/A-ORG
                                        RMT
                                        ENDM

                              * CODE FOR CALLING PROGRAM
                              ABS       SEGMENT 77B
     2000                               ORG    2000B                CAN BE ANYWHERE
                        2000  ORG       SET    2000B                MUST AGREE WITH ORG STATEMENT

     2000       0012                    VFD    12/RTAB-ORG          DISTANCE TO RELOCATION TABLE

     2001       0100 0000     XENTRY    LJM    0
                        2002  ENTRY     EQU    *-1                  ENTRY AND EXIT POINTS

     2003       5000                    LDM.   LOC                  MUST BE RELOCATED
     2005       0100                    LJM.   *+3                  MUST BE RELOCATED
     2007                     LOC       BSS    1
     2010       0100                    LJM.   XENTRY               MUST BE RELOCATED

     2012                     RTAB      BSS    0                    BEGINNING OF RELOCATION TABLE
                                        LIST   G
                                        HERE
     2012       0004                    VFD    12/*+000001-ORG                                      *RMT*    .1
     2013       0006                    VFD    12/*+000002-ORG                                      *RMT*    .1
     2014       0011                    VFD    12/*+000003-ORG                                      *RMT*    .1
     2015       0000                    DATA   0                    END OF RELOCATION TABLE

     2016                               END

                     46400   STORAGE USED          75 STATEMENTS     733 SYMBOLS      000003 INVERTED SYMBOLS
                           . 6600 ASSEMBLY       0.327 SECONDS       17 REFERENCES
```

12-5

RELOCATION 1

ALTERNATE METHOD

SECTION THIRTEEN

SAMPLE PP PROGRAM

SECTION THIRTEEN - SAMPLE PROGRAMS

CONTENTS

CS 2/72

PP PROGRAM

PP PROGRAM EXAMPLE CODE

```
                                             IDENT     SUM,C.PPFWA
                                 * THIS IS A PP PGM TO SUM 2 NUMBERS FROM CP MEMORY
                                 * IT IS A TEST TO SHOW THE LINKAGE AND HOW TO PUT IT IN THE SYSTEM
                                 * THE PP PGM IS ENTERED FROM OVL BY A LJM
                                 * IT IS A TRANSIENT PROGRAM AND WILL RUN AT 1000
                                             PERIPH                            * TELL ASSEMBLER ITS PP
                                             SST                               * GET ACCESS TO SYSTEM SYMBOLS
 1000                                        ORG       C.PPFWA         * WILL RUN AT 1000
 1000          3074                          PPENTRY   D.PPIRB,D.T0
 1004          3074                          LDD       D.PPIR          * GET CM ADDR OF INPUT REGISTER
 1005          6050                          CRD       D.PPIRB         * GET PARAM WORD FROM CM
 1006          1402                          LDN       2               * FOR 2 MORE CM WORDS
 1007          3402                          STD       2
 1010          3054                          LDD       D.PPIRB+4       * GET CM ADDR OF BUF-1
 1011          1601                          ADN       1               * REL ADDR OF BUF IN A
 1012          0200 0505                     RJM       R.TFL           * ABSOLUTIZE IT
 1014          0603                          PJN       *+3.            * MJN WONT REACH ABT
 1015          0100 1056                     LJM       ABT
 1017          6102 1065                     CRM       BUF,2           * READ TWO DATA WORDS
 1021          5000 1076                     LDM       BUF+9           * GET DATA FROM BUF+1
 1023          5500 1071                     RAM       BUF+4           * ADD DATA FROM BUF
 1025          0200 0446                     RJM       R.RAFL          * GO SEE IF MOVE PENDING
 1027          1401                          LDN       1               * TO WRITE ONE WORD
 1030          3401                          STD       1
 1031          3054                          LDD       D.PPIRB+4       * GET ADDR TO SEND ANS BACK
 1032          1603                          ADN       3               * REL ADDR OF ANS
 1033          0200 0505                     RJM       R.TFL           * GO ABSOLUTIZE IT
 1035          0721                          MJN       ABT
 1036          6301 1065                     CWM       BUF,1           * WRITE ANSWER BACK
 1040          1400                          LDN       0               * ZERO OUTPUT BUFFER
 1041          6010                          CRD       D.T0
 1042          1401                          LDN       1
 1043          3414                          STD       D.T0+4          * SET COMPLETE BIT
 1044          3054                          LDD       D.PPIRB+4
 1045          0200 0505                     RJM       R.TFL           * ABSOLUTIZE PSEUDO FET ADDR
 1047          0707                          MJN       ABT
 1050          6210                          CWD       D.T0            * PUT FET WORD BACK
 1051          1412                          LDN       M.DPP           * GET DROP ME CODE
 1052          0200 0516           EXIT      RJM       R.MTR           $ GO DROP PP
 1054          0100 0103                     LJM       R.IDLE          * GO TO IDLE LOOP
 1056          2000 1077           ABT       LDC       ABTMSG          * ABORT MESSAGE ADDR + FLAG 0
 1060          0200 0671                     RJM       R.DFM           * ISSUE DAYFILE MESSAGE
 1062          1413                          LDN       M.ABORT         * ABORT  OUT OF RANGE
 1063          0100 1052                     LJM       EXIT
 1065                              BUF       BSS       10            * BUFFER FOR CM WORDS
 1077          5502                ABTMSG    DIS       ,* BAD ADDR*
 1105                                        END
                         54202     STORAGE USED            51 STATEMENTS      712 SYMBOLS
                                   6600 ASSEMBLY         0.750 SECONDS         35 REFERENCES
```

13-1

e

```
VER 1.1          SU1,C.PPFWA                                                                              01/14/71
                      SYMBOLIC REFERENCE TABLE

     ABT        0001054                           001013,    001033,    001045
     BUF        0001057                           001015,    001017,    001021,    001034
     C.PPFWA    0001000        SCPTEXT            001071,    000000
     D.PPIR     0000075        SCPTEXT            001002
     D.PPIRB    0001050        SCPTEXT            001003,    001006,    001027,    001042
     D.TP       0000010        SCPTEXT            001037,    001041,    001046
     EXIT       0001050                           001055
     M.ABORT    0000013        SCPTEXT            001054
     M.DEP      0000012        SCPTEXT            001047
     R.TITLE    0000100        SCPTEXT            001052
     R.MIP      0000450        SCPTEXT            001050
     R.PAUSE    0000470        SCPTEXT            001009,    001023
     R.TEL      0000634        SCPTEXT            001010,    001031,    001043
```

(this page is from an earlier run)

Note the symbols used in the
PP program were defined
in SCPTEXT which was
called from the Compass card.

Core map of CP program

```
CORE MAP  02.46.39.  NORMAL                  CONTROL
         ---TIME---LOAD MODE --L1--L2-----TYPE-----------------USER---+---CALL----
         FWA LOADER  054333  FWA TABLES  054274
         -PROGRAM----ADDRESS-                                 --LABELED---COMMON--
         PROG3        000105                                        BLOCK        000100
         CPC          000113
```

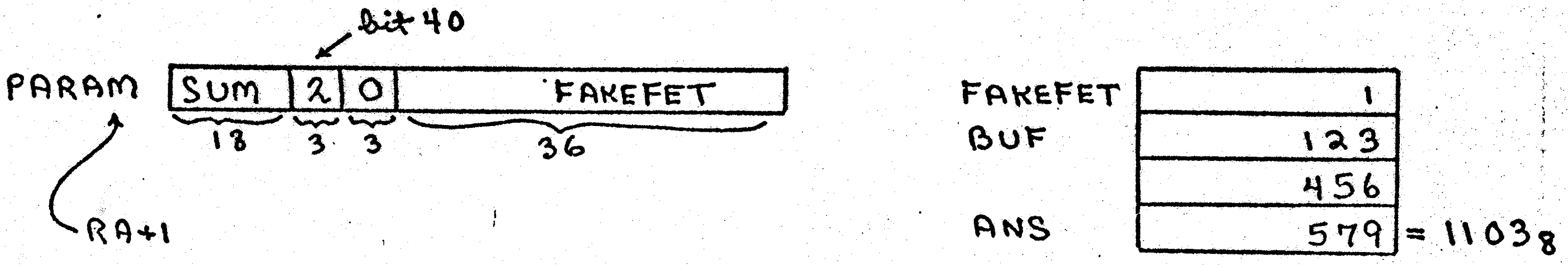
CP PROGRAM

```
PROB3                                                          COMPASS - VER 2.      03/17/72  10.[illegible]

                                         IDENT      PROB3
                                         ENTRY      TESTIT
  0  5110000001                 TESTIT   SA1        1               * CK RA+1 FOR EMPTY
              0311000000 +               NZ         X1,TESTIT
  1  5110000004 C                        SA1        PARAM           * PUT PARAM IN RA+1
              10611                      BX6        X1
  2  5160000001                          SA6        1
  3  5110000001                 LOOP     SA1        1               * WAIT TILL PICKED UP
              0311000003 +               NZ         X1,LOOP
  4  0100000000 X                        ENDRUN     -
                                         USE        /BLOCK/
  0                             FAKEFET  BSSZ       1               * PSEUDO FET TO HOLD COMPLETE BIT
  1  00000000000000000173       BUF      DATA       123,456
  3                             ANS      BSSZ       1
  4  23251520000000000000 C     PARAM    VFD        18/3LSUM,3/2,3/0,36/FAKEFET
  6                                      END        TESTIT
                   43406        STORAGE USED              18 STATEMENTS        7 SYMBOLS
                                6600 ASSEMBLY          0.240 SECONDS          12 REFERENCES
```

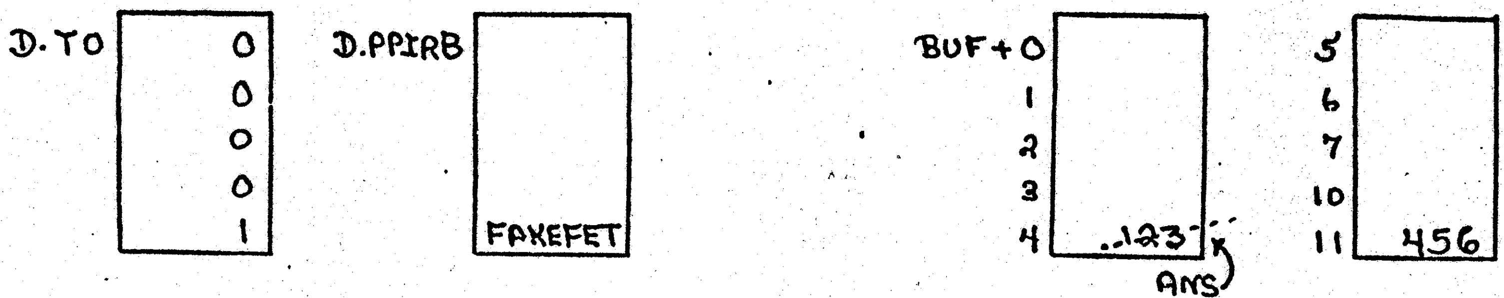
OUTPUT

```
RELATIVE          DMP(100,105)

00100     00000 00000 00000 00001  00000 00000 00000 00173  00000 00000 00000 00710  00000 00000 00000 01103
00104     23251 52000 00000 00100  51100 00001 03110 00105
```

Parameters in CP Program



Parameters in PP Program



DAYFILE

```
 02/17/72   N W L  SCOPE 3.3   LEVEL 250  06/29/71
18.04.55.JOB0008
18.04.55.JOB,CM70000,T30.
18.05.02.ENTL,77777.
18.05.04.GO.
18.05.04.COMPASS(B=PPTEST,S=SCPTEXT)
18.05.05. MINIMUM FIELD LENGTH NEEDED = 054300
18.05.05. ASSEMBLY COMPLETE.
18.05.05.REWIND(PPTEST)
18.05.05.EDITLIB.
18.05.07.         READY(SYSTEM)
18.05.08.         ADD(*,PPTEST)
18.05.08.         COMPLETE.
18.05.42.GO.
18.05.45.COMPASS.
18.05.46. MINIMUM FIELD LENGTH NEEDED = 043500
18.05.46. ASSEMBLY COMPLETE.
18.05.46.LGO.
18.05.47.CPA    051.564 CPB    000.005
18.05.47.PP     007.634 IO     000.764 NEW FL 00500
18.05.47.DMP(100,105)
18.05.47.RFL,70000.
18.05.48.CPA    051.570 CPB    000.005
18.05.48.PP     008.033 IO     000.764 NEW FL 70000
18.05.48.EDITLIB(RESTORE)
18.05.52.GO.
18.05.55.MASS STORAGE  000310 PRU
18.05.55.CPA    051.848 SEC.
18.05.55.CPB    000.005 SEC.
18.05.55.PP     010.311 SEC.
18.05.55.IO     000.955 SEC.
```

13-5

Copy of Card deck
of PPTEST

```
      CCC0PHJ.
      SHEEHYN,CM60000,T7777,P1.
      COMPASS(B=PPTEST,S=SCPTEXT)
      REWIND(PPTEST)
      EDITLIB.                                  PUT IT IN SYSTEM AND MAKE NEW CMR
      COMPASS.                                  ASSEMBLE CP PGM TO TRY IT OUT
      LGO.
      DMP(100,105)
      DMP.
      DMP(100,400)
      EDITLIB(RESTORE)
      EXIT.
      DMP(100,105)
      DMP(100,400)
7     EDITLIB(RESTORE)
8              IDENT       SUM,C.PPFWA
9     * THIS IS A PP PGM TO SUM 2 NUMBERS FROM CP MEMORY
      * IT IS A TEST TO SHOW THE LINKAGE AND HOW TO PUT IT IN THE SYSTEM
      * THE PP PGM IS ENTERED FROM OVL BY A LJM
      * IT IS A TRANSIENT PROGRAM AND WILL RUN AT 1000
               PERIPH                         * TELL ASSEMBLER ITS PP
               SST                            * GET ACCESS TO SYSTEM SYMBOLS
               BASE        O             * MOST PP PGMS ARE OCTAL
               ORG         C.PPFWA       * WILL RUN AT 1000
               RJM         R.PAUSE
               LDD         D.PPIR        * GET CM ADDR OF INPUT REGISTER
               CRD         D.PPIRB       * GET PARAM WORD FROM CM
               LDN         2             * FOR 2 MORE CM WORDS
                {

               MJN         ABT
               CWD         D.T0          * PUT FET WORD BACK
               LDN         M.DPP         * GET DROP ME CODE
      EXIT     RJM         R.MTR         $ GO DROP PP
               LJM         R.IDLE        * GO TO IDLE LOOP
      ABT      LDN         M.ABORT       * ABORT IF ADDR OUT OF RANGE
               LJM         EXIT
      BUF      BSS         12
7              END
8     READY(SYSTEM)
9     ADD(*,PPTEST)
      COMPLETE.
7              IDENT       PROB3
8              ENTRY       TESTIT
9     TESTIT   SA1         1             * CK RA+1 FOR EMPTY
               NZ          X1,TESTIT
               SA1         PARAM         * PUT PARAM IN RA+1
               BX6         X1
               SA6         1
      LOOP     SA1         1             * WAIT TILL PICKED UP
               NZ          X1,LOOP
               ENDRUN
               USE         /BLOCK/
      FAKEFET  BSSZ        1             * PSEUDO FET TO HOLD COMPLETE BIT
      BUF      DATA        123,456
      ANS      BSSZ        1
6     PARAM    VFD         18/3LSUM,3/2,3/0,36/FAKEFET
7              END         TESTIT
8
9
```

CS
2/72

13-7

```
                                          IDENT     SUM,C.PPFWA
                                   * THIS IS A PP PGM TO SUM 2 NUMBERS FROM CP MEMORY
                                   * IT IS A TEST TO SHOW THE LINKAGE AND HOW TO PUT IT IN THE SYSTEM
                                   * THE PP PGM IS ENTERED FROM OVL BY A LJM
                                   * IT IS A TRANSIENT PROGRAM AND WILL RUN AT 1000
                                          PERIPH                        * TELL ASSEMBLER ITS PP
                                          SST                           * GET ACCESS TO SYSTEM SYMBOLS
 1000                                     ORG       C.PPFWA       * WILL RUN AT 1000
 1000         3074                        PPENTRY   D.PPIRB,D.TO
 1004         3074                        LDD       D.PPIR        * GET CM ADDR OF INPUT REGISTER
 1005         6050                        CRD       D.PPIRB       * GET PARAM WORD FROM CM
 1006         1402                        LDN       2             * FOR 2 MORE CM WORDS
 1007         3402                        STD       2
 1010         3054                        LDD       D.PPIRB+4     * GET CM ADDR OF BUF-1
 1011         1601                        ADN       1             * REL ADDR OF BUF IN A

 1012         2110 0000                   ADC       100000B       * GENERATE BAD ADDRESS

 1014         0200 0505                   RJM       R.TFL         * ABSOLUTIZE IT
 1016         0603                        PJN       *+3           * MJN WONT REACH ABT
 1017         0100 1660                   LJM       ABT
 1021         6102 1067                   CRM       BUF,2         * READ TWO DATA WORDS
 1023         5000 1100                   LDM       BUF+9         * GET DATA FROM BUF+1
 1025         5500 1073                   RAM       BUF+4         * ADD DATA FROM BUF
 1027         0200 0446                   RJM       R.RAFL        * GO SEE IF MOVE PENDING
 1031         1401                        LDN       1             * TO WRITE ONE WORD
 1032         3401                        STD       1
 1033         3054                        LDD       D.PPIRB+4     * GET ADDR TO SEND ANS BACK
 1034         1603                        ADN       3             * REL ADDR OF ANS
 1035         0200 0505                   RJM       R.TFL         * GO ABSOLUTIZE IT
 1037         0721                        MJN       ABT
 1040         6301 1067                   CWM       BUF,1         * WRITE ANSWER BACK
 1042         1400                        LDN       0             * ZERO OUTPUT BUFFER
 1043         6010                        CRD       D.TO
 1044         1401                        LDN       1
 1045         3414                        STD       D.TO+4        * SET COMPLETE BIT
 1046         3054                        LDD       D.PPIRB+4
 1047         0200 0505                   RJM       R.TFL         * ABSOLUTIZE PSEUDO FET ADDR
 1051         0707                        MJN       ABT
 1052         6210                        CWD       D.TO          * PUT FET WORD BACK
 1053         1412                        LDN       M.DPP         * GET DROP ME CODE
 1054         0200 0516          EXIT     RJM       R.MTR         $ GO DROP PP
 1056         0100 0103                   LJM       R.IDLE        * GO TO IDLE LOOP
 1060         2600 1101          ABT      LDC       ABTMSG        * ABORT MESSAGE ADDR + FLAG 0
 1062         0200 0671                   RJM       R.DFM         * ISSUE DAYFILE MESSAGE
 1064         1413                        LDN       M.ABORT       * ABORT  OUT OF RANGE
 1065         0100 1054                   LJM       EXIT
 1067                            BUF      BSS       10          * BUFFER FOR CM WORDS
 1101         5502               ABTMSG   DIS       ,* BAD ADDR*
 1107                                     END
                         54202   STORAGE USED               54 STATEMENTS      712 SYMBOLS
                                 6400 ASSEMBLY            0.739 SECONDS         35 REFERENCES
```

TEST ERROR PROCESSING

```
  02/17/72   N W L  SCOPE 3.3   LEVEL 250  06/29/71
 18.12.41.SHEHY1H
 18.12.41.SHEHYN,CM70000.
 18.12.47.GO.
 18.12.47.COMPASS(B=PPTEST,S=SCPTEXT)
 18.12.49. MINIMUM FIELD LENGTH NEEDED = 054300
 18.12.49. ASSEMBLY COMPLETE.
 18.12.49.REWIND(PPTEST)
 18.12.49.EDITLIB.
 18.12.58.          READY(SYSTEM)
 18.12.59.          ADD(*,PPTEST)
 18.12.59.          COMPLETE.
 18.13.18.GO.
 18.13.20.COMPASS.
 18.13.21. MINIMUM FIELD LENGTH NEEDED = 043500  .
 18.13.21. ASSEMBLY COMPLETE.
 18.13.21.LGO.
 18.13.22.CPA   050.592 CPB   001.258
 18.13.22.PP    009.818 IO    000.766 NEW FL 00500
 18.13.22. BAD ADDR
 18.13.22.EXIT.
 18.13.22.DMP(100,105)
 18.13.23.DMP(100,400)
 18.13.23.RFL,70000.
 18.13.23.CPA   050.595 CPB   001.258
 18.13.23.PP    011.092 IO    000.766 NEW FL 70000
 18.13.23.EDITLIB(RESTORE)
 18.13.28.GO.
 18.13.31.MASS STORAGE  000310 PRU
 18.13.31.CPA   050.958 SEC.
 18.13.31.CPB   001.258 SEC.
 18.13.31.PP    013.465 SEC.
 18.13.31.IO    000.957 SEC.
```

dayfile message issued from within PP program

13-8

CS
2/72

```
PROB3                                                          COMPASS - VER 2.     02/17/72  18.33.19.      PAGE     2

                                                  IDENT    PROB3
                                                  ENTRY    TESTIT
      0  5110000001                      TESTIT   SA1      1                * CK RA+1 FOR EMPTY
                   0311000000 +                   NZ       X1,TESTIT
      1  5110000004 C                             SA1      PARAM            * PUT PARAM IN RA+1
                   10611                          BX6      X1
      2  5160000001                               SA6      1
      3  5110000001                      LOOP     SA1      1                * WAIT TILL PICKED UP
                   0311000003 +                   NZ       X1,LOOP
      4  0100000000 X                             ENDRUN
                                                  USE      /BLOCK/

      0  00000000000000000001            FAKEFET  DATA     1                * FAKEFET SHOULD CONTAIN ZERO
                                                                            * OR MTR WILL REJECT CALL
                                                                            * WITH AUTO RECALL ERR MSG

      1  00000000000000000173            BUF      DATA     123,456
      3                                  ANS      BSSZ     1
      4  23251520000000000000 C          PARAM    VFD      18/3LSUM,3/2,3/0,36/FAKEFET
      6                                           END      TESTIT
                             43406       STORAGE USED             22 STATEMENTS          7 SYMBOLS
                                         6600 ASSEMBLY          0.252 SECONDS           12 REFERENCES
```

AUTO RECALL ERROR

13-9

271a

```
  02/17/72 + N W L  SCOPE 3.3   LEVEL 250  06/29/71
18.32.53.SHEHY1T
18.32.53.SHEHYN,CM70000.
18.32.54.GO.
18.32.54.COMPASS(B=PPTEST,S=SCPTEXT)
18.32.55. MINIMUM FIELD LENGTH NEEDED = 054300
18.32.55. ASSEMBLY COMPLETE.
18.32.56.REWIND(PPTEST)
18.32.56.EDITLIB.
18.32.57.          READY(SYSTEM)
18.32.58.          ADD(*,PPTEST)
18.32.58.          COMPLETE.
18.33.16.GO.
18.33.19.COMPASS.
18.33.20. MINIMUM FIELD LENGTH NEEDED = 043500
18.33.20. ASSEMBLY COMPLETE.
18.33.20.LGO.
18.33.21.CPA    051.243 CPB    000.000
18.33.21.PP     006.274 IO     000.765 NEW FL 00500
18.33.21.AUTO-RECALL ERROR
18.33.21.EXIT.
18.33.21.DMP(100,105)
18.33.22.DMP(100,400)
18.33.22.RFL,70000.
18.33.22.CPA    051.246 CPB    000.000
18.33.22.PP     007.496 IO     000.765 NEW FL 70000
18.33.22.EDITLIB(RESTORE)
18.33.25.GO.
18.33.27.MASS STORAGE  000310 PRU
18.33.27.CPA    051.624 SEC.
18.33.27.PP     009.769 SEC.
18.33.27.IO     000.956 SEC.
```

dayfile message issued by MTR

13-10

C3
2/72

```
PR093                                                        COMPASS - VER 2.

                                    IDENT   PR093
                                    ENTRY   TESTIT
  0  5110000001           TESTIT    SA1     1               * CK RA+1 FOR EMPTY
           0311000000 +             NZ      X1,TESTIT
  1  5110000004 C                   SA1     PARAM           * PUT PARAM IN RA+1
           10611                    BX6     X1
  2  5160000001                     SA6     1
  3  5110000001           LOOP      SA1     1               * WAIT TILL PICKED UP
           0311000003 +             NZ      X1,LOOP
  4  0100000000 X                   ENDRUN
                                    USE     /BLOCK/
  0                       FAKEFET   BSSZ    1               * PSEUDO FET TO HOLD COMPLETE BIT
  1  00000000000000000173 BUF       DATA    123,456
  3                       ANS       BSSZ    1

  4  23251520000000100000 C PARAM   VFD     18/3LSUM,3/2,3/0,36/FAKEFET+1000005
                                                    * BAD ADDR IN PARAM WILL CAUSE
                                                    * MTR TO REJECT CALL
                                                    * WITH PF CALL ERROR MSG

  6                                 END     TESTIT
                 43406    STORAGE USED          23 STATEMENTS         7 SYMBOLS
                          5600 ASSEMBLY      0.267 SECONDS           12 REFERENCES
```

back of [illegible]

13-11

PF CALL ERROR

272a

```
 02/17/72 + N W L  SCOPE 3.3  LEVEL 250  06/29/71
18.35.14.SHEHY1W
18.35.14.SHEHYN,CM70000.
18.35.18.GO.
18.35.18.COMPASS(B=PPTEST,S=SCPTEXT)
18.35.19. MINIMUM FIELD LENGTH NEEDED = 054300
18.35.19. ASSEMBLY COMPLETE.
18.35.19.REWIND(PPTEST)
18.35.19.EDITLIB.
18.35.51.           READY(SYSTEM)
18.35.52.           ADD(*,PPTEST)
18.35.52.           COMPLETE.
18.37.33.GO.
18.37.36.COMPASS.
18.37.37. MINIMUM FIELD LENGTH NEEDED = 043500
18.37.37. ASSEMBLY COMPLETE.
18.37.37.LGO.
18.37.38.CPA    052.704 CPB   000.148
18.37.38.PP     023.407 IO    000.765 NEW FL 00500
18.37.38.PP CALL ERROR ←
18.37.38.EXIT.
18.37.38.DMP(100,105)
18.37.39.DMP(100,400)
18.37.39.RFL,70000.
18.37.40.CPA    052.707 CPB   000.148
18.37.40.PP     024.630 IO    000.765 NEW FL 70000
18.37.40.EDITLIB(RESTORE)
18.37.57.GO.
18.37.59.MASS STORAGE  000372 PRU
18.37.59.CPA   053.307 SEC.
18.37.59.CPB    000.148 SEC.
18.37.59.PP     030.776 SEC.
18.37.59.IO     000.956 SEC.
```

dayfile message issued by MTR

CS
2/72

13-13

```
SUM,C.PPFWA                                                  COMPASS - VER 2.     02/17/72  18.16.19.       PAGE

                                        IDENT     SUM,C.PPFWA
                                  * THIS IS A PP PGM TO SUM 2 NUMBERS FROM CP MEMORY
                                  * IT IS A TEST TO SHOW THE LINKAGE AND HOW TO PUT IT IN THE SYSTEM
                                  * THE PP PGM IS ENTERED FROM OVL BY A LJM
                                  * IT IS A TRANSIENT PROGRAM AND WILL RUN AT 1000
                                        PERIPH                       * TELL ASSEMBLER ITS PP
                                        SST                          * GET ACCESS TO SYSTEM SYMBOLS
1000                                    ORG       C.PPFWA      * WILL RUN AT 1000
1000          3074                      PPENTRY   D.PPIRB,D.T0
1004          3074                      LDD       D.PPIR       * GET CM ADDR OF INPUT REGISTER
1005          6050                      CRD       D.PPIRB      * GET PARAM WORD FROM CM
1006          1402                      LDN       2            * FOR 2 MORE CM WORDS
1007          3402                      STD       2
1010          3054                      LDD       D.PPIRB+4.   * GET CM ADDR OF BUF-1
1011          1001                      ADN       1            * REL ADDR OF BUF IN A
1012          0200 0505                 RJM       R.TFL        * ABSOLUTIZE IT
1014          0603                      PJN       *+3          * MJN WONT REACH ABT
1015          0100 1056                 LJM       ABT
1017          6122 1065                 CRM       BUF,2        * READ TWO DATA WORDS
1021          5000 1076                 LDM       BUF+9        * GET DATA FROM BUF+1
1023          5500 1071                 RAM       BUF+4        * ADD DATA FROM BUF
1025          0200 0446                 RJM       R.RAFL       * GO SEE IF MOVE PENDING
1027          1401                      LDN       1            * TO WRITE ONE WORD
1031          3401                      STD       1
1031          3054                      LDD       D.PPIRB+4    * GET ADDR TO SEND ANS BACK
1032          1003                      ADN       3            * REL ADDR OF ANS
1033          0200 0505                 RJM       R.TFL        * GO ABSOLUTIZE IT
1035          0721                      MJN       ABT
1036          6301 1065                 CWM       BUF,1        * WRITE ANSWER BACK
1040          1400                      LDN       0            * ZERO OUTPUT BUFFER
1041          6010                      CRD       D.T0
1042          1401                      LDN       1

1043          3413                      STD       D.T0+3       * SET COMPLETE BIT IN WRONG PLACE'
                                                               * TO HANG IN AUTO RECALL

1044          3054                      LDD       D.PPIRB+4
1045          0200 0505                 RJM       R.TFL        * ABSOLUTIZE PSEUDO FET ADDR
1047          0707                      MJN       ABT
1050          6210                      CWD       D.T0         * PUT FET WORD BACK
1051          1412                      LDN       M.DPP        * GET DROP ME CODE
1052          0200 0516       EXIT      RJM       R.MTR        $ GO DROP PP
1054          0100 0103                 LJM       R.IDLE       * GO TO IDLE LOOP
1056          2000 1077       ABT       LDC       ABTMSG       * ABORT MESSAGE ADDR + FLAG 0
1060          0200 0671                 RJM       R.DFM        * ISSUE DAYFILE MESSAGE
1062          1413                      LDN       M.ABORT      * ABORT  OUT OF RANGE
1063          0100 1052                 LJM       EXIT
1065                          BUF       BSS       10           * BUFFER FOR CM WORDS
1077          5502            ABTMSG    DIS       ,* BAD ADDR*
1105                                    END
                    54202     STORAGE USED                54 STATEMENTS       712 SYMBOLS
                              5600 ASSEMBLY             0.762 SECONDS          35 REFERENCES
```

```
 02/17/72   N W L   SCOPE 3.3   LEVEL 250  06/29/71
18.16.11.SHEHY1I
18.16.11.SHEHYN,CM70000.
18.16.14.GO.
18.16.14.COMPASS(B=PPTEST,S=SCPTEXT)
18.16.15. MINIMUM FIELD LENGTH NEEDED = 054300
18.16.15. ASSEMBLY COMPLETE.
18.16.15.REWIND(PPTEST)
18.16.15.EDITLIB.
18.16.17.           READY(SYSTEM)
18.16.17.           ADD(*,PPTEST)
18.16.18.           COMPLETE.
18.16.38.GO.
18.16.40.COMPASS.
18.16.41. MINIMUM FIELD LENGTH NEEDED = 043500
18.16.41. ASSEMBLY COMPLETE.
18.16.41.LGO.
18.16.42.CPA    051.387 CPB    000.000
18.16.42.PP     006.936 IO     000.766 NEW FL 00500
18.16.43.JOB HUNG IN AUTO-RECALL
18.16.43. ADDRESS =000100
18.16.43.EXIT.
18.16.43.DMP(100,105)
18.16.43.DMP(100,400)
18.16.44.RFL,70000.
18.16.45.CPA    051.390 CPB    000.000
18.16.45.PP     000.813 IO     000.766 NEW FL 70000
18.16.45.EDITLIB(RESTORE)
18.18.19.GO.
18.18.22.MASS STORAGE  000372 PRU
18.18.22.CPA    054.987 SEC.
18.18.22.CPB    000.350 SEC.
18.18.22.PP     054.617 SEC.
18.18.22.IO     000.957 SEC.
```

← dayfile message issued by MTR

13-14

CS
2/72

```
                                               IDENT     SUM,C.PPFWA
                                      * THIS IS A PP PGM TO SUM 2 NUMBERS FROM CP MEMORY
                                      * IT IS A TEST TO SHOW THE LINKAGE AND HOW TO PUT IT IN THE SYSTEM
                                      * THE PP PGM IS ENTERED FROM OVL BY A LJM
                                      * IT IS A TRANSIENT PROGRAM AND WILL RUN AT 1000
                                               PERIPH                            * TELL ASSEMBLER ITS PP
                                               SST                               * GET ACCESS TO SYSTEM SYMBOLS
 1000                                          ORG       C.PPFWA       * WILL RUN AT 1000
 1000       3074                               PPENTRY   D.PPIRB,D.TO
 1004       3074                               LDD       D.PPIR        * GET CM ADDR OF INPUT REGISTER
 1005       6050                               CRD       D.PPIRB       * GET PARAM WORD FROM CM
 1006       1402                               LDN       2             * FOR 2 MORE CM WORDS
 1007       3402                               STD       2
 1010       3054                               LDD       D.PPIRB+4     * GET CM ADDR OF BUF-1
 1011       1601                               ADN       1             * REL ADDR OF BUF IN A

 1012       2110 0000                          ADC       1000000B      * GENERATE BAD ADDRESS

 1014       0200 0505                          RJM       R.TFL         * ABSOLUTIZE IT
 1016       0603                               PJN       *+3           * MJN WONT REACH ABT
 1017       0100 1060                          LJM       ABT
 1021       6102 1072                          CRM       BUF,2         * READ TWO DATA WORDS
 1023       5000 1103                          LDM       BUF+9         * GET DATA FROM BUF+1
 1025       5500 1076                          RAM       BUF+4         * ADD DATA FROM BUF
 1027       0200 0446                          RJM       R.RAFL        * GO SEE IF MOVE PENDING
 1031       1401                               LDN       1             * TO WRITE ONE WORD
 1032       3401                               STD       1
 1033       3054                               LDD       D.PPIRB+4     * GET ADDR TO SEND ANS BACK
 1034       1503                               ADN       3             * REL ADDR OF ANS
 1035       0200 0505                          RJM       R.TFL         * GO ABSOLUTIZE IT
 1037       0721                               MJN       ABT
 1040       6301 1072                          CWM       BUF,1         * WRITE ANSWER BACK
 1042       1400                               LDN       0             * ZERO OUTPUT BUFFER
 1043       6010                               CRD       D.TO
 1044       1401                               LDN       1
 1045       3414                               STD       D.TO+4        * SET COMPLETE BIT
 1046       3054                               LDD       D.PPIRB+4
 1047       0200 0505                          RJM       R.TFL         * ABSOLUTIZE PSEUDO FET ADDR
 1051       0707                               MJN       ABT
 1052       6210                               CWD       D.TO          * PUT FET WORD BACK
 1053       1412                               LDN       M.DPP         * GET DROP ME CODE
 1054       0200 0516                EXIT      RJM       R.MTR         $ GO DROP PP
 1056       0100 0103                          LJM       R.IDLE        * GO TO IDLE LOOP

 1060       1406                     ABT       LDN       6             * SET FLAG BITS FOR DFM
 1061       1014                               SHN       12
 1062       2100 1104                          ADC       ABTMSG        * ADD MSG ADDRESS
 1064       0200 0671                          RJM       R.DFM         * ISSUE DAYFILE MESSAGE
 1065       0300                               UJN       0             * HANG PP

 1067       1413                               LDN       M.ABORT       * ABORT  OUT OF RANGE
 1070       0100 1054                          LJM       EXIT
 1072                                BUF       BSS       10            * BUFFER FOR CM WORDS

 1104       5355                     ABTMSG    DIS       ,*$ DIRTY BAD MSG*

                                               END
```

2/3-15

HANG PP

```
0000  7763 4005 7207 3605 0374 0005 0410 0003  P07   0010  ---- ---- ---- ---- ---- 0700 ---- ----  P07
0020  ---- ---- 0100 0002 0022 0006 3243 0040  P07   0030  7600 0024 0001 0002 ---- 0005 1363 ----  P07
0040  ---- 0005 ---- ---- 0001 0105 ---- ----  P07   0050  2325 1527 ---- ---- 0100 1366 0005 4767  P07
0060  ---- 0362 ---- 0477 ---- 0362 0006 2221  P07   0070  0001 0130 0005 ---- 4110 4112 1600 0162  P07
0100  1600 ---- ---- 7074 6016 3017 0512 0200  P07   0110  0724 1470 1701 0576 3074 6016 3017 0471  P07
0120  1217 1007 3476 0200 0471 2000 0773 0200  P07   0130  0135 0100 1000 ---- 0110 0131 5400 0211  P07
0140  5400 0256 1401 6010 3010 1014 3111 0472  P07   0150  6010 1402 3415 1005 3513 3013 1014 3114  P07
0160  6115 0001 3414 1063 3413 3006 1073 1217  P07   0170  0524 3001 3316 0561 3017 1377 3302 0555  P07
0200  3006 1067 1203 0523 3007 1277 1014 3110  P07   0210  6105 0773 0100 0134 3075 6216 1403 0200  P07
0220  0516 1413 0200 0516 0100 0103 3010 5400  P07   0230  0247 3011 5400 0250 3012 5400 0251 2000  P07
0240  5247 0200 0274 3017 1066 0444 0352 0377  P07   0250  0013 0043 0011 3400 ---- 4112 0773 1400  P07
0260  7777 ---- ---- ---- ---- ---- ---- ----  P07   0270  ---- ---- ---- 0100 0243 3410 1471 0307  P07
0300  1012 0671 0100 2205 3410 1473 0200 0320  P07   0310  3613 3075 1602 6213 5000 0370 0361 0100  P07
0320  0310 1006 5400 0370 5010 0007 5400 0371  P07   0330  3075 5410 0606 0200 0407 5000 0100 0403  P07
0340  0200 0446 1462 1701 0576 3075 1602 6013  P07   0350  3013 1703 0413 0643 1602 0757 0566 2000  P07
0360  0402 0200 0663 3613 0312 3017 6525 0366  P07   0370  7125 1120 3017 5500 0371 3713 3075 1602  P07
0400  6013 0343 0016 0356 0370 ---- 0100 0335  P07   0410  3010 5400 0425 3076 1001 5410 0004 1402  P07
0420  3410 1400 6001 3075 6310 0247 6211 1403  P07   0430  3414 1406 0200 0516 0351 6000 0100 3376  P07
0440  0522 3014 3455 2113 3455 0100 3542 3176  P07   0450  1020 0010 3012 0461 0200 3471 1452 1701  P07
0460  0575 0355 3076 5400 0100 3077 6276 0357  P07   0470  0100 0125 1400 5400 0100 3077 6276 0370  P07
0500  1014 3156 3155 1006 0100 1016 0705 1014  P07   0510  3256 1006 0765 1500 0367 6100 0722 3410  P07
0520  3074 1601 6210 0200 0527 0367 3100 0525  P07   0530  3074 1601 6010 3013 0471 5000 0100 0403  P07
0540  3233 0446 1462 1701 0575 0362 ---- ----  P07   0550  ---- ---- ---- ---- ---- ---- ---- ----  P07
0560  0100 2352 3411 1402 3414 0200 0516 3013  P07   0570  5400 0102 3012 5400 0101 0362 0100 2430  P07
0600  2100 0110 6010 3414 6210 3376 0467 3011  P07   0610  1714 0705 1704 0762 1714 0660 1440 6010  P07
0620  3012 5200 0102 5400 0102 0615 2100 1750  P07   0630  5400 0102 1501 3111 0302 3011 5200 0101  P07
0640  0603 2101 ---- 5400 0101 3077 6276 0100  P07   0650  0576 3412 4012 2200 7700 4300 4412 3610  P07
0660  4010 0567 0100 0353 3410 4010 3400 0367  P07   0670  0100 1066 3410 1063 3411 3075 3412 1413  P07
0700  3400 4010 4400 0402 3610 3600 1120 0571  P07   0710  3312 6213 3612 1207 0403 3017 0560 1401  P07
0720  0200 0516 0345 0100 0111 3377 6216 1400  P07   0730  5400 0101 5400 0102 0366 ---- ---- ----  P07
0740  ---- ---- ---- ---- ---- ---- ---- ----  P07   0750  ---- ---- ---- ---- ---- ---- ---- ----  P07
0760  ---- ---- ---- ---- ---- ---- ---- ----  P07   0770  ---- ---- ---- 2325 1500 0773 ---- 0021  P07
1000  3474 6056 0200 0446 3374 6050 1402 3402  P07   1010  3054 1601 2110 ---- 0200 0202 0503 [illegible]  P07
1020  1060 6102 1072 5300 1103 5500 1076 0200  P07   1030  0446 1401 3401 3054 1603 0200 0505 0721  P07
1040  6001 1072 1400 6010 1401 3414 3054 0200  P07   1050  0505 0707 6210 1412 0200 0516 0100 0103  P07
1060  1406 1014 2100 1104 0200 0671 ---- 1413  P07   1070  0100 1054 ---- ---- ---- ---- ---- ----  P07
1100  ---- ---- ---- ---- 5355 0411 2224 3155  P07   1110  0201 0455 1523 0700 ---- ---- ---- ----  P07
1120  3644 3053 1277 1014 3154 0200 0505 0707  P07   1130  6240 1412 0200 0516 0100 0103 2000 1601  P07
1140  0200 0671 1413 0366 2000 1671 0100 1140  P07   1150  2000 1714 0100 1143 1401 3432 3037 3413  P07
1160  0311 2022 ---- 1071 3234 0756 3034 3411  P07   1170  0311 2030 ---- 3234 0603 0103 1144 3034  P07
1200  3412 1417 3230 0516 4230 3446 3455 3014  P07   1210  3456 3011 0403 0100 1357 0236 ---- 3035  P07
1220  3334 0503 0100 1344 3732 0403 0100 1156  P07   1230  3037 0415 2003 3355 5400 1655 2000 0503  P07
1240  5400 1656 2000 2355 5400 1657 1454 6010  P07   1250  3010 0404 1007 3276 0406 3011 0406 1007  P07
1260  [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] 1063  P07   1270  3006 1003 3306 1370 5500 1653 3037 0504  P07
1300  2003 2222 0337 1374 0502 1422 1006 3307  P07   1310  1370 5500 1652 1403 3407 3076 1632 6307  P07
1320  1544 3051 1340 1640 3451 3075 6250 1400  P07   1330  6010 2000 1750 3412 1437 0200 0516 1412  P07
1340  0200 0516 0100 0103 3051 1240 0503 0100  P07   1350  1770 1400 6010 3076 1635 6210 0373 1402  P07
1360  3544 0362 0100 1217 0203 0446 3455 3014  P07   1370  3435 0370 0100 ---- 3076 1622 6010 3014  P07
1400  3435 0370 1456 6020 3034 3220 0603 0100  P07   1410  1154 3431 1400 3426 1401 3425 3025 1007  P07
1420  1026 3427 6020 3024 1702 0736 5425 1624  P07   1430  3430 3027 1602 6020 3020 0426 2177 2777  P07
1440  0623 3027 1620 6020 3022 0416 3027 2100  P07   1450  0151 6020 3020 1366 0707 3027 2100 0145  P07
1460  6023 3023 0435 1400 5425 1624 3430 3030  P07   1470  3526 3625 1120 0403 0100 1416 3026 3231  P07
1500  0603 0100 1154 1401 3427 1400 3425 3426  P07   1510  5027 1624 0424 3231 0711 3026 3231 0712  P07
1520  5027 1624 3226 0513 0305 5027 1624 3226  P07   1530  0706 5027 1624 3426 3027 3425 3027 1120  P07
1540  0547 3025 0415 0200 1565 3031 3226 0710  P07   1550  0437 3431 1400 5427 1624 0100 1503 2000  P07
1560  4000 3432 0100 1156 0100 ---- 1007 2100  P07   1570  0152 3401 6020 3024 0567 3001 1070 1217  P07
1600  2100 1700 3421 2000 3422 3420 1400 3422  P07   1610  3423 2000 0703 3424 3075 6220 1400 6020  P07
1620  1437 0200 0516 0100 1564 ---- ---- ----  P07   1630  ---- ---- ---- ---- ---- ---- ---- ----  P07
1640  ---- ---- ---- ---- 2701 1124 1116 0755  P07   1650  0617 2255 3333 3333 3333 5503 1555 5555  P07
1660  ---- 1505 1555 0122 0755 0522 2217 2200  P07   1670  ---- 1505 1555 0122 0755 0522 2217 2246  P07
1700  2205 2125 0523 2455 0530 0305 0504 2355  P07   1710  1411 1511 2400 ---- 1505 1555 0122 0755  P07
1720  5522 2217 2246 2324 0124 2523 5501 1422  P07   1730  0511 0431 5503 1715 2014 0524 0500 ----  P07
1740  1116 0322 0501 2311 1607 5506 1455 0124  P07   1750  5511 1624 0522 0317 1555 0320 5516 1724  P07
1760  5501 1414 1727 0504 5551 1505 1552 ----  P07   1770  3076 2100 0165 6001 3001 0403 0100 1120  P07
2000  2000 0100 3471 1400 3405 1423 0200 2071  P07   2010  1407 3405 1424 0200 2071 2011 2224 0200  P07
2020  0671 1417 3405 1425 0200 2071 1426 3405  P07   2030  2000 0164 0200 2071 3034 1277 3400 1003  P07
2040  3300 1370 5500 2265 3034 1071 3434 1207  P07   2050  5500 2264 3034 1270 6503 2000 0220 1003  P07
2060  5500 2264 2011 2243 0200 0671 0100 1120  P07   2070  0100 2034 3176 6010 3012 3113 3114 0503  P07
2100  0100 2210 3412 1014 3113 2175 4357 0710  P07   2110  3413 1063 3412 3071 5565 2226 0353 3012  P07
2120  1014 3113 2177 0127 3737 3413 1063 3412  P07   2130  5505 2226 0764 3113 2177 7033 0706 3413  P07
2140  3071 5505 2227 0307 3013 1712 0705 3413  P07   2150  5605 2227 0371 1012 1005 5505 2230 3015  P07
2160  2177 7633 0706 3414 3071 5505 2231 0367  P07   2170  3014 1712 0705 3414 5665 2231 0371 1012  P07
2200  1306 5505 2232 5005 2226 1071 1133 0513  P07   2210  2000 2210 5505 2226 1277 1133 0334 1412  P07
2220  5515 2226 0100 2670 0320 0155 5555 3333  P07   2230  3557 4335 4455 0320 0255 5555 3336 [illegible]  P07
2240  4135 3400 ---- 2020 5555 5555 3334 4157  P07   2250  3534 3455 1117 5555 5556 3333 3357 [illegible]  P07
2260  4155 1605 2755 0614 5533 3340 3333 ----  P07   2270  ---- ---- ---- ---- 3602 3014 4502 [illegible]  P07
2300  3011 1105 0562 3056 1005 3256 1006 3167  P07   2310  1701 0614 3564 3266 1314 3165 3257 [illegible]  P07
```

HUNG PP — PARTIAL DUMP

13-16

Example showing assembly error because MJN could not reach ABT

13-17

```
                                     IDENT    SUM,C.PPFWA
                                     PERIPH                   * TELL ASSEMBLER ITS PP
                                     SST                      * GET ACCESS TO SYSTEM SYMBOLS
                                     BASE     0               * MOST PP PGMS ARE OCTAL
    1000                             ORG      C.PPFWA         * WILL RUN AT 1000
    1000    0200 0430                RJM      R.PAUSE
    1002    3075                     LDD      D.PPIR          * GET CM ADDR OF INPUT REGISTER
    1003    6050                     CRD      D.PPIRB         * GET PARAM WORD FROM CM
    1004    3054                     LDD      D.PPIRB+4       * GET CM ADDR OF BUF-1
    1005    1601                     ADN      1               * REL ADDR OF BUF IN A
    1006    0200 0634                RJM      R.TFL           * ABSOLUTIZE IT
(A) 1010    0742                     MJN      ABT
    1011    1402                     LDN      2               * FOR 2 MORE CM WORDS
    1012    3402                     STD      2
    1013    6102 1055                CRM      BUF,2           * READ TWO DATA WORDS
    1015    5000 1066                LDM      BUF+11          * GET DATA FROM BUF+1
    1017    5500 1061                RAM      BUF+4           * ADD DATA FROM BUF
    1021    0200 0430                RJM      R.PAUSE         * GO SEE IF STORAGE MOVE WAITING
    1023    1401                     LDN      1               * TO WRITE ONE WORD
    1024    3401                     STD      1
    1025    3054                     LDD      D.PPIRB+4       * GET ADDR TO SEND ANS BACK
    1026    1603                     ADN      3               * REL ADDR OF ANS
    1027    0200 0634                RJM      R.TFL           * GO ABSOLUTIZE IT
    1031    0721                     MJN      ABT
    1032    6301 1055                CWM      BUF,1           * WRITE ANSWER BACK
    1034    1400                     LDN      0               * ZERO OUTPUT BUFFER
    1035    6010                     CRD      D.T0
    1036    1401                     LDN      1
    1037    3414                     STD      D.T0+4          * SET COMPLETE BIT
    1040    3054                     LDD      D.PPIRB+4
    1041    0200 0634                RJM      R.TFL           * ABSOLUTIZE PSEUDO REF ADDR
    1043    0707                     MJN      ABT
    1044    6210                     CWD      D.T0            * PUT REF WORD BACK
    1045    1412                     LDN      M.DROP          * GET DROP RES CODE
    1046    0200 0450       EXIT     RJM      R.MTR           * GO DROP PP
    1050    0100 0100                LJM      R.IDLE          * GO TO IDLE LOOP
    1052    1413           (ABT)     LDN      M.ABORT         * [illegible] OUT OF RANGE
    1053    0100 1046                LJM      EXIT
    1055                    BUF      BSS      12B
    1067                             END

A  TYPE ERRORS  1            ADDRESS FIELD BAD
   OCCUPIED [illegible] PAGES
```

Error Example: In another run, some of the instructions were moved, closer to the others which used them. This run hung 6 pp's and killed the whole system, necessitating initial deadstart. WHY? 11/14/71

```
                                    IDENT     SOT,C.PPEWA
                                    PERIPH                            * TELL ASSEMBLER ITS PP
                                    SST                               * GET ACCESS TO SYSTEM SYMBOLS
                                    BASE      0                  * MOST PP NUMS ARE OCTAL
  1000                              ORG       C.PPEWA            * WILL RUN AT 1000
  1000      0200 0430               RJM       R.PAUSE
  1002      3075                    LDD       D.PPIR             * GET CM ADDR OF INPUT REGISTER
  1003      6050                    CRD       D.PPIRB            * GET PARAM WORD FROM C1
  1004      1402                   {LDN       2                  * FOR 2 MORE CM WORDS
  1005      3402                   {STD       2
  1006      3054                    LDD       D.PPIRB+4          * GET CM ADDR OF BUF-1
  1007      1601                    ADN       1                  * RFL ADDR OF BUF IN A
  1010      0200 0634               RJM       R.TFL              * ABSOLUTIZE
  1012      0603                    PJN       *+3                * MJN WONT REACH
  1013      0100 1054               LJM       ABT
  1015      6102 1057               CRM       BUF,2              * READ TWO DATA WORDS
  1017      5000 1070               LDM       BUF+1              * GET DATA FROM BUF+1
  1021      5500 1063               RAM       BUF+4              * ADD DATA FROM BUF
  1023      0200 0430               RJM       R.PAUSE            * GO SEE IF STORAGE MOVE WAITING
  1025      1401                   {LDN       1                  * TO write 1 word
  1026      3401                   {STD       1
  1027      3054                    LDD       D.PPIRB+4          * GET ADDR TO SEND ANS BACK
  1030      1603                    ADN       3                  * REL ADDR OF ANS
  1031      0200 0634               RJM       R.TFL              * GO ABSOLUTIZE IT
  1033      0721                    MJN       ABT
  1034      6301 1057               CWM       BUF,1              * WRITE ANSWER BACK
  1036      1400                    LDN       0                  * ZERO OUTPUT BUFFER
  1037      6010                    CRD       D.T0
  1040      1401                    LDN       1
  1041      3414                    STD       D.T0+4             * SET COMPLETE BIT
  1042      3054                    LDD       D.PPIRB+4
  1043      0200 0634               RJM       R.TFL              * ABSOLUTIZE PSEUDO FET ADDR
  1045      0707                    MJN       ABT
  1046      6210                    CWD       D.T0         PUT IT BACK
  1047      1412                    LDN       M.DPP              * GET DROP ME CODE
  1050      0200 0450     EXIT      RJM       R.MTR              $ GO DROP PP
  1052      0100 0100               LJM       R.IDLE             * GO TO IDLE LOOP
  1054      1413          ABT       LDN       M.ABORT            * ABORT IF ADDR OUT OF RANGE
  1055      0100 1050               LJM       EXIT
  1057                    BUF       BSS       12
  1071                              END
```

REPORT 1