CONTROL DATA
CORPORATION

CONTROL DATA®
6000/CYBER 70/CYBER 170
SYSTEM MAINTENANCE MONITOR (SMM)

Volume 1 of 3

REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01	Original printing. Preliminary Edition.
(3-29-66)	
02	Publications Change Order 15180 added new diagnostic programs and made miscellaneous correc-
(12-6-66)	tions. This edition obsoletes all previous editions.
03	Publications Change Order 15664 added new diagnostic programs and made miscellaneous
(2-3-67)	corrections.
04	Publications Change Order 16216 added new diagnostic programs and made miscellaneous
(4-17-67)	corrections.
05	Publications Change Order 17594 added new diagnostic programs (TD, DF2, MM65, and ECS16)
(9-29-67)	and made miscellaneous corrections.
06	Publications Change Order 18561 added pages 445-1 through 445-11 (DF2) left out of Rev 05.
(1-25-68)	
07	Publications Change Order 19335 added General Information section and new diagnostic programs.
(5-27-68)	Manual extensively revised and divided into two volumes. This edition obsoletes all previous
	editions.
08	Manual is revised and divided into five volumes and is complete through edition 2.1/2.2. This
(12-2-68)	edition supersedes all previous editions.
09	Manuals are revised to include new tests and revise or exclude other tests. This manual is
(2-15-69)	complete through edition 3.0.
10	Manual revised to correct page numbers. This manual is complete through edition 3.0.
(5-1-69)	
11	Manuals are revised to include new and revised tests and to exclude other tests. This manual is
(8-25-69)	complete through edition 3.1.
12	Manuals are revised to include new and revised tests and to exclude other tests. This manual is
(3-15-70)	complete through edition 3.2.
13	Manuals are revised to include new and revised tests. This manual is complete through edition 3.3.
(9-15-70)	
14	Manuals are revised to include new and revised tests, to exclude other tests, and to include
(9-15-71)	Volume 6. This manual is complete through edition 3.4.
15	Manuals are revised to include new and revised tests and to replace other tests. This manual is
(11-15-72)	complete through edition 3.5.
16	Manuals are revised to include miscellaneous corrections and add PSM Test to Volume 1.
(1-20-73)	
17	Manuals are revised to include new and revised tests and to exclude other tests.
(6-1-73)	
18	Manuals are revised to include new and revised tests.
(4-1-74)	

Publication No.
60160600

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 Lexington Avenue North
St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

REVISION RECORD (CONT'D)

REVISION	DESCRIPTION
A	Manual released. This edition obsoletes all previous editions.
(6-2-75)	

Publication No.
60160600

PREFACE

The CONTROL DATA® 6000/CYBER 70/CYBER 170 System Maintenance Monitor is a collection of programs designed to aid in the maintenance and checkout of CONTROL DATA 6000/CYBER 70/CYBER 170 Computer Systems.

SMM 6000/CYBER 70/CYBER 170 resides on magnetic tape with tape to disk options and has the following three modes of operation.

ENS	A PPU-based diagnostic is loaded to and executed from PPU.
Stand-Alone	Single programs are loaded by system program LDR after being called through keyboard input to system program CPC.
Auto	Multiple programs may be loaded and executed using various PPUs and the CPU. Communication is handled through pseudo registers in central memory. Load program LDR resides in PPU 0 and central processor control program CPC resides in PPU 10.

The purpose of SMM is to provide the customer engineer with a diagnostic to be used for preventative and emergency maintenance. Diagnostics can be written using SMM and have freedom to exercise the hardware in any way, shape, or form.

Additional information pertaining to CYBER 170 diagnostics may be found in the supplemental edition to this manual (publication no. 60409500).

GENERAL CONTENTS

CONTENTS

VOLUME 1

DEADSTART OPTIONS

SMM SYSTEM

UTILITY ROUTINES

TEST LOAD

PROGRAMMING FOR SMM 6000

APPENDIXES

VOLUME 2

CENTRAL PROCESSOR DIAGNOSTICS

Command Tests

MEMORY DIAGNOSTICS

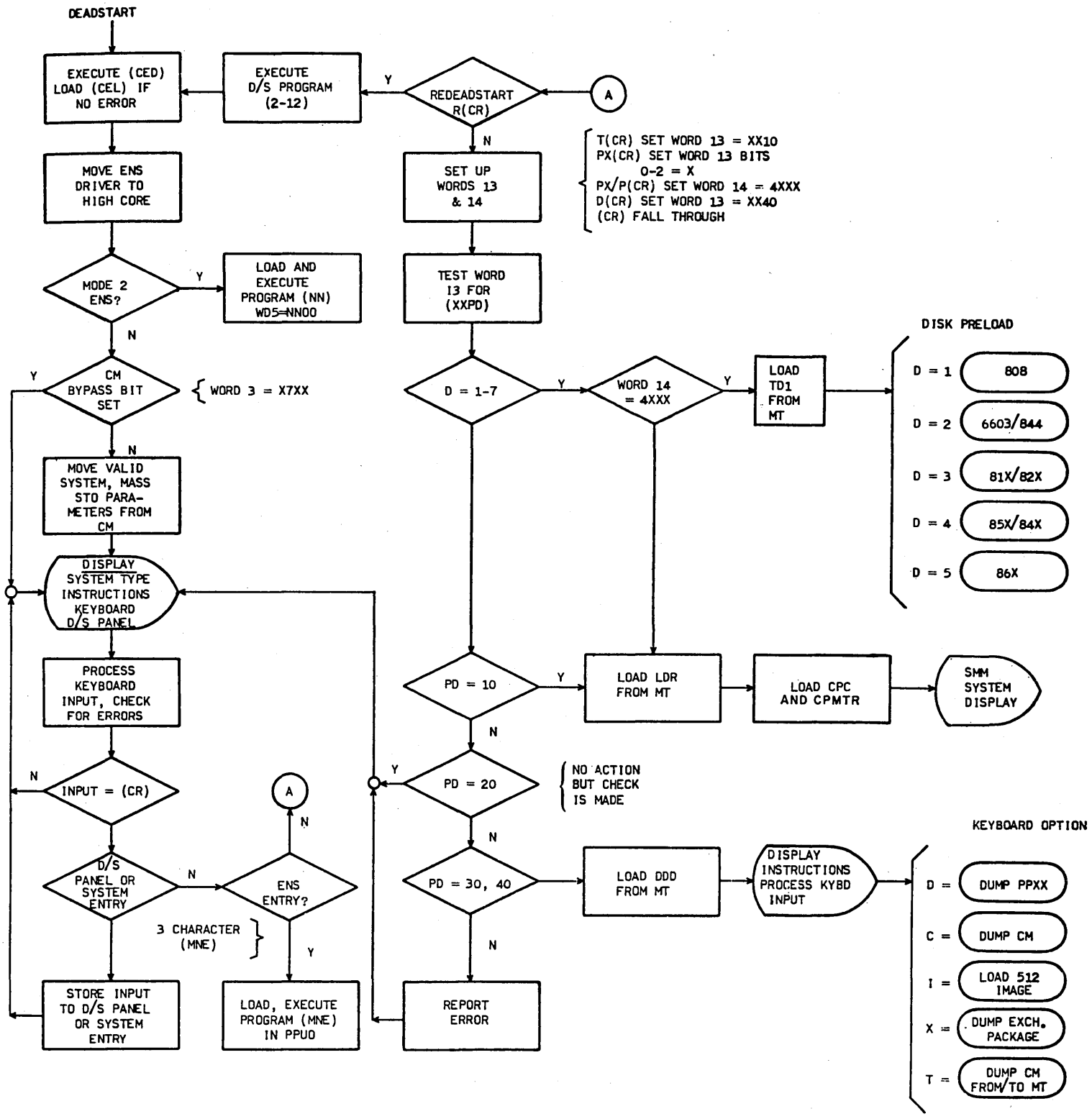
PERIPHERAL PROCESSOR DIAGNOSTICS

ECS DIAGNOSTICS

VOLUME 3

PERIPHERAL EQUIPMENT DIAGNOSTICS

6000/CYBER DEADSTART FLOWCHART



DEADSTART PANEL SETTINGS

HARDWARE

Deadstart of a 6000 computer consists of the following.

1. The CPU is stopped, but all registers remain intact.
2. All I/O channels are master cleared.
3. All 6681's are set selected.
4. PPU A registers are forced to a value of 10000_8.
5. All PPUs are set inputting on the channel number corresponding to their PPU number. That is, PP-5 is set inputting on channel 5.
6. All channels are set empty and active.
7. The deadstart synchronizer dumps the contents of the deadstart panel over channel 0 into PPU-0.
8. PPU-0 begins execution of the deadstart program.

SETTINGS

To load 80-column binary cards:

Address	Contents	
0001	75CC	Deactivate channel CC.
0002	77CC	Connect card reader E.
0003	E000	
0004	77CC	Select binary.
0005	0001	
0006	77CC	Select 6681 to read.
0007	1500	
0010	2000	Load word count.
0011	7760	
0012	74CC	Activate channel CC.
0013	71CC	Input to address 0.
0014	0000 or 7773	

NOTE

E may be 4, 5, 6, or 7 and CC may be 12 or 13.

With this panel setting, an 80-column binary card deck will load into PP-0 at address zero and begin execution at FWA, where FWA-1 is the contents of column 1 of card 1 (that is, contents of address zero of the program). The computer appears to the program as described in deadstart hardware.

To load binary cards:

Address	Contents	
0001	75CC	Disconnect channel CC.
0002	77CC	Connect card reader E.
0003	E000	
0004	77CC	
0005	0001	
0006	77CC	
0007	1400	
0010	74CC	
0011	71CC	
0012	7666	
0013	XXXX	
0014	XXXX	

NOTE

E may be 4, 5, 6, or 7 and CC may be 12 or 13.

1. The above panel setting will load a binary card deck punched by COMPASS into PP-0 at address zero. The program will begin execution at FWA, where FWA-1 is the contents of address zero of the program.
2. Card 1 of the deck must be removed and the deck must be preceded by a special loader card. The deck must be terminated by a card with 6789 punch in column 1.
3. The computer will appear to the program as outlined in deadstart hardware.
4. For binary decks from COMPASS, refer to the programming section.

5. To obtain the special loader card, assemble and punch program DSCL in the SMM program library. The channel is preset to 12 and must be changed if this is not the case.

PANEL SETTING FOR CHANNEL 12 OR 13

01	75CC	Deactivate channel CC.
02	77CC	Connect tape.
† †03	E00U	Equipment E, unit U.
† †04	77CC	Rewind tape.
† †05	0010	
06	77CC	6681 read.
07	1400	
10	74CC	Activate channel C.
11	71CC	Input to address 13.
†12	0013	
†13	----	Not used V3.5.
†14	----	

DEADSTART SEQUENCE

The first two physical records on the deadstart tape are the deadstart command test CED which is executed in PPU0 immediately after deadstart. Since CED is a modified PC1, it will hang the PPU (0300 instruction) if it detects an error condition. P must be scoped and the listing of CEQ consulted to determine the failing instruction. Since no further execution is possible, a listing of CED is a site necessity. Following the successful execution of CED, the preload control program CEL is executed displaying usage instructions on the left screen and the deadstart panel on the right.

NOTE

If following a deadstart the screens are blank, it should be assumed that CED has detected a failure in PPU0.

† Change from V3.4, refer to CEL writeup for program system selection.
† † Consult CEL writeup for CM bypass and ENS mode II settings.

SMM TAPE ON CHANNEL 4, 5, 6, OR 7

Address	Contents	
0001	75CC	Disconnect channel CC.
0002	77CC	Connect equipment E.
0003	E000	
0004	77CC	Set binary.
0005	0001	
0006	77CC	Set 6681 to read.
0007	1400	
0010	74CC	Activate channel CC.
0011	71CC	Input to address 7666.
0012	7666	

NOTE

E may be 4, 5, 6, or 7.

The above setting will load a card into PP-0 at address 7666. The routine on the card idles the PPU on the channel bearing the SMM tape and reads record 1 from the SMM tape. Record 1 is the deadstart diagnostic (CED) which tests PPU 0 and then loads the preloader (CEL).

The card to be used is obtained by assembling and punching program (PTL) on the SMM program library. PTL assumes card reader on channel 12, tape on channel 7, equipment 5, unit 0. PTL must be modified if this is not the case. SMM considers that a PPU has been idled and brings it out of idle condition for use.

WARMSTART

SMM from MTS (66X) tape subsystem controlware must be loaded and running.

DEADSTART MTS ON CHANNEL 12, 13, OR 32, 33

Use the following deadstart settings.

```
 1- 75CC
 2- 1701
 3- 0576
 4- 77CC or 00CC          (Deselect 668X or PSN if the MTS channel
 5- 21RS or 00RS           does not have a 668X on it.)
 6- 77CC
 7- ED6U MTS warmstart function
10- 74CC
11- 71CC
12- 0013
13- ----
14- ----
```

where CC = MTS channel
R = 7=Bypass CM (SCOPE, R=CMR NBR)
S = SCOPE, save PP0
E = MTS equipment
D = 66X density (0=556, 2=800)
U = 66X unit

MTS ON CHANNELS 1 THROUGH 11 OR 20 THROUGH 31

SMM will have to be initialized by deadstarting a binary card deck obtained by assembling PT6 on the update I OLDPL tape.

Use the following deadstart settings for PT6.

```
1- 75CC
2- 1701
3- 0576
4- 77CC
5- ERTT
```

where CC = CR channel
E = CR equipment
R = 7=bypass CM (CEL)
TT = MTS channel

6- 77CC
7- 14DL
10- 74CC
11- 71CC
12- 7664
13- UU
14- ----

where D = 66X density (0=556, 2=800)
L = 2=ENS mode
UU = 66X unit (0-17)

Sample job deck for assembling PT6:

```
JOB Card
REQUEST,OLDPL,HI.                    SMM.5-3 OLDPL I
UPDATE(Q)
COMPASS(I,B=PUNCHB)
7 8 9 (EOR)
* IDENT PT6CH
* D PT6.32
* CRCH   EQU    12B                  Card Reader Channel
* C PT6
6 7 8 9 (EOF)
```

NOTE

The binary card deck will be the following:

Card 1 - Binary card (modified version of DSCL)
2 - 7,8,9 (EOR)
3 - Binary card
4 - Binary card
5 - 7,8,9 (EOR)
6 - 6,7,8,9 (EOF)

COLDSTART

Deadstart SMM from MTS (66X) tape subsystem controlware deck must be available.

CEJ is a routine that is read from the card reader by the deadstart panel. It checks the PP0 SAVE switch on the panel and saves the contents of PP0 in central memory if it is not set. It then reads the 1ST binary card of segment CEY and goes to execute it. CEJ must be exactly one binary card in length. The CEJ and CEY cards must be obtained from the SCOPE operating system for that site.

The following is the MTS coldstart deadstart deck structure.

CEJ 1 binary card (record) (unprefixed)
7/8/9 EOR
CEY binary deck (unprefixed)
7/8/9 EOR
MTS controlware (unprefixed)
7/8/9 EOR
6/7/8/9 EOR

The following are card reader panel settings.

	CR on Channel 12 or 13	CR on Channels 1 through 11
1	75CC	73CC
2	2400	0013
3	2400	75CC
4	77CC	77CC
5	ERTT	ERTT
6	77CC	77CC
7	14DS	14DS
10	74CC	74CC
11	71CC	71CC
12	7664	7664
13	00UU	0000
14	----	*7112

where CC = Card reader channel number

E = Card reader equip number (must be .GE,4)

R = CMR number

U = MTS unit number (if CR on 12 or 13 only)

TT = MTS DS channel NC

D = MTS CS tape density (2**4)

S = PP0 save switch (C**2)

NOTE

If MTS is on channel 12, change word 14 to 7113.

DEADSTART CONTROL PROGRAMS

PERIPHERAL PROCESSOR DEADSTART COMMAND TEST (CED)

INTRODUCTION

CED (a modified, segmented version of PC1) is an inline, fixed operand sequential command test that executes in PPU 0 following deadstart. Its purpose is to verify basic operand of PPU 0 prior to the execution of the preload program CEL. To obtain listings of CED, call CEQ from the OLDPL tape.

REQUIREMENTS

HARDWARE

CED will execute in any 6000/CYBER configuration: PPU 0, channel 0, mag tape equipment, and channel. CED does not require that central memory be operational nor does it attempt to check CM instructions.

SOFTWARE

CED is resident as the first record on the SMM 6000/CYBER deadstart tape.

OPERATIONAL PROCEDURE

LOADING PROCEDURE

CED is loaded by the standard SMM deadstart panel program. This program is reentered at address 6 to load the second segment of CED and the preloader CEL.

PARAMETERS

No parameters are required.

OPERATOR COMMUNICATION

Since no hardware other than that necessary for deadstart is assumed operational, no messages are displayed. If an error condition is detected, a 0300 is executed effectively stopping the PPU. Consult the listing of CED for failure description.

DESCRIPTION

Since CED is essentially a modified PC1, refer to the writeup of PC1 for test description, keeping in mind that only PPU 0 is tested.

PRELOAD CONTROL PROGRAM (CEL)

Following the successful execution of the deadstart diagnostic CED, CEL is loaded and initiated. The function of CEL is to interpret keyboard input directives, describing a parameter change or requesting the loading and initiating of a PPU 0 program, a system utility program, or the SMM system. The following is a list of these directives.

† SYbb, IIII (cr)	Change the configuration for 6000 and CYBER 70 description, where bb = number of banks and IIII = CPU type. Example: SY40, 7428.
SYbbb, III, c, pp (cr) or SYbb, III, c, pp (cr)	Change of configuration description for CYBER 170. bbb = Number of banks C = Number CPU (1 or 2) III = CPU type pp = Number of PPUs Example: SY100, 175, 1, 20 or SY40, 175, 2, 10
xxWyyyy(cr)	Change the contents of address xx of the deadstart area (1 through 14B) to yyyy (for 6000/CYBER 70).
xxWyyyy (cr)	Change the contents of address (xx) of the deadstart area (1 through 20B) to yyyy on CYBER 170 machines only.
xx+yyyy (cr)	Similar to the previous entry, except 1 is added to xx allowing sequential entry.
yyyy (cr)	Status/control register functions. (Refer to CYBER 17X Reference Manual.) This command will execute the status/control register function, where yyyy is the octal function code.
† CR	Loads and initiates the program selected by the contents of addresses 13 and 14. This is typically used on successive deadstarts following a preload of mass storage (PX).
MNE (cr)	Direct loads and initiates PPU 0 program MNE without using CM. A nonexistant MNE will give an error display.
T (cr)	Loads and initiates the SMM system using the deadstart tape as the library media.

† The configuration and mass storage device parameters are saved in CM addresses 1 and 242, respectively when SMM is initiated so successive deadstarts will not require redefining these parameters. If this feature is not desired, set 03 = E7UU to eliminate this use of CM.

†PX (cr)	Loads and initiates the utility tape to mass storage program TD1 using device type X. (The starting position, device channel, and equipment are assumed to be contained in addresses 13 and 14 on the right screen display.) X=1-808, 2-6603/844, 3-81X/821, 4-85X/841, 5-86X. If X is absent, the device type described in address 13 is used. Refer to TD1 writeup for 13 and 14 parameters.
D (cr)	Loads and initiates the utility dump program DDD.
R (cr)	Redeadstart by returning PPUs to deadstart condition and execute the deadstart program displayed beginning with address 2. (A use of this feature is to deadstart from a different tape unit without changing the deadstart panel. Set 03W EOUU and then R cr.)
CP (cr)	Clears all PPU memories to zero, except PP0.
SP (cr)	Sets all PPU memories to ones, except PP0.
CC (cr)	Clears all central memory to zero.
SC (cr)	Sets all central memory ones.
AC (cr)	Sets address pattern in all of central memory.
* (cr)	Sets all PPUs to 2X speed (CYBER 170 only).
= (cr)	Toggle right screen display selection (deadstart area or status control words).

Site parameters describing system configuration (SYbb,IIII) and mass storage starting position, channel, unit, and equipment (DS13 and DS14) may be permanently edited onto the deadstart tape using edit routines EDT (SMM) or EDIT (SCOPE).

ENS MODE 2

Should the need for a direct load of a PPU 0 program without keyboard input arise, rewind the deadstart tape, set address 4 of the deadstart panel to 0302 and 5 to XX00 (where XX is the program number), and deadstart.

† The configuration and mass storage device parameters are saved in CM addresses 1 and 242, respectively when SMM is initiated so successive deadstarts will not require redefining these parameters. If this feature is not desired, set 03 = E7UU to eliminate this use of CM.

MEMORY DUMP ROUTINE (DDD)

DDD is called by using the CEL loader and typing D. DDD will be loaded and display the following messages.

P, C, X, I, F, or T

P	This selection allows the operator to dump any PP to the printer. LP CH (Type in channel number) LP NO (Type in equipment number)
C	This selection allows the operator to dump central memory to the printer. CM from (Type in starting CM address to be dumped) CM to (Type in last CM address to be dumped)
X	This selection allows the operator to exchange out CP0 or CP1 and dump the exchange package to the printer.
I	This selection is used only to fill the 512 image.
F	This selection is used only to fill the 580 image.
T	This selection allows the operator to dump central memory to tape 1 or load one file from tape 1 into central memory. (Number of banks selected must be in octal.)

All information required for any of these options is displayed on the 6000 console.

DDD resides in PPU-0 in low core starting at location 22B. If PPU-0 is dumped, low core has been clobbered by DDD. If low core must be saved, transfer PPU-0 memory to another PPU before loading DDD.

Example: To dump PPU-0 to PPU-1, set deadstart panel to:†

```
0001    2001
0002    0000
0003    7301
0004    0000
0005    0300
```

Then deadstart to transfer PPU-0 to PPU-1.

---

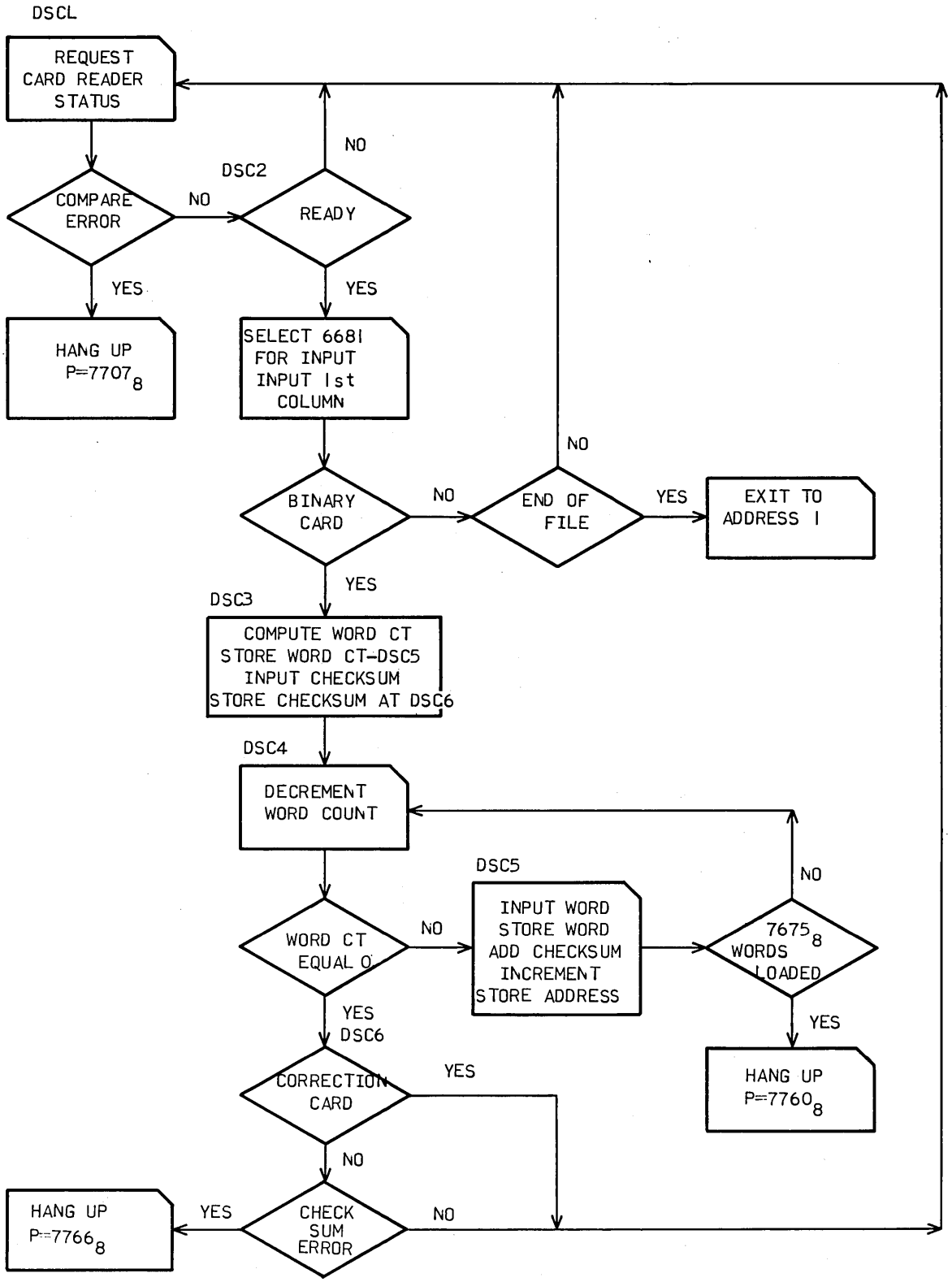
†On a CYBER 17X the PPM reconfiguration switches may be used to achieve this operation.

Reset the deadstart panel to load DDD and dump PPU-1.

DDD contains the option to add multiple access controllers if site configuration requires. The following example activates this option when assembled during a correct or update run of DDD.

```
*IDENT XXXX
*I DDD.15
DF.1015     EQU         1
*C DDD
```

DEADSTART CARD LOADER (DSCL)



DEADSTART CARD LOADER FROM READER TO DISC (PCL)

OPERATIONAL PROCEDURE

RESTRICTIONS

1. SMM must previously have been loaded on to the disc.
2. Words 13 and 14 of the deadstart panel must be equal to the parameters used to load the disc.
3. A PUNCHB deck of LDR must follow this card. (The deck cannot contain correction cards punched due to compare errors.)
4. An EOF (6789) card must follow the deck of LDR.

LOADING PROCEDURE

1. Ensure the card reader is not ready.
2. Set deadstart panel for card reader deadstart.

```
0001 - 75XX     XX = card reader channel
0002 - 77XX
0003 - E000     E  = card reader equipment
0004 - 77XX
0005 - 0001
0006 - 77XX
0007 - 1400
0010 - 74XX
0011 - 71XX
0012 - 7666
0013 - SSSS     S  = SMM disc parameters
0014 - PPPP     P  = SMM channel, unit, equipment
```

3. Ready the card reader.

PCL AND LDR DECKS

Run the following job to obtain PCL.

```
JOB,CM60000.
REQUEST(OLDPL)                    Assign SMM UPDATE tape.
UPDATE(Q)
COMPASS(I=COMPILE,B=PUNCHB)
7 8 9
*COMPILE,PCL
6 7 8 9
```

NOTE

Retain only one card (PCL).

Run the following job to obtain LDR.

```
JOB,CM60000.
REQUEST(OLDPL)                    Assign SMM UPDATE tape.
UPDATE(Q)
COMPASS(I=COMPILE,B=PUNCHB)
7 8 9
*COMPILE,LDR
6 7 8 9
```

NOTE

Take out all header cards and correction cards.

No error messages are produced by PCL due to its length.

1. PCL will stop on compare errors, etc.
2. Refer to the listing of PCL to define the reason for not continuing the load.

TAPE TO DISK (TD1)

OPERATIONAL PROCEDURE

RESTRICTIONS

1. The disk must be connected to a channel other than the tape controller.
2. CM must be reliable in order to build a directory.
3. Refer to CEL writeup and deadstart panel settings.

LOADING PROCEDURE

Specify the parameters under CEL, type P and CR. After deadstarting, hit any key to dump the tape on the disk.

PARAMETERS

Word 13 = ZZZX

Where X = 1 for 808
2 for 6603/844 †
3 for 813/814/824
4 for 853/854/841
5 for 861/863/865

Word 14 = YYUE. Enter channel number

Where ZZZ = Starting address to load SMM
Y = Channel number of the disk
U = Unit number
E = Equipment number

ERROR MESSAGES

In the event of an error, type any key to continue.

† 844 assembly by default. To assemble 6603, delete equate (EQU) for SP.844.
Example:

```
IDENT XXX
*DELETE SMMSP.12   Deletes SP.844 EQU, which is defined for 844 driver.
*COMPILE TD
```

ENDING MESSAGE

This gives the remaining available area of the disk (that is, the portion not used). This includes from the beginning of the disk until the first address that must not be used, and from the last address that must not be used up to the end of the disk. Also, displays deadstart settings for words 13 and 14. The right screen displays an optional deadstart setting for 6603's or 808's on channel zero. Pass 1 (808's) need not be inserted on the panel, unless the disc is repositioned.

SMM INITIALIZATION

SMM consists of a loader/monitor program (LDR), a central processor control program (CPC), a central memory resident, and a collection of programs. This section considers the SMM elements LDR, CPC, central memory, and DDD.

1. LDR loads and enters its initialization routine.
   a. Tape channel, equipment, and unit numbers are read from deadstart settings and plugged into I/O instructions.
   b. The test list is written in the upper 200 CM words.
   c. CM addresses 0 through 400 cleared.
   d. Tape parameters are written to CM address 241 and the loading device parameters are inserted in address 242.
   e. Number of PPUs are determined. Drop is written to PPU input registers that are not available.
   f. Computer type number of memory banks, number of CPUs, and number of PPUs are written into CM address 1.
   g. CPMTR is read from the SMM library to central memory, starting at address 250.
   h. CPC is read from SMM device and loaded into PPU 8.
   i. LDR exits to its monitor loop.
2. CPC enters its initialization routine.
   a. Computer type and number of memory banks are set in the left screen display.
   b. Field length and relative address are determined and set in exchange areas.
   c. If two CPUs were indicated by LDR in CM address 1, a flag is set, which controls CPU P register display, CPU control, and memory display.
   d. Upon recognizing a stand alone, cr will go to multiprogramming mode of operation. For CYBER 170, a status control register monitor (MTR) will be loaded to PPU 1.
   e. CPC exits to its main loop.

CENTRAL MEMORY

SMM uses the following central memory locations.

Address	Contents
0	Zero
1	59 47 35 23 11 0 BKS | SYS | TYPE | CPU | PPU BKS — Number of 4K CM banks (OCT) SYS TYPE — Type of system in display code CPU — Number of CPUs -1 PPU — Number of PPUs
2	53 35 29 11 0 FL | RA | CPU This is set by CPC/EXC/MAN/etc. and used by LDR when loading programs.
3	02000 00003 Loop CPU instruction.
4	Name of last CPU 0 program loaded.
5	59 47 35 23 11 0 PP Num | Accept | Address | Bits J display pointer.
6	Name of last CPU 1 program loaded.
7	EXC | DMP DMP running flag Nonzero when DMP is running EXC loading flag Nonzero when EXC is calling a CPU test

10

N M	E X	71C Z		PARMS

PP1 IR, Zero if PP is idle.

NME = Test name

CZ = Loading channel, cleared after test is loaded, then:

Z = Go/stop bit (1= go, 2=stop)

X = Abort flag (1= abort)

PARAMS, if ≠ 0 are stored in MCP+2

11

FTN	PARMS

PP1 output register used to send functions to LDR and MTR.

12 through 17 PP1's message buffer

20 through 27 PP2's area

30 through 77 PP3 through PP7's area

100

CP	C	AUTO	EXC	

101

FTN	PARMS

102

Program name when FTN=1

PP8/10 (CPC) IR

Output register
(byte 2=PP* if X.LP1, etc.)

AUTO=0 when auto mode
EXC≠0 when EXC, MAN, or BD1 is running

103 through 106 Message buffer

107

17	0
00 XX	ADR of MSG

DMP/CPU communication word

XX = 2 = CPU message

XX = 10 = dump dayfile to LP

110 through 237 PP9/11 through PP19/31 area

240

00 PP	

Display control word

PP = PP that has the display

241

	00CC	E0UU	MT	P

Magnetic tape parms, set by LDR

242

000Z	00CC	E0UU	0PPP	0000

SMM device parms

Z = Device code

First sector

First position/cylinder

244

FFFF	MMMM	MMMM	MMMM	MMMM

245

MMMM	MMMM	MMMM	MMMM	MMMM

F = Function code to MTR
M = Message buffer to MTR

246

00KK	KKKK	00LL	LLLL	00PP

K = Starting address of K display
L = Starting address of L display
P = PPU in which MTR is loaded

250 through 274 Location of CPMTR program

275

	XXXX

XXX ≠ 0 error CEJ occurred
Bit 4 = 1 signifies CPU 1
= 0 signifies CPU 0

276 Counter for number of times CPMTR entered

300 through 313

00PP	

Channel status table for channels 0 through 13
PP = PPU number using channel, zero if not used

320 through 333 Same as 300 through 313 for second set of PPUs

334

2XAA				BKSP

AA = Quantity added to exchange address to locate output exchange package after error CEJ

BKSP = No loop CEJ error if zero
Loop CEJ error if one

335

2000		EEEE	EEEE

2000 = SMM did not exchange if 0 SMM exchanged in
E = CPU 0 exchange package address

336

2XAA		BKSP

Same as 334, except for CPU 1

337

2000		EEEE	EEEE

Same as 335, except for CPU 1

340	Address of CPU 0 monitor exchange address
360	Address of CPU 1 monitor exchange address
400 through 577	CPU 0 exchange package area
600 through 777	CPU 1 exchange package area
1000 through 10777	Overlay area for PPU 1
11000 through 20777	Overlay area for PPU 2
21000 through 101777	Other PPU overlay areas in order
61000	Beginning address of CPC overlay
†XX5777	DMP recovery pointers

†XX is determined from memory size to put information in uppermost core.

†XX6000 through XX6777	DMP I display buffer
†XX7200 through XX7237	40-word buffer for ECS reads
†XX7270 through XX7275	6-word message buffer CPC to DMP
†XX7300 through XX7347	Buffer area for J display
†XX7400 through XX7600	Buffer area for H display
†XX7601 through XX7776	Message instructions for H display

†XX is determined from memory size to put information in uppermost core.

COMPONENTS

AUTO MULTIPROGRAMMING (AUTO)

Multiprogramming of peripheral tests starts unless AUTO is selected. [(Auto is selected by (CR) after initial load.)] At this time, the system display appears on the left screen. If the system has been identified as a CYBER 170, MTR will be loaded to PPU 1. Load tests in the usual manner (X.MNE.) or a test list (TLX.). All tests may be started or stopped and parameter entries made from the system display (CPC). The option also exists to give display control to individual PPs through an X.DIS or X."cr". If, for some reason, a PP hangs on a channel, the following entries are available to free the PP.

DCNX, FCNX, ACNX, MCHX, and IANX.

If these entries should fail to idle the PP, set RA to 0000 and clear the channel status table entry for the hung channel (for example, PP1 is hung on channel 6: type 306,0. and reset RA back to its original value).

Sequential addressing is available in CPC and works in the same manner as sequential addressing for PS (which is explained later).

The test list entry (TLX) X=1-100 contains preset lists of 3 through 5 tests. Select H display under CPC to display the contents of all available test lists.

USE OF IF STATEMENTS IN PS

In order to run a PP0 program and not load overlays or multiprogram, DF.PP0 must be defined before calling the PS routine.

```
Example:    DF.PP0    EQU    1
            *CALL,PS
```

In order to have the PS package load overlays, DF.OVL must be defined before calling the PS routine.

```
Example:    DF.OVL    EQU    1
            *CALL,PS
```

In order to have the PS package not idle all the PPUs, DF.NIDL must be defined before calling the PS routine.

```
Example:    DF.PP0      EQU    1
            DF.NIDL     EQU    1
            *CALL,PS
```

In order to have the PS package run under the multiprogramming scheme, DF.MLP must be defined before calling the PS routine.

```
Example:    DF.MLP      EQU    1
            *CALL,PS
```

In order to have the PS package load overlays and multiprogram, both DF.OVL and DF.MLP must be defined before calling PS.

```
Example:    DF.OVL      EQU    1
            DF.MLP      EQU    1
            *CALL,PS
```

This will allow SMM to use the program under multiprogramming by defining DF.MLP and calling PS.

Refer to appendix A for breakpoint option.

If DF.MLP is defined before calling PS, the following changes will take place.

1. PS originates the program at 1500 instead of 1000.
2. Sequential addressing is possible (with the use of a + key).

   Example: 1500 + 0002
   (after carriage return)
   1501 +

3. More than one message may be displayed on the left screen. The only requirement is that the last message sent must terminate with a zero.

```
Example:    DIS1    DATA    H*MNE RUNNING*
                    DATA    7300B,6100B
                    DATA    H*CXX EQXX UNXX*
                    DATA    7400B,6100B
                    DATA    H*SECTION 1*
                    DATA    0
            DIS2    LDC     DIS1
                    RJM     PS.CPMSG+1 (or PS.MSGS+1 or
                                        PS.MSG+1)
```

If DF.1015 is defined, PS will select the MAC (whose code is in 1505) in PS.RCH and deselect the MAC in PS.DCH. A shortened version of PS is assembled if DF.1015 is defined.

The 1015 select code must be in the following form.

1505 = 3X00 X = access code

PS.RMAC and PS.DMAC are also offered as routines to select and deselect the MAC.

Pool PPs I/O Registers

	PP1 - CM location	10,11	12-17	
	PP2 - CM location	20,21	22-27	
	PP3 - CM location	30,31	32-37	
	PP4 - CM location	40,41	42-47	
	PP5 - CM location	50,51	52-57	
	PP6 - CM location	60,61	62-67	
	PP7 - CM location	70,71	72-77	
CPC -	PP8 - CM location	100,101	102-107	- Not a pool PPU
	PP9 - CM location	110,111	112-117	
PP10 -	PP19 - CM location		120-237	

Input Register

Byte CM.XXX	1	2	3	4	5
	N A	M E	CHAN	Not Used	Parameter Entry

1	2
N A	M E

These two bytes contain the name of the test that the PPU is running.

Contains HUNG if the operator wishes to drop PPU (X.DROP.).

Contains 0000 if the program recognizes the operator (X.DROP.) function.

3

CHAN

Contains the IAM instruction that the idle routine needs to load the program into its PP (for example, 7101 for PPU1).

Contains the GO bit (0001 = GO).
Contains the STOP bit (0002 = STOP).
Contains the address of the parameter word (15XX).

4

Not Used

Not used in pool PPS. A 1 will prevent CPC from exchanging the CPU.

5

Parameter Entry

Contains the parameter entry word.

Output Register

Byte	1	2	3	4	5
CM.XXX	Function to LDR	Not Used	CHAN	PP Number	Not Used

1

Function to LDR

0004 - Tells LDR to load overlays into central memory.

PPU	Central Memory Location for Overlays
1	1000 - 11000
2	11000 - 21000
3	21000 - 31000
4	31000 - 41000
5	41000 - 51000
6	51000 - 61000
7	61000 - 71000
9	71000 - 101000
10 through 19	101000 - 211000

0005 - Tells LDR and CPC to deadstart.

0006 - Resets display timeout counter. Communicates with LDR while PPU is using the display so that LDR will not give display to CPC.

0007 - Gives display to PPU; used only by CPC.

0010 - Asks LDR to reserve a channel for PPU's use.

0011 - Asks LDR to drop a channel after PPU is through using it.

00XX- Asks MTR to execute a status/control register operation (refer to MTR documentation).

3

CHAN

Holds the channel number that the PPU wants when sending a function 10 or 11 to LDR.

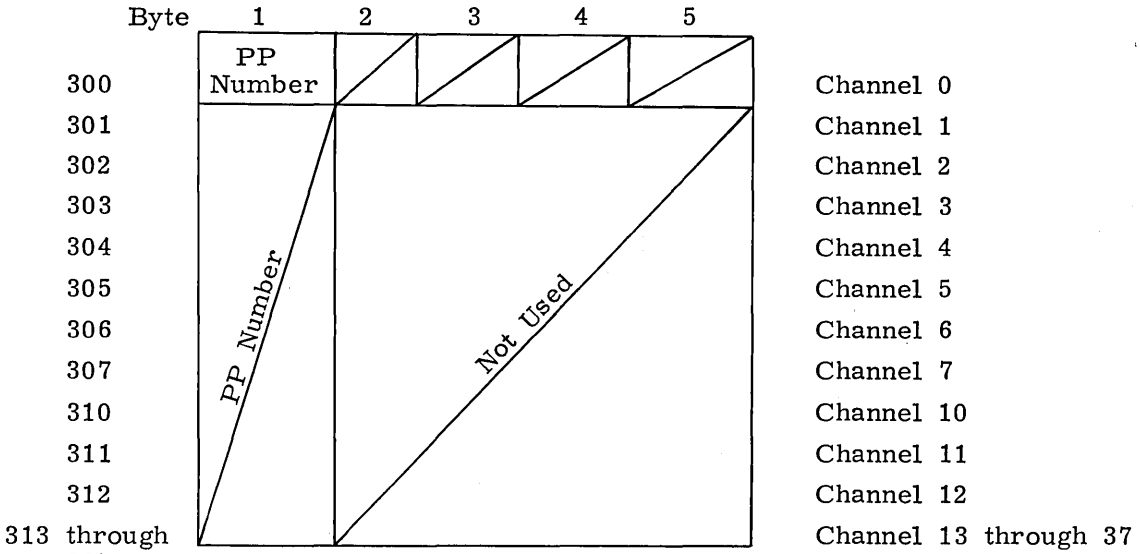
4

PP Number

Holds the PPU number of the requesting PPU when sending a function 7 to LDR.

Channel Status Table

Central Memory Location



Byte 1

PP Number

Holds the number of the PPU that is using the channel and contains a zero when no PPU is using that channel.

Central memory location 240 (byte 1) contains a 0010 if CPC has the display. If CPC does not have the display, it contains the number of the PPU that has the display.

CPC FUNCTIONS

(CR) — Sends an idle routine to all available PPUs (normally 1 through 7, 9) and sets G display.

X.
or
X. DIS. — Gives display to PPX by sending a function -7 to LDR.

X. D
or
X. DROP. — If performed before typing AUTO., it will hang PPX on an 0300 instruction and no tests may be loaded into that PPU. (This allows a CE to bypass bad PPUs.)

If performed after typing AUTO, it will send a HUNG to PPX's input register; whereupon PPX will idle itself and look for a test to be loaded.

X.MNE.	If performed before typing AUTO, it will load any test and give control to the test. If X.MNE, CCUE. is used, CCUE will be stored in location 1502 of the PPU. If performed after typing AUTO, it may only be used to load CPU tests and specific PPU tests which will multiprogram. PPU tests will be loaded into the first available PPU signified by IDLE on the G display, unless X = the requested load PP number.
X.G or X.GO.	Starts PPX running when it has stopped at a call to PS.MSGS. and is waiting for the display.
X.S or X.STOP.	Stops PPX when the program goes to PS to display a message.
X.*	Assigns the * key to PPU X.
X.YYYY, ZZZZ cr	Enters PP MEMORY address YYYY with data ZZZZ. The COMMA can be replaced with a PLUS for sequence storing. This is used mainly to change parameters in a SUPER P interfaced program. NOTE This can be used for normal PS interfaced SMM programs as long as ZZZZ ≠ to 0.
X.RUYYY cr	Starts a SUPER P interfaced program (which can be identified by a / following the program name after loading) at PPU location YYY.
IANX.	Inputs to A on channel X. (The word which is input is displayed to the right of CPC.)
ACNX.	Activates channel X.
DCNX.	Disconnects channel X.
FCNX.(,Y.)	Sends a release function out on channel X or function Y.
MCHX.	Master clears the 6681 on channel X.
TLX.	Test list which loads from 3 through 5 tests sequentially into PPs. The tests must be in the order in which they appear on the tape unless a disc load is used. Select H display.
GO.	Starts all PPs running when they have stopped at a call to PS.MSGS.

DROP. Idles all running PPs.

/ key Causes DMP, if running, to dump its accumulated buffer to the printer.

M,zzzz cr
or
X.M,zzzz cr

Sends the CPC keyboard buffer to DMP which will enter it in the I display and its central buffer to be printed with any test messages.

NOTE

This is an excellent way to head error messages with date/time, special conditions (margins or delay probes) or parameter settings.

J=X,zzzz cr Assigns the J display to PPU X. If the PPU responds, a PPU memory display will appear (if the J display is displayed upon either screen). ZZZZ is the memory address to be displayed (ZZZZ optional).

NOTE

At this time only D44 is available to the J display.

Jy,zz00 cr Sets the areas of PPU memory to be displayed for the J display.

y=0	Upper memory block
y=1	Middle memory block
y=2	Lower memory block
y=3	or
y=4	All memory blocks with a 100b increment
zz	PP memory address to be displayed

D-y cr Rolls the I display dayfile back (circular) to enable the operator to see previously entered error messages. Y is the number of lines to roll (1 through 17) each time it is executed.

G DISPLAY SYSTEM STATUS

```
                        CHANNELS DEDD DEDD EEEE
                                 -2--  -6--  1345

PP
0           LDR
1           DF8          PARITY G004 T004 S000
2           DF1          ALL GROUPS TEST
3           LPT          SEC 4 RUNNING
4           DMP          DUMPING DAYFILE TO CM
5           MTX          SEC 3 RUNNING
6           CP1          SEC 1 RUNNING
7           DF4          RUNNING SEC 1
10          CPC
11          IDLE
          2. DIS.
```

SMM 6000 (AUTO) MODE OF OPERATION

INITIAL LOAD†

CPC sends an idle loop program to all available PPUs 1 through 7, 9 through 19.

CPC sets G display on left screen and returns to its display loop.

PPU 1 THROUGH 7 AND 9 THROUGH 19 IDLE LOOP

Each PP reads its input register in central memory looking for a nonzero entry in byte 0.

If byte 0 is nonzero, byte 2 is checked for an IAM instruction and stored into the PP's read routine. The PP then hangs on an input of 10000_8 words over channel 10.

† CR brings up auto.

Loading PP 1 through 7 and 9 through 19

1. CPC senses a test load entry has been made and sends a call to LDR to load the test.
2. LDR searches the directory for the test desired, and when found, it tells CPC to stop using the display channel.
3. LDR sets the first available PP's input register equal to the name of the test that is to be loaded and inserts a 7110_8 instruction in byte 2.
4. The test program is now sent to the PP over channel 10.
5. When the program has been completely loaded into the PP, LDR sends the PP number that the test has been loaded into over channel 10.
6. LDR now disconnects channel 10, which starts the PP running and tells CPC to use the display channel 10.
7. LDR then returns to its monitor loop.

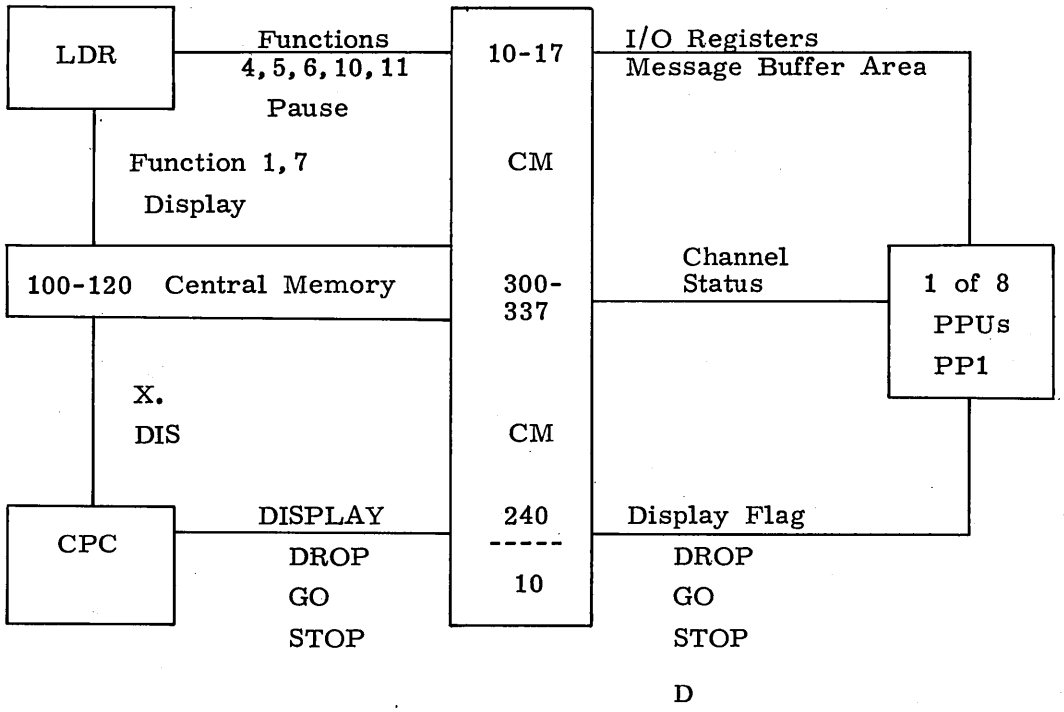
PPU 1 through 7 and 9 through 19 Entry to Test Program

1. The PP reads the last word sent by LDR and stores it as its PP number.
2. The PP checks its input register for any parameters entered at load time. If so, it stores them in location 1502_8 (normally the channel and equipment parameter location).
3. The PP now enters the program at the starting address (normally location 1512_8).

PPU 1 through 7 and 9 through 19 Pause Loop (in PS Resident)

1. The test program running in the PP returns to the pause loop in PS periodically to maintain communication with SMM (auto) system.
2. The PP reads its input register and checks bit 0 of byte 1.
3. If bit 0 is clear, operation is returned to the program.
4. If bit 0 is set, the PP will clear all of its registers and reenter its idle loop.

BLOCK DIAGRAM OF AUTO (MLP)



REAL-TIME DISPLAY AND TIMING CLOCK FOR 6000 SMM (CLK)

PURPOSE

The intent of this routine is to provide a digital clock that may be sampled in order to compute run times of diagnostic software and related hardware operations. It can also be used as a source for a console display of the current real time.

REQUIREMENTS

HARDWARE

1. One 6000 PPU.
2. One 6000 real-time channel clock with a 4.096-millisecond period.
3. Equipment required by 6000 SMM to load any diagnostic.

SOFTWARE

This program will execute under SMM 6000's multiprocessing scheme (that is, MLP or auto).

ACCESSORIES

None required.

OPERATIONAL PROCEDURE

LOADING PROCEDURE

This program is called as CLK under the 6000 SMM system called auto. (Refer to SMM systems.)

PARAMETERS

There are no parameters. If a T is typed under the PPU display, the following message is displayed.

```
TIME.  00.00.00.
```

The desired time can then be entered. A backspace will cause the last digit entered to be zeroed out. A carriage return enters the desired value into the clock.

SECTION DESCRIPTION INDEX

Not applicable.

OPERATOR COMMUNICATION

MESSAGE FORMATS

Under the auto G display the message format is:

```
11  CLK  *00.00.00.
```

Where	11	PPU number that CLK is running into
	*	Blank if the time has been set by the operator or
		* to indicate the time since CLK was loaded.

In the PPU registers in central memory (PPIR = PPU number *100B) under the E display:

CM Location	1	2	3	4	5	Display Code Values
PPIR+0	0314	1300	0000	0000	0000	CLK
PPIR+1	0000	0000	0000	0000	0000	
PPIR+2	4733	3357	3333	5733	3357	*00.00.00.
PPIR+3	ZERO	SCND	MLSC	MCSC	CHCK	ABCDEFGH
PPIR+4	CLPH	MSC1	MSC2	0000	0000	IJKLMN
PPIR+5	0000	0000	0000	0000	0000	
PPIR+6	0000	0000	0000	0000	0000	
PPIR+7	0000	0000	0000	0000	0000	

Where PPIR CLK's PPU number times 100B.

ZERO Data zero which terminates the display clock message when displayed by SMM.

SCND A 12-bit octal second counter (SEC).

MLSC A 12-bit octal millisecond counter which is reset to zero when it reaches 1750B (that is, 1750B milliseconds = 1000 milliseconds = 1 second) (MSC).

MCSC A 12-bit octal microsecond counter which indicates the time in microseconds, since the clock was last sampled (USC).

CHCK The current value of the channel 14B real-time clock.

CLPH The current phase of the real-time clock that CLK is writing to indicate that 1 millisecond has passed.

MSC1 Upper 12-bits of a 24-bit millisecond counter.

MSC2 Lower 12-bits of a 24-bit millisecond counter.

The above information should be helpful to a programmer wishing to time an operation without requiring a PPU to go into a timing loop. Such loops can cause a communication gap that may be undesirable under system operations.

MESSAGE DICTIONARY

Not applicable.

DESCRIPTION

This routine samples the clock about once every 45 microseconds. At each sample, a check is made to see if the clock's phase has changed (that is, the upper two bits of the

real-time clock. When the clock phase changes, a millisecond has elapsed and all millisecond counters are updated. After 1000 milliseconds has elapsed, the second counter is incremented. This same procedure is used for minutes and hours. After 24 hours the clock returns to zero. A more detailed description may be obtained by studying the listing.

CENTRAL MEMORY CONFLICT PROGRAM (CMC)

OPERATIONAL PROCEDURE

RESTRICTIONS

1. This program should be run under the SMM auto mode of operation, along with EXC or individual CPU or CM tests or the SCOPE system.
2. No RA value is added to the CM address specified in the keyboard entry (under SMM). Refer to parameters for RA under SCOPE operation.

LOADING PROCEDURE

1. Called as X.CMC. or X.CMC,2. A 2 will set parameter location 1500 to stop on compare errors. Refer to SCOPE operation for online loading parameters.
2. May be called into as many PPUs as the system allows (under auto).

PARAMETERS AND/OR ENTRIES

Location 1500 = 0 - Do not stop on compare errors
= 2 - Stop on compare errors
= Sense switch 1 for SCOPE operation.

Entries

The left screen displays entries under SMM.

D	Release display to SMM system.
S	Stop read/write.
SPACE	Continue read/write.
(CR)	Restart and display entries.
T,X.	Set delay between reads/writes = X (1-7777). Preset to 1.
R,X,Y.	Read X(1-1000) words from central memory location Y into PP buffer at location 3000.
W,X,Y.	Write X(1-1000) words to central memory location Y from PP buffer at location 3000.

RW,X,Y.	Read X (1 through 1000) words from central memory location into PP at location 3000 and write X words to central memory location Y from PP location 3000.
WR,X,Y.	Write X (1 through 400) words from PP location 5400 to central memory location Y and read X words to PP location 3000 from central memory location Y, then compare data read to data written and display any compare errors.

SCOPE Operation

Load call = X.CMC,ABNNNN,FFFFFF.

Parameters for A, B, N, and F.

A = 1 through 7 to set time between reads and writes

	(octal)	(binary)		
B =	0	(X00)	-	Read only
=	1	(X01)	-	Write only
=	2	(X10)	-	Read/write no compare
=	3	(X11)	-	Write/read with compare
=	4	(1XX)	-	Add RA to F and request FL from system

N = Number of words to read or write

F = Central memory address to read or write (refer to B for RA parameters)

MESSAGES

NORMAL

READING CM

Read CM only.

WRITING CM

Write CM only.

READ/WRITE CM

Read, write, and no compare.

WRITE/READ/COMPARE CM

Write, read, and compare data for errors.

MAKE ENTRY TO READ/WRITE CM

Assign display and make desired entries.

ERROR

EXP - VVVVWWWWXXXXYYYYZZZZ
REC - VVVVWWWWXXXXYYYYZZZZ
COMPARE ERR ADDR CCCCCC

C = CM address where compare error occurred.
V-Z = The data expected (EXP) and the data received (REC-failing data).

NOTE

Data is compared one 12-bit word at a time until five (12-bit) words have been checked. Only those 12-bit words that actually do not compare are displayed as numbers. The 12-bit words that do compare are displayed as V, W, X, Y, or Z, depending upon the byte.

Under SMM, locations 0 through 1000 are protected on a write central memory. Illegal entries are ignored.

CENTRAL PROCESSOR CONTROL (CPC)†

OPERATIONAL MODES

CPC provides displays of CPU P register(s), CPU breakpoint address, channel status, CPU and PPU program names, and exchange package and central memory displays. CPC controls starting and stopping of CPU(s) and provides test mode for troubleshooting.

CPC calls one overlay to CM address 61000 absolute and, upon going into auto, down-loads it into part of PP 10 memory. This gives CPC the following messages and modes of operation.

NOT AUTO	The mode that CPC originally comes up in with the PPUs waiting with their channels active and empty.
AUTO	The mode that CPC goes into after a CR. All PPUs are idled and the G and J displays are brought up.
WRONG	Illegal entry.
NO OVL	Notifies user that the CPC overlay 1CP was not loaded from CM when going to auto. Program loading and execution will continue normally, but any command that is contained within the overlayed area will not execute.
LOC 0 NOT 0	Notifies user that absolute location zero (0) of CM is nonzero. The notation DS stands for deadstart because the condition is irrevocable from CPC.
ERROR CEJ 01	Notifies user that CPMTR has been entered with an exchange and P from the exchanged out program was 1 or 0. 1 indicates CPU 0.

KEYBOARD ENTRIES

1. Setting Optional Displays

   Enter: XY. (cr)

   (CPC) Sets X display on left screen
   Y display on right screen

   Choices for X, Y = A, B, C, D, E, F, G, H, I, J, K, L

†Refer to Auto.

2. Setting Memory Display Areas

   Enter: Xn, A. (cr)

   CPC sets field n of display X to display 10 words of memory from address A.

   X may be C, D, or E (refer to section 2).
   n may be 0, 1, 2, 3, 4
   n = four sets all fields

3. Enter Memory

   Enter: A, V. (cr)

   CPC stores value V in address A.

   Enter: LA, V. (cr) to left justify entry

   Enter: A+V. for sequential storing of data

NOTE

The relative address RA is first added to A.

4. Enter CPU Register

   Enter: Rn, V. (cr)

   CPC stops CPU if running and enters value V into register Rn. (sets exchange package). R may be A, B, X, RA, FL, RE, FE, or MA, and n may be 0 through 7 for A, B, or X, while others do not require n. When RA is used, the FL value is automatically updated.

5. Enter Breakpoint Address

   Enter: BK , V. (cr)

   CPC sets breakpoint address to value V. That is, instructions at address V are saved and replaced with program stop.

6. Set CPU Control Mode

   Enter: TEST. (cr)

   CPC sets up test mode where the P register is continually monitored for breakpoint address. When breakpoint address is reached, the CPU is exchanged out and exchanged back in with the following.

   a. If A display is selected, exchange-in is done with input package.

   b. If B display is selected, exchange-in is done with last output package.

Enter: RUN. (cr)

CPC sets up RUN mode where the CPU runs until breakpoint is reached. If breakpoint is reached, the CPU is exchanged out and breakpoint address is restored.

7. Set New Exchange Address

   Enter: EX, V. (cr)

   CPC sets exchange address to value V. RA and FL control in CPC are read from the new area.

   NOTE

   The relative address RA is not added to V.

8. Start Central Processor (Refer to CPMTR)

   Enter: Spacebar

   CPC exchanges central processor with the following.

   a. If B display is not selected, exchange is done with input package.

   b. If B display is selected, exchange-in is done with last output package.

9. Stop Central Processor (Refer to CPMTR)

   Enter: Backspace key

   CPC exchanges central processor with the following.

   a. If B display is not selected, exchange is done with input package with P and RA cleared.

   b. If B display is selected, exchange is done with last output package with P and RA cleared.

10. Clear Pause Bit and Start all PPUs Running

    Enter: GO. (cr)

    CPC clears the lower byte in CM address 100 and sets all PPUs running (under auto).

11. Change CPU Control (Dual CPUs only)

    Enter: Equal Key

    CPC sets displays and keyboard controls (RCP, DCP, TEST, RUN) for CPU-n.

    Enter: Equal key (=) to toggle back and forth between CPU-0 and CPU-1 control.

12. Load Program

    Enter: X.MNE. (cr) or X.MNE,CCUE. Enter channel and equipment to loaded PP or X.MNE,Z, X=PP to load under auto.

    CPC issues functions 0001 to LDR to load program MNE. Also, pause bit is cleared.

13. Transfer Central Memory to PPU

    Enter: MTP,X. (cr)

    CPC issues function 0002 to (LDR) to transfer program at CM address (RA+X+1) to the next available PPU. Address X upper byte must be nonzero.

14. Drop All PPUs under auto.

    Enter: DROP. (cr) (All PPUs that are running will be idled.)

15. Drop PPU under auto.

    Enter: X.DROP. (cr) Idles PPU X.

16. Memory Byte Entry (Dependent upon which memory display is selected)

    Example: E display up.

    CM. 100 = 7777 6666 5555 4444 3333

    Entry: 100,5,1234. will change the last byte (3333) to 1234.

    Example: C or D display up.

    CM 100 = 77777 66666 55555 44444

    Entry: 100,4,12345. will change the last byte (44444) to 12345.

17. X Register Byte Entry

    Example: X1 = 7777 6666 5555 4444 3333

    Entry: X1,5,1234. will change the last byte (3333) to 1234.

18. Set Memory

    Entry: SET,100,200,XXXX. will set all bytes to XXXX from location 100 up to and including location 200. If X is five characters long, four bytes will be set instead of five.

19. CM. Cr clears central memory from 400 - protecting ra, fl, and ma for CPU 0 (in nonauto mode only).

20. Scan Memory

    Example: + entry moves all displays 40 locations forward and
    - entry moves all displays 40 locations backwards.

21. (cr) - initial (cr) will bring in auto multiprogramming.

22. ' Key as the first entry will abort the loading of a program from tape.

23. X.* will assign display to PPU X. Any later * command in auto mode will perform DIS function to assigned PPU.

24. (/) Slash entry will cause dayfile program (DMP) if running to dump accumulated buffer to the printer.

25. Right blank key will do a repeat entry.

26. CBPX (cr) - clears the CPMTR cabability to loop on error CEJ conditions, X = CPU number.

27. SBPX (cr) - enables CPMTR to loop on an error CEJ condition, X = CPU number. A use program, monitor flag clear, exiting with any MODE error will be restarted using its INPUT exchange package.

28. IR,xx (cr) - enters the interlock register maintenance bits on CYBER 70.

    Bit 0 = phase 1, bit 1 = phase 2, bit 2 = phase 3, bit 3 = phase 4
    To advance a phase, delay all others.

    EX. IR,15 (cr) advances phase 2

    EX. IR,4 (cr) delays phase 3

NOTE

This command is valid in stand-alone mode only. Use DLY in auto mode.

29. Right parenthesis key,), enables and disables:

    a. A two-octal byte interlock register maintenance bit display to the right of the channel display (on CYBER 70 only).

    b. The CPMTR idle instruction at central memory location 3.

NOTE

The maintenance bit display must be off when running the test IRT in auto mode.

30. Refer to the supplemental edition to this manual (publication no. 60409500) for commands to CYBER 170 interface MTR.

31. X.YYY,ZZZZ CR enters PP memory address YYYY with data ZZZZ. The comma can be replaced with a plus for sequence storing. This is used mainly to change parameters in a super P interfaced program.

NOTE

This can be used for normal PS interfaced SMM programs as long as ZZZZ ≠ to 0.

32. X.RUYYY CR is used to start a super P interfaced program (which can be identified by a / following the program name after loading) at PPU location YYY.

33. M,ZZZZ CR or X.M,ZZZZ CR will send the CPC keyboard buffer to DMP which will enter it in the I display and its central buffer to be printed with any test messages.

NOTE

This is an excellant way to head error messages with date/time, special conditions, margins or delay probes, or parameter settings.

34. J=X,ZZZZ CR assigns the J display to PPU X. If the PPU responds, a PPU memory display will appear (if the J display is displayed upon either screen). ZZZZ is the memory address to be displayed (ZZZZ optional).

NOTE

At this time, only D44 is available to the J display.

35. JY,ZZ00 CR sets the areas of PPU memory to be displayed for the I display.

To be displayed for the I display.

Y=0	Upper memory black
Y=1	Middle memory block
Y=2	Lower memory block
Y=3	Or
Y=4	All memory blocks with a 100B increment
ZZ	PP memory address to be displayed

DISPLAYS

NORMAL DISPLAYS

CPC displays title line, LDR message line, status of I/O channels, and keyboard buffer on left screen at all times. On the right screen, CPU P register and mode and breakpoint address of selected CPU are displayed at all times. Also, on each screen is one of the following optional displays.

OPTIONAL DISPLAYS

A Display

Both input and output exchange packages are displayed when CPU is stopped. Only input package is displayed when CPU is running.

B Display

Input exchange package.

E Display

Four fields of 10 CM words each are displayed. Words are displayed in five groups of 12-bit bytes. Displayed code conversions appear opposite octals.

D Display

Same format as C display, except locations displayed are incremented by 40_8.

C Display

Four fields of 10 CM words each are displayed. Words are arranged in four groups of 15-bit bytes.

F Display

Fake display, only lines described under normal displays appear on screen selected to display F.

G Display

Auto system display for multiprogramming.

H Display

Test list display. H display selected will display TL1-TL100

I Display

CPU message display. CPU REQUESTS I DISPLAY will be displayed.

J Display

J=0 displays CPC commands.

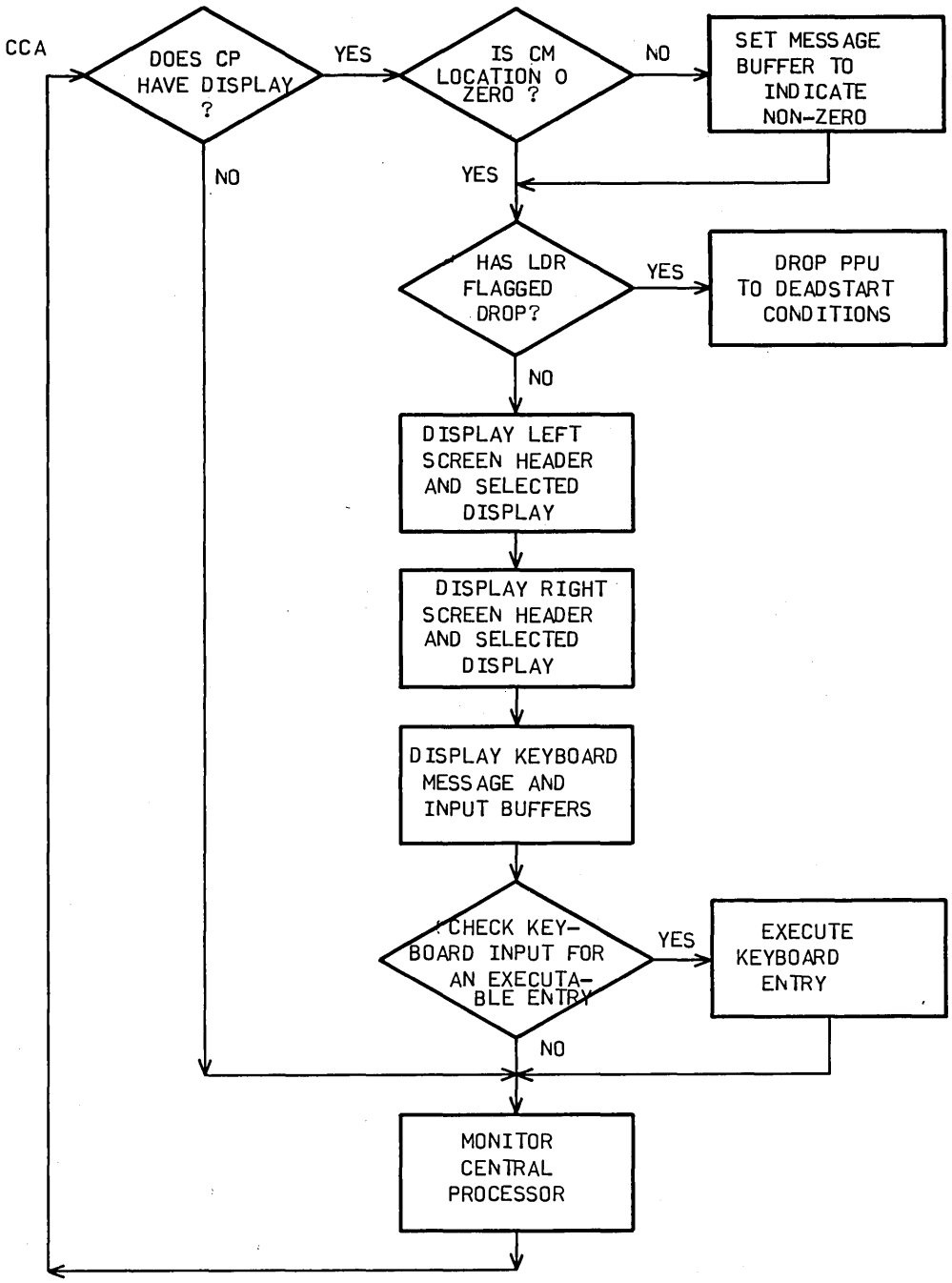
J=X allows a PPU memory display for D44-type programs or a PPU driven CPC display such as the command display provided with J=0.

K Display

The K display is used on CYBER 170 in conjunction with MTR to display PPU memory, channel parity, transmission parity, and CSU SECDED errors.

L Display

The L display is used on CYBER 170 in conjunction with MTR to display CPU status, status control register, and SECDED error history.


CCA
DOES CP HAVE DISPLAY ?
YES
IS CM LOCATION 0 ZERO ?
NO
SET MESSAGE BUFFER TO INDICATE NON-ZERO
NO
YES
HAS LDR FLAGGED DROP?
YES
DROP PPU TO DEADSTART CONDITIONS
NO
DISPLAY LEFT SCREEN HEADER AND SELECTED DISPLAY
DISPLAY RIGHT SCREEN HEADER AND SELECTED DISPLAY
DISPLAY KEYBOARD MESSAGE AND INPUT BUFFERS
CHECK KEY-BOARD INPUT FOR AN EXECUTA-BLE ENTRY
YES
EXECUTE KEYBOARD ENTRY
NO
MONITOR CENTRAL PROCESSOR

CENTRAL PROCESSOR MONITOR (CPMTR)

PURPOSE

Provides a program to do the following.

1. Get the CPU out of monitor mode after execution of CPU -XJ instruction or PP MAX or MAN if P is not 0 or 1 on the entry to CPMTR.
2. Handle error CEJ:
   a. Move exchanged-out program to output exchange package area.
   b. Set up either an idle exchange package or A.
   c. Reinitiate a program that error CEJ exists (that is, exchange to rerun the program, setting up a loop on error CEJ condition).
   d. Clear the monitor flag by an XJ to BJ.
   e. Signal CPC that an error CEJ has occurred.

In addition, provides a means of starting and stopping a CPU and using an MXN/MAN exchange as an installation parameter as opposed to EXN exchange control from CPC. This simulates operating system operation more easily, because all CYBER systems and many 6000 systems use MNX/MAN operating systems. The 742X CYBER systems have a defined hardware problem where an EXN to one CPU, interrupting the other CPU which is executing a CEJ or ECS instruction, will hang chassis 5/9 requiring a dead-start to clear.

NOTE

CPMTR will remain the same for all three options; only the code in CPC will change.

Also provides CPC with an ECS display capability.

ENTRY

CPMTR can be entered by an MXN, MAN, XJ, or error CEJ CPU0 MA=340, CPU1 MA=360.

A CPU program may change MA by setting up a call to CPMTR in its RA+1, and exchanging into monitor state with B1 = its RA. The following format for RA+1 is used.

Upper 24 bits = new MA
Lower 18 bits = CMA

Upon reentry to program, A6, B6 will be destroyed.

COMMUNICATION

SIGNAL	Signals error CEJ to CPC
COUNTER	Counter that tells how many times CPMTR has been entered
BKP	Should be set to nonzero to cause a program that exited CPU0 with an error CEJ to be reinitiated
EXCHADRA	CPU0 input exchange package address
EXCHADRB	CPU1 input exchange package address

USAGE

HANDLE ERROR CEJ

If a program should encounter any of a number of error conditions on a CYBER 70 mainframe with the CEJ siwtch enabled (and no disabling ground caps or wires), an error sequence is started. First, the error mode and address at which it occurred are stored in RA, and second, a CEJ is performed. This CEJ, which is a result of any error exit condition, will be designed as an error CEJ. This error CEJ uses MA for the exchange address and sets the monitor flag. The monitor exchange package resides at the address specified by MA when the monitor flag is clear. In CPMTR, the P register from the exiting user program is tested for 0 or 1 (0 = 74XX exit, 1 = 73XX). If either of these conditions is true, CPMTR will move the user program output exchange package (which replaced the monitor package in memory at MA when we exchanged) to the user output exchange package area (exchange address + 20, normally 420 for CPU0). CPMTR then creates an idle exchange package in the monitor exchange area and CEJs (B J + K) to it. This puts the CPU into user mode (monitor flag clear) and executes an idle program. Coincidently, the monitor package is moved from the CPU back to its monitor exchange package area in central memory. At this point, CPMTR signals CPC of the error CEJ that it has processed. CPC then clears the CPU running flag in the PP and issues the message ERR CEJ XY, where x=0 indicates CPU0 and Y= 1 indicates CPU1 (both could possibly be up concurrently).

NOTE

MA is preset to 340 for CPU0 and 360 for CPU1.

HANDLE CONFLICT CEJ, MXN, AND MAN

If a CE wishes to add conflicts to an executing CPU program in the form of a CEJ, he simply inserts the CEJ anywhere in the main line of the program. The CE or user program on encountering the CEJ will exchange to MA and follow much the same path as stated in error CEJ, except the user P register will not be equal to 0 or 1. To clear the monitor flag and restart the user program, CPMTR executes a CEJ to B J + K. This initiates the user program at the execution address +1 of the conflict CEJ.

DAYFILE AND MEMORY DUMP (DMP)

INTRODUCTION

This is a line printer dump routine used to dump PPs and central memory without the need to deadstart DDD. Also, dumps the SMM (auto) dayfile.

Runs under SMM (auto) mode.

OPERATIONAL PROCEDURE

RESTRICTIONS

1. Runs under SMM (auto) mode only.
2. The PPU that is to be dumped must not be in a hung state; may be idle.
3. PP0 and PP8 cannot be dumped.
4. Cannot dump PP if printer channel = communication channel.
5. Illegal to run two copies of DUMP. Dropping one copy will clear interface flag with PS package which disables remaining copy.

LOADING PROCEDURE

Called into a PP by X.DMP. Must be called into a PP number less than EXCs PP number.

PARAMETERS

All parameters are displayed on the console. Type X.DIS. to look at parameters.

MESSAGES

DUMPING CENTRAL MEMORY

While dumping CM.

DUMPING PPX

While dumping PPX.

PPX CANNOT BE DUMPED

The PP is not communicating with the SMM system. Restart DMP. Type R or try and free the PPU through channel entries from CPC.

LP NOT READY

The line printer is not ready. Check paper-out condition.

SET LOC 1502 = LP(CCEE)

Displayed at beginning of dump routine. Parameters may be changed at completion of any dump.

DUMPING DAYFILE TO CM

Displayed during monitor of the dayfile.

TYPE CF--F, L--L, R--R. TO DMP F TO L RELATIVE TO R

Dump CM from address (F--F) to address (L--L) relative to address (R--R)

SET LOC 1500 = 2000 - DO NOT DUMP DAYFILE TO I DISP.

SET LOC 1500 = 4000 - DO NOT DUMP DAYFILE TO CM

DAYFILE RQ I DISPLAY

NOTE

Type GO. if any hangups occur because of paper out, not ready, etc.

I DISPLAY RESTRICTIONS

The I display, under auto (generated by DMP) can be destroyed by any test call which loads to central memory locations 20000_8 through 20777_8.

Examples:		PP1	DMP	(W/I display)
		PP2	EXC	
	or	PP1	DMP	(W/I display)
		PP2	MM4	
	or	Loading a CPU test (CU3, CM6, etc) with an RA < 21000_8		

6400/6600 MULTIPROCESSING ROUTINE (EXC)

OPERATIONAL PROCEDURE

RESTRICTIONS

EXC must not be loaded until after AUTO is typed in. Only those CPU tests that are displayed on the left screen can be run. At least two CPU programs must be selected. EXC may only be running in one PP (except 65/6700). Requires a minimum FL of 31500 to load. No overlay program may be loaded into a PP higher than EXC's PP number.

PARAMETERS

Exchange Rate

Displayed on the lower right-hand screen, can be from 0000 through 7777. Entered by an EX,YYYY. entry, where YYYY = exchange rate. This number is loaded into A register and counted down to zero by a subtract 1 from A instruction.† When A is zero, an exchange is done on one of the CPU programs.

Preset to 20_8 (50 milliseconds between exchanges).

Field Length

To change, type CFL,Y,XXX.Y = tests A through T.

Test List

The CPU programs that can be run are displayed on the left screen, with the field length necessary to run them. The programs that are active have an * beside their name. The test list is preset to CT3, CU2, ALS, and FST. To change type in TL, MNE, MNE, MNE, MNE. (The entry can be 2, 3, or 4 programs long and more than one test may be reselected.)

†On CYBER 17X's use twice the value when running at 2X speed.

CPU1

If two EXCs are loaded on a dual CPU system, the second one will be set up to run CPU1. EXC is changed to EXC1 for CPU1. A GO. entry must not be used. The second EXC must be loaded into a PPU with a lower PPU number.

Rotate List

The rotate list option is activated by typing an =. This starts a sequencer which loads four CPU tests, runs them for approximately 30 seconds, then loads four new tests. There are five sets of four tests each. To deactivate the rotate list, type a /. This option may be selected or deselected at any time. The number of tests loaded is dependent upon the available core size.

The rotate list will select appropriate tests for the system type it is being run on. It also picks a random selection of tests. The + key adds ECM to the rotate list, and - key deselects CT3 from list.

Procedure for Isolating and Controlling a Failing CPU Diagnostic under Auto

When a CPU diagnostic fails at a legitimate error stop address, bring up DIS on the EXC control point and obtain the failing program EXCH package address and RA. Stop EXC with S under DIS. or X. STOP (CR). EX, addr at the package address (addr) of the failing program. The A display now contains the failing program input and output exchange packages. The memory displays will be referenced to the input package RA.

MESSAGES

1. CPUXRAXXXXXXFLXXXXXX

   Give the PPU an X. GO or do an N. DIS. and change the exchange rate and test list.

2. MNE = YYYYYY MNE = ZZZZZZ MNE = XXXXXX MNE=WWWWWW

   Display of program addresses of running programs.

   NOTE

   The = sign will be replaced with an * when a program has a message to be displayed.

3. EXC now monitors the area in RA where P will be stored if an error exit is taken. If this area goes nonzero, the message ERR CEJ REF EXCH ADDR will appear and EXC will stop. This indicates that on a CYBER with ERROR CEJ enabled, the exchange package for the program that encountered the error condition will be stored at the exchange address selected. The CE would then check RA in that package, and compare it with the RAS for the programs running in EXC to find out which program failed.

   EXC now stops after encountering a message from any of the programs it is running with the message PROG XX MSG MMMMMMMM, where xx is the program number and MMMMMMMM is the message.

DESCRIPTION

The test is designed to allow a faster exchange rate for multiprocessing CPU programs than is available on the operating system. Running four programs with an exchange rate of 0 will cause an exchange jump instruction to be issued approximately every 2.5 microseconds (+ time needed for the exchange). The program address and name of each program will be displayed in the MSG buffer of the PP that EXC is running in. The input exchange package for program 1 is at absolute CM 400 for CPU0 and 600 for CPU1.

The input package for program 2 is at 440_8, the input package for program 3 is at 500_8, and the input package for program 4 is at 540_8. If EXC is stopped, the output packages will be in sequence also (420, 460, 520, 560). When EXC is displaying, S stops tests and spacebar starts them. D returns to CPC and F removes most of display.

NOTE

The input and output packages change from the above locations when the programs are running. If the programs are stopped (X.STOP or S), the input and output packages will be in the sequence described above.

Type F to shorten the display time. Another F will bring back the original display.

LOADER/MONITOR (LDR)

LDR monitors PPU output registers for call codes for LDR action. LDR performs the following operations in response to the call codes.

Call Code	LDR Action
0001	Load program.
0002	Transfer central memory to PPU.
0003	Not used.
0004	Load overlay to CM library.
0005	Deadstart LDR and CPC PPUs.
0006 through 11	Refer to Auto.

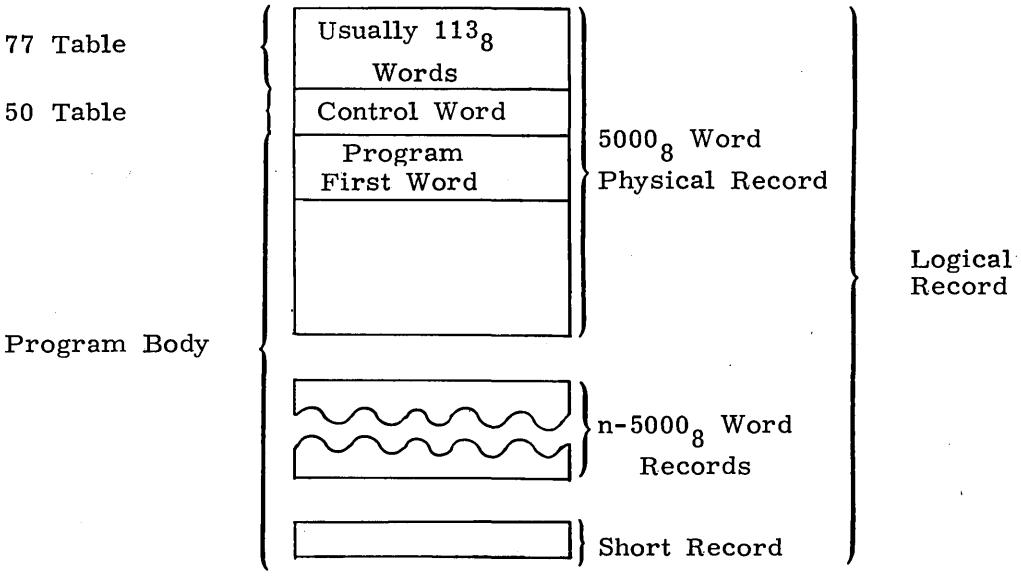
1. Load Program - Code 0001
   This call causes a program to be loaded from tape. Tape will be searched, if necessary.

   a. Keyboard Entry
      X.MNE. (cr)

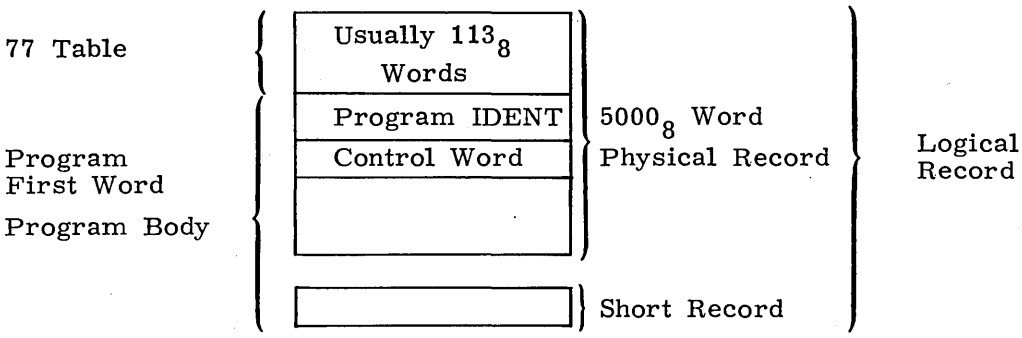
   b. Call Format
      CM Address 101 = 0001 0000 0000 0000 0000 (Output register)
      102 = MMNN EE00 0000 0000 0000 (Message buffer)

   c. LDR Action
      LDR rewinds and searches the SMM tape for the program whose mnemonic is left-justified in the calling PPUs message buffer, CM address 102 for PPU 8. If the end of file is reached, the message NOT IN DIRECTORY is written into CM address 103 to be picked up by CPC. If the program is found, its 77 table is stripped (usually 113 PPU words) and control word determines disposition of the record. CPU programs are identified by a five-word 50 table (refer to SCOPE Reference Manual, loader operation) which follows the 77 table. PPU program disposition is determined by the value of the programs first-word address which immediately follows the 77 table.

CPU PROGRAM LOGICAL RECORD



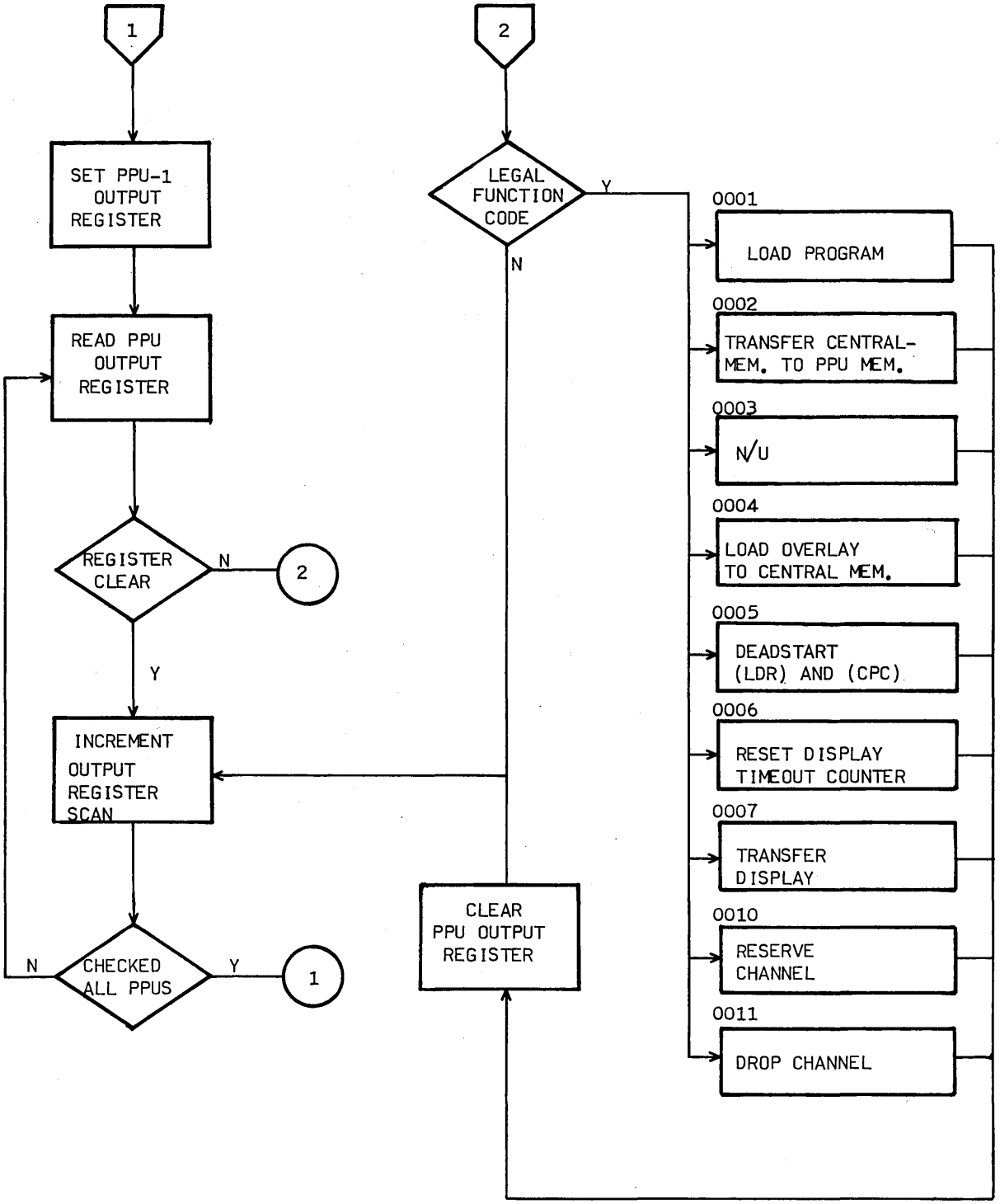
PPU PROGRAM LOGICAL RECORD



NOTE

A logical record does not necessarily contain one 5000_8 word record. The idea is that it ends on a record length less than 5000_8 words.

LDR MONITOR LOOP



Control Word	Program Type	(LDR) Action
0001	PPU-1	The program is shipped to the next available PPU, initially PPU-1. The program executes at address 2.
5000	CPU	Program is loaded into central memory at a relative address specified by CPC, initially 10000_8.
0000	PPU-0	CPC is requested to drop and to deadstart PPU-8. The program overlays LDR in PPU-0 and starts execution at address 1.
0001	PPU1-7, 9-19	Refer to auto.

2. Transfer Central Memory to PPU - Code 0002

   This call provides a method to enter an octal PPU program in central and then transfer it to a PPU.

   a. Keyboard Entry

      MTP, XX, (CR) XX is CM address.

   b. Call Format

      CM Address 101 = 0002 0000 0000 0000 0000

      CM Address XX = MMNN EE00 0000 0000 0000

      MMNNEE is the program mnemonic; must be nonzero.

   c. LDR Action

      LDR transfers 1462_8 CM words from CM address XX+1 biased by RA as set by CPC to the next available PPU-1. The program starts at the value +1 of the upper byte of CM address XX+1.

3. Unused Function

4. Load Overlay to CM Library - Code 0004

   This call facilitates loading of overlay segments into a central memory library when a PPU overlay program is loaded. The call is normally used by the PS interface.

   a. Keyboard Entry

      None.

   b. Call Format

      CM Address 101 = 0004 0000 0000 0000 0000

      102 = MMNN EE00 0000 0000 0000

      MNE is overlay segment mnemonic.

c. LDR Action

NOTE

If the calling PP is greater than the PP in which EXC resides, then the call is replaced by a drop function.

LDR searches the SMM tape for segment MNE and loads it into the overlay library which starts at CM address 1000_8. Each segment loaded is prefaced with an entry control word. The following format is used.

Entry Word	MMNN EE00 FFFF EEEE LLLL
Where MNE	Segment mnemonic
F	Entry first-word address in library
E	Ending address in library
L	Length of segment in CM words

5. Drop LDR and CPC - Code 0005

This call gives PPU programs the ability to drop SMM control.

a. Keyboard Entry

None.

b. Call Format

CM Address 101 = 0005 0000 0000 0000 0000

c. LDR Action

LDR requests CPC to drop; PPU-8 will go to deadstart conditions. LDR sets PPU-0 to deadstart conditions. This operation is completed when the call register is cleared by LDR.

ADDITIONAL FEATURES

1. SMM can be initialized by loading LDR from the card reader when SMM is on a disk or drum. The disk/drum parameters are in words 13 and 14 of the deadstart panel. Requires PCL loader card to load LDR (refer to writeup).

2. Deadstart exchange the CPU.

   In the case of CPU hangups, do a deadstart exchange of CPU0, CPU1, or both.

   a. Do a normal deadstart to get CPC display.

   b. Set up the CPU program you want to execute under the CPC A display.

c. Change the deadstart panel to the following.

```
1 - 0100
2 - 0100  LJM 100 D.S. EXCH CPU0
3 - 0300  HANG PP0 (LDR)
```

To deadstart exchange CPU1, change location 2 to 0120.

d. Put the deadstart switch in the D.S. position (ON); every deadstart will exchange the CPU.

e. To start both CPUs, use the following deadstart settings.

```
1 - 0100
2 - 0100  GO EXCH CPU 0
3 - 0100
4 - 0120  GO EXCH CPU 1
5 - 0300  Hang PPU 0
```

NOTE

LDR transfers the input PKG to the output EXCH PKG area (400 to 420 for CPU0, 600 to 620 for CPU1), then exchanges the CPU with the output EXCH PKG. A normal exchange jump (2600/2601) instruction is used for the exchange. LDR exits to location 3 of the deadstart panel.

3. Before calling CPC to PP10, LDR loads CPMTR to CM addresses 247-377.

4. LDR now loads the exchange package associated with a CP test being loaded to the exchange address set for the appropriate CPU.

Example:

```
EX,200000  cr
FST  cr
```

SPACEBAR — Allows a CPU program to be started by an exchange to the upper banks on a CYBER mainframe.

5. A0 is preset to equal FL.

6. MA is preset in the input exchange package to 340 for CPU0 and 360 for CPU1.

7. FE is preset to run ECM.

8. X0 is preset to equal the contents of central memory absolute address one.

9. PSM can now be loaded to PP12 - 23 while in NOT AUTO.

10. SMM can now reside on 1 844 pack for several mainframes to access concurrently (DUAL ACCESS).

MULTIPROGRAMMING ROUTINES

TAPE COPY ROUTINE (CPY)

OPERATIONAL PROCEDURE

RESTRICTIONS

1. All tapes used by the operator must be on the same channel and equipment.
2. Only 6000 formatted tapes may be copied (that is, the physical record length must not exceed 5000_8 words).
3. Will not run with 66X tape units.

LOADING PROCEDURE

SMM Auto Operation

X.CPY

Loads CPY. The channel and equipment number of the system deadstart tape is inserted into location 1502.

X.CPY,CCEE.

Loads CPY. CCEE is inserted into the tape parameter location 1502. CCEE cannot = 0000.

SMM Stand-Alone Operation

X.CPY.

Same as in auto operation.

PARAMETERS AND/OR ENTRIES

Location 1502 = CCEE; the channel and equipment numbers desired.

All entries are displayed on the left screen at load time.

Entries

E, I, T, W. (CR)

E = B - Binary copy
C - Coded copy
V - Verify

NOTE

Verify uses the last mode used (binary or coded).

I = Input unit number 0 through 17

T = Output unit number 0 through 17

NOTE

If the input unit number = output unit number, records specified by W will be skipped instead of copied.

W = May have the following format.

Absent - One record will be copied from I to T.

1-7777 (octal number) records will be copied from I to T.

+X - X octal number of files will be copied from I to T. If X is absent, one file will be copied from I to T.

MNE - Copy I to T up to and including record name MNE and stop.

NOTE

If MNE is used and the entry is terminated with a /, CPY will stop before copying the first 5000_8 words of record MNE to unit T. A space will complete the copy.

Parameters

K, X.	Backspace unit X (0 through 17) one logical record.
R, X.	Rewind unit X (0 through 17).
XXXX, YYYY.	Store Y in location X.
0221, 0222	Restart CPY.

D	Release display to SMM system.
SPACE	Continue after error message.
(CR)	Set repeat entry flag.

MESSAGES

NORMAL MESSAGES

REPEAT	Performs previous entry upon (CR).
COPY	Copies or skips a record.
VERIFY	Verifies a file.

ERROR MESSAGES

URPE	Tries to recover three times.
UWPE	Tries to recover three times.
NOWE	No write enable.
NRDY	Not ready.
ERR	Verifies error.

MAINTENANCE BITS CONTROL ROUTINE (DLY)

INTRODUCTION

This routine controls the maintenance bits of the interlock register. It is a maintenance aid which allows control of the various clock phases. It runs under SMM 6000/CYBER.

OPERATIONAL PROCEDURE

RESTRICTIONS

1. Runs under auto only.
2. Runs only on CYBER machines (72, 73, 74) and 6000 types with QSE for the interlock register. Routine will exit if channel 15 is nonexistent.

PARAMETERS AND/OR ENTRIES

1. All entries are displayed on the left screen after DLY has been loaded.
2. Location 70 = delay time (that is, the time before a clock phase change is made). This time is in milliseconds per count.

   Entries

   P1 = Phase I test point 6 delayed
   P2 = Phase II test point 1 delayed
   P3 = Phase III test point 2 delayed
   P4 = Phase IV test point 4 delayed
   A1 = Phase I test point 6 advanced
   A2 = Phase II test point 1 advanced
   A3 = Phase III test point 2 advanced
   A4 = Phase IV test point 4 advanced
   ON = Display PPU memory on both screens
   OF = Do not display PPU memory
   RR = Change clock phases randomly at a rate determined by (70)
   TM = Toggle mode, toggle between phase selected and no advanced or delayed phase
   AC = Clear all bits of the interlock register
   AA = Set all maintenance bits

MESSAGES

The messages displayed coincide with the following type-ins.

Type-In	Message
P1	PH 1 DELAYED
P2	PH 2 DELAYED
P3	PH 3 DELAYED
P4	PH 4 DELAYED
A1	PH 1 ADVANCED
A2	PH 2 ADVANCED
A3	PH 3 ADVANCED
A4	PH 4 ADVANCED
AC	ALL MAINT BITS CLEAR
AA	ALL MAINT BITS SET

DEADSTART TAPE EDITING ROUTINE (EDIT)

EDIT is a FORTRAN/SCOPE program designed to provide all the deadstart tape editing capability of EDT with the additional capability of setting individual test MCP parameters as well as test list definition. Edit will assemble and execute under RUN as well as FTN.

DECK STRUCTURE

The following job provides a listing of EDIT and a binary deck used in deck B.

SCOPE 3.3

```
JOB/ACCOUNT CARD
REQUEST (OLDPL, HI)
UPDATE (Q)
FTN(I=COMPILE, B=PUNCHB)
7
8
9
*C EDIT
6
7
8
9
```

SCOPE 3.4

```
JOB/ACCOUNT CARD
REQUEST (OLDPL, HI)
UPDATE (Q)
FTN(I=COMPILE, B=PUNCHB, S=CPCTEXT)
7
8
9
*C EDIT
6
7
8
9
```

The following job edits the deadstart tape (OLDSMM) and creates a new one (NEWSMM) with EDIT directives and program LDR changes.

```
JOB/ACCOUNT CARD
REQUEST (OLDPL, HI)
UPDATE (Q)
UNLOAD (OLDPL)
COMPASS (I=COMPILE, B=CHANGES)
REQUEST (OLDSMM, HI)
REQUEST (NEWSMM, HI)
LOAD (INPUT)
EXECUTE (EDIT)
```

7
8
9
    LDR Corrections
    *C LDR
7
8
9
    EDIT Binary Deck
7
8
9
7
8
9
    EDIT Directives
7
8
9
6
7
8
9

EDIT DIRECTIVES

The following directives may be used with EDIT.

ADDT.	Read lfn CHANGES and add/replace the programs from CHANGES to NEWSMM.
REC, RRRR.	Unconditionally copy $RRRR_8$ programs from OLDSMM to NEWSMM (default 100_8).
DS13, PPPT.	Set PPPT in CEL for address 13 (mass storage position/type parameter).
DS14, CCUE.	Set CCUE in CEL for address 14 (channel, unit, equipment of mass storage).

SYBB,SSSS.	Set BB,SSSS in CEL (BB-* CM banks, SSSS-system Type, that is, SY40, 6400).
TLlll ,MNE MNE MNE MNE.	Replace test list lll with MNE...MNE. (1 to 4 test mnemonics).
DDDD,CCEE. DDD,CCEE. DDD.	Define device type DDDD (or DDD) and place CCEE, if present, in MCP+2 of all appropriate programs (example, 512,0507).
MNE,MCP,MCP+1,MCP+2,..., MCP+13.	Set MCP parameters in program MNE. Example LP1,2,1,,0000.-yields 1500-0002, 1501-0001, 1502- unchanged, 1503-0003 in LP1.
DEF,MNE,MNE.MNE...MNE.	Define MCP entry point for new tests. (MNE, -MNE MCP=1000; MNE. - MNE MCP=1500; MNE - MCP is not defined.)

EDIT OUTPUT

EDIT lists all input directives and annotates those in error with a self-explanatory message. EDIT also lists the contents of the NEWSMM produced and notes what activity, if any, was performed on each program (activity indicators: R=replaced, A=added, E=edited (1502 set), M=MCP parameters modified).

Equipment Designator	Programs Transferred
SY04,XXXX	MM2
SY10,XXXX	M1A, M1B, M3A, M3B, MM2, M65, M6C, M6S
SY14,XXXX	M98, M9S, M9C, M1C, M1P, M3C, M3P, M2C
SY20,XXXX	MM2, MM1, MM3, M3R, M1R, M65, M6C, M6S, MM4, MM46
SY30,XXXX	M98, M9C, M9S, M1C, M1P, M2C, M3C, M3P
SY40,XXXX	MM2, MM1, M1R, MM3, M3R, M65, M6C, M6S, MM4, MM46, MCY, MCZ, MCS
SY60	
SY80	
ECSX	ECT, ECS, ECD, ECC, ECM, ECX, ECE, ECF, ECP, LPD
SDAX	SDA
BRCX	BRC
808	DFM, DUG, DFT, DT2, XF8, DF8, DT1, DFX, D38
6603	DF1, DF3, DF6, DF8, XF8, DFP, DF5, XF6

Equipment Designator	Programs Transferred
60X	MTT, MTX, MT4, TCT, S69, TAP, STT, RGG
501	LPT
512	LP1, XP1
415	CPT, CP1, CP2
1403	PRT
3245	CP2, PSQ
405	CRT, CR1
861	DR1
863	DR3, XR3
865	DR5, XR5
3234	DF4, XF4
2321	DCT
609	MT9, T9T
60Q	MTQ
65X	MMT, XMT, TCT, T5X
66X	MTS, XTS, MTC, MTZ
2265	MT1, XT1
280	D28
2021	MT2
3291	D91, MCC
329A	D10
3553	DF7, XF7, D41, DF9
329B	D11
6678	DDX
329C	D12, DDT
Q657	TT1, TT4
3294	D13, X13, TR1, XR1
6676	TT2, TT3
252	DST, XST, SJP
3266	TT5, TT6, RT4
250	D50, X50, SJ0
6674	RT3
GRID	GRD, XRD, RGI, RGT
3691	PTT
6671	RT5, XT5, RT6, XT6, RTX, RT7, XT7
3692	TP1

Equipment Designator	Programs Transferred
T122	SC8, TR7
3293	PLT
3682	SC1, SC2
6682	T82, SC9
DCA	CAT
SPAM	SPA, SP2, SPB, SPT, SP1
3276	TTT
626	WTT, MT6
627	WTQ
MUX	RMT
MATX	MAT, MAC
SACX	SAC, XAC, CSM, SIO, XLC, PLC
MEJ	MXJ, MXC, MXD, MAN, MNC, MNP, MNX
CMI	BD1, BDP, CMS, CMT
INRG	IRT
DDPX	DDP, CDP
INMP	IMC, IMT
ICLC	ICL
THEO	TIC
RTGD	RGT, RGI
LCCX	LCC, BCY, XCY, XLC
844	DP8, XP8, MY9, XY9, D44, FMT, XMT
BCXC	BCX, XCX, MY8, XY8, PLD, CID, XID, CDM, XDM
580	FTP
17X	CNF, CN1, CPM, CSC, CSS, CSP, FFZ, PPM, SCD, SCR, STM, TRC, TRP, CHP, EJ1
175	C01, CPX, DAT, IAS, IW5, SK1, SK2, SK3, MA5, CT7, CU4

Programs CPY, DMP, LST, CMC, DS1, ERX, PCX, and CH2 are tabled so their MCP parameters may be modified. The table may be extended using the DEF directive to cover the addition of new programs.

TAPE EDIT ROUTINE (EDT)

OPERATIONAL PROCEDURE

RESTRICTIONS

1. All tapes used by the operator must be on the same channel and equipment as the deadstart tape.
2. Any set of units 0 through 7 or 10 through 17 may be used. However, sets of units may not be mixed together (for example, unit 1 may not be run with unit 12).
3. If no edit unit is requested at load time, EDT will assume unit X0 is to be the master tape and unit X1 as the edit tape or scratch tape. X is obtained from the deadstart panel.
4. Will not run with 66X tape units.

LOADING PROCEDURE

SMM Auto Operation

X.EDT.

This entry will force the edit routine to use units X0 and X1.

X.EDT,U.

This entry will force the edit routine to use the deadstart tape as the master and unit U as the edit tape. U may be 1 through 7.

SMM Stand-Alone Operation

X.EDT

Units X0 and X1 are used.

PARAMETERS AND/OR ENTRIES

All entries are displayed on the right and left screen after EDT has been loaded.

ADDY

This entry will allow records to be replaced or added to the edit tape from unit Y. Unit Y must contain binary records of programs that have been assembled under COMPASS and must terminate with an EOF. Y may be 0 through 7.

The long master tape must be used when changing system entries (SYBB,XXXX.) or deleting records previously added.

SPACE

Starts edit operation.

*

Releases the display back to the SMM system. This entry is active only during parameter entry time under SMM auto operation.

D

Same as * entry, except it is active during the edit operation.

BP,XXX.

This entry will stop EDT on program name XXX before writing the first 5000 words of program XXX on the edit tape. Actual location 1500 of the program will be in location 4620 of EDT. Any changes to the program may be made at this time and a space will write the modified program to the edit tape. Up to 16 programs may be selected with individual BP,X. entries. Location 3120-3124 = location 1 of CPU tests for the first 761 CM words.

TTT. or TTT,CCEE or SYBB,XXXX. (BB=number of banks)

These entries are explained on the display.

Example: If you have a 6400 (65K) machine with a 501, 60X's, 415, 808, and a 405, make the following entries.

TTT=SY20,6400. = 65K.	Selects the 65K memory tests and 6400 CPU0 (20 banks).
808,0001.	Selects the 808 tests and inserts 0001 in locations 1502 or 1002. CCEE may not be 0000.
501,1207.	Selects the 501 tests and inserts 1207 in locations 1502 or 1002.
60X,1305.	Same as above for a 3X2X/60X tape controller.
415,1205.	Same as above for a 415 punch.
405,1204.	Same as above for a 405 card reader.

Location 0064

This location contains the number of SMM records that are automatically copied to the edit tape. It is preset to copy all of the SMM service routines.

MESSAGES

NORMAL MESSAGES

COPYING XXX

Record XXX is being copied to the edit tape.

EDITING XXX

Record XXX is being edited.

REPLACED XXX

Record XXX was replaced from the add unit.

ADDED XXX

Record XXX was added to the edit tape from the add tape.

EDIT COMPLETE

Editing has been completed. EDT must be reloaded to do another edit.

BP-XX

A program has been read which was selected by a BP,XX. entry. Make changes and hit spacebar to continue. XX is the program name requested to breakpoint.

ERROR MESSAGES

UNIT XX NOT READY

NO WRITE ENABLE, On edit unit

CONNECT REJECT UXX

READ PARITY ERROR

Tries three times. This record will be lost if a spacebar is hit.

WRITE PARITY ERROR

Tries three times. This record will be lost if a spacebar is hit.

SAMPLE SYSTEM ENTRIES

```
SY04,6416 =  16K, 6416 system
SY10,6400 =  32K, 6400 system
SY30,6500 =  98K, 6500 system
SY40,6600 = 131K, 6600 system
SY20,7214 =  65K, CYBER 70-7214 system
SY40,7428 = 131K, CYBER 70-7428 system
SY14,7313 =  49K, CYBER 70-7313 system
```

FULL ADDRESSING FOR CENTRAL MEMORY UTILITY PROGRAM (FAD)

DESCRIPTION

FAD is an auto-mode multiprocessing utility program. When FAD is called, it enters a PPU and writes all of central memory with an address pattern (that is, each location contains its address as data) in one of three modes. Central memory is written from a low address to the last address in memory. The last address is obtained by FAD from word 1 of central memory which can be altered under CEL/ENS using an SYxx,yyyy command. Once it has completed addressing memory, it releases the PPU back to SMM in the idle state.

USAGE

The various modes of FAD are called by typing one of the following under the auto mode display.

FAD•cr or FAD,0•cr

This command causes FAD to load and write addresses in central memory from absolute address 360B to the end of central memory.

Example: location 1001=0000 0000 0000 0000 1001

FAD,1•cr

This command causes FAD to load and write addresses in central memory from the RA given in the exchange package at location 400B to the end of central memory.

Example: location RA+74576 = 0000 0000 0000 0007 4576

FAD,2•cr

This command causes FAD to load and write jump instructions to the current address in central memory from the RA given in the exchange package at location 400B to the end of memory.

Example: location RA+36574 = 0200 0365 7400 0000 0000

FAD, 3·cr	This command causes FAD to load and halt at the parameter stop message. Unless the parameter at location 1502 remains unchanged, the go command causes FAD to do nothing but loop back to the parameter stop message.
FAD, 4·cr through FAD, 7·cr	These commands cause FAD to load and execute in the same manner as FAD, 0 through FAD, 3 respectively, except that a parameter stop occurs first.

MESSAGES

SET PARAMS - mm/dd/yy

This message is displayed at parameter stop only. This message must be displayed if the user desires to use an n·DIS command. The mm/dd/yy represents the month, day, and year of the latest assembly of FAD.

SETTING FULL ADDRESS

This message is displayed to indicate that FAD is currently running (that is, executing) commands FAD, 0 through FAD, 2.

PERIPHERAL SERVICE MULTIPROGRAMMING ROUTINE (PSM)

SMM CALL

PSM,ABCD. CR where 1502=ABCD after loading.

NOTE

Location 0045=3000 after loading.

PURPOSE

PSMS creation was influenced by the following items.

1. Encourage CEs in the field to write peripheral programs, since this leads to a more thorough understanding of the equipment and gives the CE a better feel for operating system problems.
2. Retain as many PS interface package features as possible; the CE should already be familiar with its operation and use.
3. Incorporate as many service routines as possible from PSP for use under auto, because they have already proven their usefulness in the field.
4. Enable a peripheral program to run simultaneously with a central program under SMM control.

USAGE

DISPLAY

1. The PPU memory display can be changed the same as the PS interface, (that is, 0200W20AA,0204W20AA, 0210W20AA,0214W20AA; where AA X 100 is the location to be displayed).
2. A programmer can create his own messages on the left screen by calling one of the three standard PS message displaying routines. (Refer to use of message options.)
3. Selectable CPC G display for monitoring other PPs while programming a PPU.

KEYBOARD ENTRIES

The following entries do not require carriage return.

F — Clears and returns the PP memory display from the right screen (used to speed up program execution).

G — Starts program execution at the address specified by the contents of location 0045 (same as PSP).

S — Stops program execution by returning the program to the display loop.

NOTE

If a programmer has been displaying a message, this message is cleared from the screen and a normal PSM display will replace it. Therefore, if a programmer is displaying its own messages, an RJM-PS. MSGS should be used to stop program execution. If this is done, a SPACE or GO from SMM will continue the program following the PS.MSGS call.

) — Ends PSM and idles the PPU.

= — Enables sequence storing when used anytime after a valid octal input (need not be the fifth entry).
Example: 20=XX CR or 3000=XXX CR

+ — Increments by 100 the locations displayed by the first three PP memory displays (anytime).

- — Decrements by 100 the locations displayed by the first three PP memory displays (anytime).

D — Drop; returns the display to CPC (anytime).

/ — Allows the programmer to insert data or code at any address while shifting all code above that location up one location.

NOTE

No attempt is made to correct any code that references an address within the shifted area.

RIGHT BLANK KEY — Toggles the left display between the instruction and CPC G displays (use to monitor how other peripheral tests are progressing while writing a program).

BACK-SPACE	Clears the last keyboard entry.
SPACE-BAR	If it is the first entry: 1. And no message call to PS. MSGS has been made by the programmer, then 3000B is stored in location 0045 and program execution is started at location 3000. 2. If a message call to PS. MSGS has been made, then program execution continues at the next instruction after the RJM-PS. MSGS. If other than the first entry, it is a delimiter.
LEFT BLANK KEY	Clears keyboard entry pointer, clears sequence storing flag, and clears keyboard message flag.
CR	As the first entry, sets repeat entry flag. Otherwise, executes the command entered.

SERVICE ROUTINES

RBCCXE, YYYY, ZZZZ CR	Reads absolute binary cards from channel CC, equipment E to PPU location YYYY, where ZZZZ is the number of cards (octal count).
LOCCXE CR	Load octal cards routine. Loads Hollerith keypunched octal numbers to locations specified. Format example: 3000= 1 2 3 3010-10 Results: 3000=0001 3001=0002 3002=0003 3010=0010
=	Immediately following an octal number reset the address to which the numbers following the = sign are sequence stored.
SPACES	Acceptable delimiters. Any number can be used between entries.

"."	Signifies end of information and routine will continue reading until one is encountered.
	NOTE Data is sequenced stored if a new address is not specified each time. There is no limit on the number of cards that can be input.
PBCCXE, YYYY, ZZZZ CR	Punches absolute binary cards. (Refer to RB ROUT.)
	NOTE A MAC will be assumed by the previous three routines on the channel specified by CC if DF.1015 (MAC code) has been enabled in PSM and then X is defined as the MAC number.
CM CR	Clears PP memory from 20 through 67, 3000 through 7777, and resets location 0045=3000.
RCAAAA, BBBB, CCCC or RCAAAA, BBBBB, CCC or RCAAAA, BBBBBBCCC	Reads central memory to PPU address AAAA from central address BBBBBB, where CCCC is number of central words (octal count).
	NOTE Any central address can be referenced by this routine and delimiters are not needed. PP address is four numbers, central address is six numbers, and word count is one through four numbers.
WCAAAA, BBBB, CCCC or ETC.	Writes central memory. Refer to RC routine for format and use.
CHDDDD, EEEE, FF CR	Changes channel routine, which will change the channel number of any channel instructions from location DDDD for EEE number of locations, where FF = channel number.
WPGGGG, HHHH, IIII CR	Writes pattern in PPU memory from address GGGG for HHHH number of words, where IIII is the pattern.
RWJ, CCOE CR	Rewinds tape unit J on channel CC, equipment E.
	NOTE This routine can also be used as a connect routine.

ULK, CCOE CR — Unloads tape unit K on channel CC, equipment K.

NOTE

If code is not specified in the previous two routines, then the deadstart channel and equipment is substituted.

If DF.1015 (MAC code) is defined to PSM then the previous two routines do not exist.

PROGRAMMING CONSIDERATIONS (use 20 through 67, 1500 through 1510, 3000 through 7777)

PSM communication with CPC/LDR, ending a program with:

1. An LJM-232 executes the program once, and then returns to the display loop.
2. An LJM-236 will loop through a program while retaining system communication.
3. An LJM-100 at anytime is illegal, because system communication will be lost.

Once a program has been started, either by a SPACE, G, or X.Go., the point at which the program will be entered can be changed by storing the new starting address in location 0221.

The request and release channel routines should be incorporated in your program if there is any chance that another PPU may become active on a channel you are using. Previous to making use of these routines, store 1502-CCXX, where CC is channel number and XX can be anything. To request a channel, do an RJM-1141 and to release the channel, do an RJM-1161.

If a 6681 or 6684 is present on the channel you wish to drive equipment on, issue a (77CC-2000) connect before trying to connect the equipment. Although the 81 and 84 are connected on deadstart, SMM diagnostic program DF selects them after use.

If a programmer should wish to completely take over the display console and associated keyboard input, then a REQUEST DISPLAY (RJM-325) must be done before any instruction can be executed on the display channel.

Use of the message option. If a programmer wishes to replace the RUNNING message with an informative or error message both in the PPU and in the CPC G display, then use one of the following routines.

PS. MSG	Displays message on the left-hand screen and PPU memory on the right screen, RJM-1030.
PS. MSGS	Same operation as PS. MSG, except PSM will stay in the display loop until the SPACEBAR is hit or a GO is received for CPC, RJM-1061.
PS. CPMSG	Same as PS. MSG, except no PPU memory display, RJM-1046.

To make use of any of these routines, simply load A with the address of the first location of the message you wish to display (which must end with a word of zeros) and do a return jump to one of the message routines.

NOTE

Besides displaying a message, the message routines carry on normal communication with LDR/CPC. Also, if 10000 is added to the message address, then DMP will pick up message.

Example 1:

```
3000-2000
3001-3100
3002-0200
3003-XXXX
3004-0100
3005-3000
```

Load A register with address of the information to be displayed, and then do a return jump to one of the display routines. Either loop on your program or jump to 232 or 236.

Example 2:

```
3000-2001
3001-3100
3002-0200
3003-XXXX
```

Adding 10000 to message FWA enables DMP to pick up the message; jump to message routine.

If the RB, LO, PB routine halts without clearing the keyboard display buffer and returning the PPU display to the right screen, one of the following conditions may exist.

1. LO routine has not found a period on a card read, and card reader is not ready.
2. RB routine has not yet read the number of cards specified, and the reader is not ready.
3. PB routine has not yet punched the number of cards specified, and the punch is not ready.
4. Any of the three routines have found the channel active after a function 2000, peripheral equipment rejected the connect attempt, 6681 rejected or had transmission parity set during transfer.

To correct any of the above conditions, type S to stop the program, F to return the PP memory display, and then drop back to CPC. If PSM still has the channel reserved and the channel is full, then condition 4A. If the channel is reserved and appears to be running, then conditions 1, 2, or 3. If the channel has been released, then conditions 4B or 4C.

KEY LOCATION CALLS

PS.PAUSE Checks for system abort, stop, go, and parameter entry flags. Does not call RCH or DCH.

Example:
```
        RJM     PS.PAUSE+1
RJM-1261
```

PS.RCH Requests the channel number contained in MCP+2 (1502) from the SMM system. If the program has not separated a call to RCH by a call to DCH, no operation will be done.

RJM-1141

PS.DCH Operates the same as a call to PS.RCH, except it releases the channel to the SMM system.

RJM-1161

PS.CPMSG Displays a message on the left screen only. If DF.MLP has been defined, the message will be sent to the PPS message buffer in central memory. Call PS.PAUSE and control is given back to the program. The message to be displayed may be any length, as long as each new message contains its own coordinates. The entire message must terminate with a 12-bit word of zeros.

```
EX.1            LDC     DIS1
                RJM     PS.CPMSG+1
        RJM-1046
```

NOTE

If you wish the initial message to be sent to the SMM dayfile, add 10000B to the display FWA.

Example:

```
LDC     DIS1+10000B
DIS1    DATA    H*RUNNING SEC1*
        DATA    6000B,7200B
        DATA    H*STATUS = XXXX*
        DATA    6000B,7300B
        DATA    H*TEST ALL UNITS*
        DATA    0
```

EX. 2

```
        LDC     DIS2
        RJM     PS.CPMSG+1
DIS2    DIS     ,*RUNNING SEC2*
```

PS. MSG — Performs the same operation as CPMSG, except PP memory is displayed on the right screen.

RJM-1030

PS. MSGS — Performs the same operation as PC. MSG, except PS will stay in the display loop until the spacebar is hit or a GO is received under SMM system operation.

RJM-1061

PS. RMAC and PS. DMAC — If DF. 1015 is defined, then the programmer can select the MAC (whose code is in MCP+5) by making a call to PS. RMAC and deselect the MAC by making a call to PS. DMAC. WA will be displayed when access is locked out. 1015 code must be in MCP+5.

```
FORM = 3X00B    X=ACCESS
                PS.RMAC     RJM-2515
                PS.DMAC     RJM-2546
```

KEY LOCATIONS USED BY PS

PS. PPN — Contains the PP number the program is currently running in (location 414).

PS. PPIR — Contains the PP input register pointer to central memory (location 1).

PS. PPOR — Contains the PP output register pointer to central memory (location 2).

PS. PPMB — Contains the PP message buffer pointer to central memory (location 1331).

To use the breakpoint option, store the first address in location 77, the last address in location 76, then 221W222 runs portion. A register is saved and last address becomes first address. Subsequent portions, if sequential, only need 76=LWA, then 221W222 or SPACE to run segment.

If a program is to do its own displaying, the first instruction of the program should be an RJM PS.RWDIS+1. This will keep the program in communication with the system and keep the program from displaying if DIS is not assigned to the control point.

Carriage return before any entry sets REPEAT ENTRY FLAG which is cleared by the clear key.

PERIPHERAL SERVICES (PSX)

PSX is a 3000 peripheral service routine designed to allow the CE to make alpha keyboard entries to accomplish various I/O tasks. It runs under either SMM, auto mode, or SCOPE.

KEYBOARD ENTRIES

Entry	Function
RQCXX.	Sets up PSX to use channel XX; this must be the first I/O entry word.
RLCXX.	Releases channel XX to SMM for use by another PPU.
FCHX.	Functions the channel/6681 with function X.
CONX.	Connects equipment on requested channel [X=connect code and (A) = 6681 status].
FNCX.	Functions equipment with function X [(A) = 6681 status].
OUTX.	Outputs X number of words from location 7000 [X may be 1 through 1000 and (A) = equipment status].
INPX.	Inputs X number of words to location 6000. 1500 read mode is used [X may be 1 through 1000 and (A) = equipment status].
EST.	Takes equipment status and displays on left screen C=XXXX [(A) = equipment status].
CST.	Takes 6681 status and displays on left screen C=XXXX [(A) - 6681 status].
B,XY	Transfers X buffer to Y buffer area (X=I, O, P) (Y=I, O, P).
PXXXX.	Sets P address of PPU to XXXX.
MFXX.	Sets right screen memory display field F to XX times 100. F may be A, B, C, or D.
ENT,F,L,X.	Sets X in PP memory from location F to location L.
XXXX.	Sets XXXX in PP memory at current P address.
D	Releases display to SMM (CPC).
G	Starts program (which is in test mode) and makes one pass starting from location 5000.
SPACE	Starts program (which is in test mode) from location 5000 and runs until S key is depressed.

Entry	Function
S	Stops program running in test mode.
T	Sets test mode.
(CR)	Sets repeat entry flag.
Blank (55)	Clears test mode flag and resets P address to 5000.

OPERATION

If test mode is not set, all I/O entries are executed as they are entered.

6681 status (C), (E) equipment status, and the current (P) address are displayed at all times.

If test mode is set, all I/O entries are stored in the current P address area (5000 through 5777) and not executed until a G or a space is entered. P is automatically updated for each entry made. Octal entries (XXXX.) may be mixed with I/O entries or they may be the only entries used.

Locations 20 through 67 and 5000 through 5777 are reserved for operator use.

Locations 6000 through 6777 are reserved for the input buffer. If no input operation is to be performed, this area may be used to store a program.

Locations 7000 through 7777 are reserved for the output buffer. If no output operation is to be performed, this area may be used to store a program.

Example:

One wishes to read cards from card reader on channel 12, equipment 4, punch the card on card punch channel 12, equipment 5, and output the card on line printer channel 11, equipment 6. Enter the following.

Entry	Description
T	Sets test mode.
RQC12.	Sets up PSX to use channel 12.
CON4000.	Connects card reader.
FNC1.	Sends function 1 to card reader.
INP120.	Inputs one card.
B,IO	Transfers input buffer to output buffer.
CON5000.	Connects card punch.
FNC1.	Sends function 1 to card punch.
OUT120.	Outputs one card.
RLC12.	Releases channel 12 to SMM.
RQC11.	Sets up PSX to use channel 11.
CON6000.	Connects line printer.
OUT104.	Outputs one line (one card).
RLC11.	Releases channel 11 to SMM
SPACE	Starts program running until S key is depressed.

To use MODE I connect and function, set location 1500-0020. To wait not busy before sending a function to the equipment, set location 1663=XXXX, where XXXX = the wait count in seconds.

Any channel error encountered while doing an I/O operation will be displayed in the standard PSIO format.

If a channel error exists during a run in test mode, start the program over by a G, or space, or continue on by releasing the display to CPC and typing x.GO, where X= PPU that PSX is running in. Set 1501=0001 to ignore channel errors and loop, waiting for an accept, reply, etc. from the controller.

The SCOPE version of PSX operates in the same fashion as the SMM version with the following exceptions.

1. Only the display may be assigned to the control point. D entry will release the display to the system. Any channel error will abort PSX. If peripherals are to be used, they must first be turned off. Also, the entry RQCX. must be used before the program and RLCX. must be the last entry.
2. PSX may be called in by job cards or DIS. No central memory is used.
3. Direct locations 30 through 47 may be used under the system version of PSX.

CPU TEST MODE UTILITY PROGRAM (TST)

DESCRIPTION

TST is an auto-mode multiprocessing utility program. When TST is called, it enters a PPU, locks out CPC's CPU control, and begins to repeatedly exchange the CPU selected. It uses the exchange package and program in the CPU that the user sets up before loading TST.

USAGE

The following are commands that the user may use to control TST.

TST,0•cr or TST•cr	This command causes TST to load and begin repeated exchanges to CPU0, using the exchange package at location 400B as the input exchange package address and 420B as the output exchange address.
TST,1•cr	This command causes TST to load and begin repeated exchanges to CPU1, using the exchange package at location 600B as the input exchange package address and 620B as the output exchange address.
n•DROP•cr	Where n is the PPU number for TST, this command causes TST to clear CPC lockout, exchange the CPU out, and idle that PPU.
n•STOP•cr	This command causes TST to exchange the CPU out and clear CPC lockout. It then waits for either a go or drop command.
n•GO•cr	This command causes TST to exchange the CPU in.

MESSAGES

SET PARAMS - mm/dd/yy

This message is displayed at parameter stop only. This message must be displayed if the user desires to use the n.DIS. command. The mm/dd/yy represents the month, day, and year of the latest assembly of FAD.

NOTE

Parameter stop results when the commands TST,4. or TST,5. are used.

*** CPU IN TEST MODE ***

This message is present in the CPC message buffer only when TST is repeatedly exchanging; thus, it is a running message.

INPUT PACKAGE = xxx (EXCH ADDRESS = yyy)

This message is displayed at all times in the PPU's message buffer. For CPU0, xxx is 400 and yyy is 420. For CPU1, xxx is 600 and yyy is 620.

NOTES

1. The time between two exchanges is approximately 194 microseconds.
2. The user must load or write into central the CPU program desired.
3. The user must set up the exchange package at 400 or 600 before running TST.

STAND-ALONE ROUTINES

PERIPHERAL SERVICE ROUTINE (PSP/PSQ)†

DISPLAYS

RIGHT SCREEN

A, B, C, D, is displayed from top to bottom.

To change:

A display: Ayyyycr or Acr increments by 100 or 0200x20xxcr
B display: Byyyycr or Bcr increments by 100 or 0204x20xxcr
C display: Cyyyycr or Ccr increments by 100 or 0210x20xxcr
D display: Dyyyycr or Dcr increments by 100 or 0214x20xxcr
Eyyyycr or Ecr

Eyyyycr will set A to yyyy, and B and C 100 greater, respectively.

Ecr will increment A 300, and B and C 100 greater, respectively.

cr = carriage return, xx = address in 100ths, yyyy = address (lead zero may be omitted).

LEFT SCREEN

Top line: PPO* Channel Status
Middle: Description
Bottom: Keyboard display

*PPO = PPU number PSP is running in.

INPUT

KEYBOARD

1: If first character is numeric, format is xxxxzyyyycr.

xxxx = address to store at, yyyy = contents (yyyy leading zeros not needed), z = any character

2: If first character is alpha, format is shown for each routine.

† PSQ is PSP for 3245 controller.

Single Character

G Execute program stored at address contained in address 45.
or 0221xyyyycr yyyy = address (leading zeros may be omitted).

F Remove displays from right and left screen, except for channel status and keyboard input.

F Restore displays.

S Stop running program if looping through PST
or 0221x0232.

* Set sequence storing.

* Clear sequence storing.

Blank keys:	Clear keyboard display.
Backspace:	Clear last keyboard entry.
Carriage return:	Terminate keyboard input.
Comma:	Separator between numerics.

To loop on the program, jump back to address 0236 to retain display.

To execute the program once, jump back to address 0232 to retain display.

ROUTINES

Leading zeros may be omitted, four numbers naximum between commas.

CM

Clear PPU memory addresses 0 to 37 and 3000 to 7777.

Format: CMcr

PP

Transfer PSP to PPU requested and display PPU number top of left screen.

Format: PPxcr x = PPU number

PM

Read 200 words from PPU memory requested, starting at address specified and display right screen in A and B.

Format: PMx,yyyycr x = PPU number yyyy = starting address

RB

Read binary cards.

Format: RBcheq,yyyy,cccccr

cheq = Channel/equipment	Exam:	1204
yyyy = Starting address	Exam:	3000
cccc = Number of cards	Exam:	2

NOTE

Cheq may be substituted with a comma if card reader is 1204.

PB

Punch binary cards.

Format: PBcheq,yyyy,cccccr

Same as RB.

NOTE

Cheq may be substituted with a comma if punch is 1307.

CC

Copy card from reader to punch to two consecutive end-of-file cards (column 1, 6789 punch).

Format: CCcheq,cheqcr cheq = channel/equipment

First cheq is card reader (comma if card reader is 1204).

Second cheq is punch (comma if punch is 1307).

NOTE

After RB, PB, or CC is done once, do not type in cheq until PSP is reloaded.

WC

Write central memory.

Format: WCxxxx, yyyy, wwwwcr
xxxx = PPU starting address
yyyy = Central memory address
wwww= Number of central words

RC

Read central memory.

Format: RCxxxx, yyyy, wwwwcr
Same as WC.

WP

Write pattern in PPU memory.

Format: WPxxxx, nnnn, pppp
xxxx = PPU starting address
nnnn = Number of words
pppp = Pattern to store

Addresses 42, 43, and 44 are the translated values from keyboard input for the following routines.

PP, PM, RB, PB, CC, WC, RC, WP.

CH

Channel insertion option (PSP only).

Format: CHXXXX, YYYY, ZZ. cr.

zz = The channel number to be added to all I/O instructions from location XXXX, up to XXXX+YYYY.

PERIPHERAL SERVICE ROUTINE (PST)

PURPOSE

Provides certain I/O service routines under SMM and enables a peripheral program to be run simultaneously with a central program under SMM control.

OPERATIONAL PROCEDURE

The program runs under SMM control or deadstart from a card deck. After program loads, type X.DIS, where X = PPU number.

PARAMETERS

Routines may be started and parameters entered as indicated on the display. All entries (except G and S) must be ended with a carriage return.

SPECIAL OPTIONS

S	Stops execution of a routine.
G	Restarts execution of a routine.
ON	Forces a memory display during execution.
OFF	Restores normal off mode of memory display during execution.
DCnn	Disconnects channel nn.
MCnn	Executes a master clear function (1700B) on channel nn.
CC	Returns to central display.

Connect codes for 3000 type equipments may be set by entering the Chippewa mnemonic for the equipment followed by a one- or two-digit channel and a single-digit equipment code.

Example: CR125 sets card reader to channel 12B equipment 5.

Status displays are self-explanatory, except the abbreviation RD represents read equipment and WT represents write equipment.

Automatic incrementing and decrementing of peripheral store addresses may be obtained by using the + and - keys.

The 53B blank key clears the entry line.

Leading zeros may be deleted on all data entries, except the addresses.

EXECUTION

If a program is started by entering 0221, XXXX will remain running (provided it returns to 0236) when control is given to CC (central display).

CALLING AND ADDING TESTS†

PPU TEST — SMM

1. Load SMM system. (Refer to deadstart sequence and settings.)
2. Load test.
   a. When SMM has loaded, (CR) will select AUTO.
   b. Enter test mnemonic:
      X.MNE. (cr)
   c. Test will load from tape.
      1) If test is a PP-0 test, SMM will drop out and the test will take over display.
      2) If test is not a PP-0 test, enter N.DIS. to release display console to test.

PPU TEST— SCOPE

1. Load SCOPE system.
2. Load test.

   There are several possible ways to load a PPU test under SCOPE. In the following examples, LPT is called to test line printer 20.

† Refer to parameter in appendix A.

a. Control Cards

JOB.
REQUEST(20), A. Operator assigns printer
ONSW1. Stop on error flag
LPT(100) Call test; section 6 is selected
$6_{7_{8_9}}$

b. Initiate from Job Display - DIS

Enter:

7. ASSIGN 20. (cr) Assign printer 20
7. DIS. (cr) Call DIS

DIS will take over display.

Enter:

ONSW1. (cr) Stop on error flag
LPT, 100. (cr) Call test; section 6 is selected
DROP. (cr) Drop DIS

CPU TEST — SMM

1. Load SMM system.
   Refer to deadstart sequence and settings.
2. Load test.
   a. When SMM has loaded, CR brings up AUTO.
   b. Enter test mnemonics:
      MNE. (cr)
   c. Test load; message LOADING on left screen disappears.
   d. Hit spacebar to start test.

CPU TEST — SCOPE

1. Load SCOPE.

2. Load test.

   There are two ways to load CPU tests under SCOPE. In the following examples, CT3 is loaded and run in a field length of 10,000. Parameter word 2 at address RA+ 104 is set to 1.

   a. Control Cards

      JOB,T777,CM10000.
      Mode 0. Set error mode - 0
      CT3,,,1. Address 4 = 0001
      $6_7 8_9$

   b. Initiate from Job Display - DIS

      Enter:

      7.DIS. (cr) Call DIS)

      DIS takes over console.

      Enter:

      ENEM,0. (cr) Set error mode = 0
      ENS.CT3. (cr) Enter control statement
      RSS. (cr) Load test
      104,1. (cr) Set parameter
      RCP. (cr) Start test
      DROP. (cr) Drop DIS

      or Enter: No parameter change

      ENEM,0. (cr)
      ENS.CT3. (cr) Enter control statement
      RNS. (cr) Load and start test

NEW PPU TEST — SMM

1. Add to tape and run.

   a. Run the following job under SCOPE.

      JOB.
      REQUEST(SMM) Assign SMM tape
      REQUEST(NEW) Assign scratch tape

```
COPYBR(SMM,NEW,15)      Copy SMM system records
COPYBR(INPUT,NEW)       Add test
COPYBF(SMM,NEW)         Copy remainder of SMM
7 8 9
Binary deck             New test
6 7 8 9
```

b. Run test as in 1.

NEW PPU TEST — SCOPE (3.3 and lower only)

1. Temporarily add and run test.

   a. Add program LPT:

```
JOB.
EDITLIB.
7 8 9
READY(SYSTEM)
ADD(LPT,INPUT,DS)       Add test to peripheral lib
COMPLETE.
7 8 9
Binary Deck
6 7 8 9
```

   b. Replace program LPT:

```
JOB.
EDITLIB.
7 8 9
READY(SYSTEM)
DELETE(LPT)             Delete old LPT
ADD(LPT,INPUT,DS)       Add new LPT
COMPLETE.
7 8 9
Binary Deck
6 7 8 9
```

c. Run test as in B.

2. Permanently add test, consult CE analyst or Installation Handbook.

NEW CPU TEST — SMM

1. Add to tape and run.

   a. Use procedure as in new PPU test - SMM.

   b. Run as in CPU tests - SMM.

NEW CPU TEST — SCOPE

1. Assemble and run CPU test.

JOB,T500,CM60000.	Job card
COMPASS(B=TEST)	Assemble CPU program and place binary output on file TEST.
MODE 0.	Set error mode = 0.
TEST.	Call program from file TEST.
7_{8_9}	
(New test source cards)	
$6_{7_{8_9}}$	

2. JOB,T500,CM10000.
   MODE 0.
   INPUT.
   7_{8_9}
   Binary deck
   $6_{7_{8_9}}$

3. Add to tape and run

   a. Same as add procedure in new PPU test - SCOPE.

   b. Run test as in CPU tests - SCOPE.

   c. For SCOPE 3.4 and above, consult CE analyst or Installation Handbook.

PROGRAMMING SPECIFICATIONS

GENERAL

PROGRAMMING FOR SMM 6000

Programming CPU tests is relatively straight-forward for SMM or SCOPE. The program should be a good diagnostic, easy for the CE to use, and well documented. Examples of CPU diagnostics are covered elsewhere in this document.

In programming a PPU test, it is imperative that the programmer use and understand the various interface packages available. These interfaces reside on the SMM UPDATE tape as COMMON decks, and as such, any program may use them. Through the use of a *CALL card (that is, *CALL,PS) the interface becomes an integral part of the program. For a detailed description of the various interfaces, including diagrams and sample program, refer to interfaces available.

INTERFACES AVAILABLE

SMM INTERFACE (PS)

PS is a PPU package, 1477_8 words in length which serves as the standard front end for most peripheral diagnostics. Included in PS are a PPU memory display, I/O channel status display, a program breakpoint option, a message processor, a link between program and keyboard, and an overlay load routine.

INITIALIZATION

PS initialization depends on the program type, either stand-alone or multiprocessing.

STAND-ALONE PROGRAMS

Stand-alone PPU programs load into PPU-0. Control is given to the program at address (START).

MULTIPROCESSING PROGRAMS

Multiprocessing programs are identified to PS by having the symbol DF.OVL and/or DF.MLP defined in the program. The main program is loaded into PPU-1 and its segments, as defined in a table identified by the symbol PS.OVLT, are loaded into the CM overlay library. Control is given to program at address (START).

NOTE

Segments must not directly reference interface.

KEY LOCATION CALLS

PS.PAUSE

Used by the programmer if DF.MLP is defined. It will check for system abort, stop, go, and parameter entry flags. Does not call RCH or DCH.
Example: RJM PS.PAUSE+1

PS.RCH

Used by the programmer if DF.MLP is defined. It will request the channel number contained in MCP+2 (1502) from the SMM system. If the program has not separated a call to PS.RCH by a call to PS.DCH, this call becomes a do nothing.
Example: RJM PS.RCH+1

PS.DCH

Operates the same as a call to PS.RCH, except it releases the channel to the SMM system.
Example: RJM PS.DCH+1

PS.CPMSG

Displays a message on the left screen only. If DF.MLP has been defined, the message will be sent to the PP's message buffer in central memory. Also, PS.CPMSG calls PS.PAUSE and control is given back to the program. If DF.MLP is used, the message to be displayed may be any length as long as each line of the message contains its own display coordinates. The entire message must terminate with a 12-bit word of zero.

If the program is to run under SCOPE, the message must not exceed a length of 20 words. Refer to PS.MSGS for dayfile messages under SMM.

```
Example 1:          LDC     DIS1  (For long SMM message only.)
                    RJM     PS.CPMSG+1

            DIS1    DATA    H*RUNNING  SEC1*
                    DATA    6000B, 7200B
                    DATA    H*STATUS = XXXX*
                    DATA    6000B, 7300B
                    DATA    H*TEST ALL UNITS*
                    DATA    0

Example 2:          LDC     DIS2   (For SMM or SCOPE.)
                    RJM     PS.CPMSG+1

            DIS2    DIS     ,*RUNNING SEC2*
```

PS.MSG

Performs the same operation as PS.CPMSG, except PP memory is displayed on the right screen. Refer to PS.DCH for dayfile messages under SMM.

Example: RJM PS.MSG+1

PS.MSGS

Performs the same operation as PS.MSG, except PS stays in the display loop until the spacebar is hit or a GO is received under system operation. Messages to PS.MSGS should be biased by 10000B to permit the SMM dump routine to transfer messages to the line printer as well as the display.

```
Example:            LDC     DIS1+10000B
                    RJM     PS.MSGS+1

            DIS1    DATA    H*ERROR STOP*
                    DATA    0
```

PS.OVL

Loads the overlay specified in PS.MN and PS.E from central memory into the PP, starting from the contents of the A register upon entry to PS.OVL. Five words must be left available in front of the area the overlay is to be loaded into. These five words contain the name of the overlay that is loaded.

```
Example:        LDC     2R8G    Load overlay 8GA.
                STD     PS.MN
                LDC     0100B
                STD     PS.E
                LDC     FWA-5
                RJM     PS.OVL+1
                LJM     FWA
```

Each overlay must be defined by mnemonic in an overlay table. The overlay name table must be identified by tag name PS.OVLT. The table must terminate with a VFD 24/OLDUM and a zero word. †

```
Example: PS.OVLT    VFD     24/OLMNE
                     :
                    VFD     24/OLDUM
                    DATA    0
```

Each PP reserves 7000B central memory words for overlays.

PS.END

Idles PP and terminates program.

PS.BP1

This option is used when a breakpoint address is desired in a PPU program. Enter memory location 76 with the address to breakpoint and location 77 with the address to begin execution of the program. To execute the breakpoint option, enter 0221 W 0222. Care must be taken so that breakpoint is not in the second word of a two-word instruction. Also, special attention should be used when the program has overlays, since initialization overlays can be loaded before the section to be breakpointed.

KEY LOCATIONS USED BY PS

PS.PPN

Contains the PP number the program is currently running in.

†Overlay names in the table must be in the same order as they appear on tape.

PS.PPIR

Contains the PP input register pointer to central memory.

PS.PPOR

Contains the PP output register pointer to central memory.

PS.PPMB

Contains the PP message buffer pointer to central memory.

TERMS TO DEFINE BEFORE CALLING PS

LIST PS

To save time and space, PS is not normally listed at the front of a PPU program. To list, define LISTPS.

Example: LISTPS EQU 1

USEBP

To select the breakpoint option, define USEBP.

Example: USEBP EQU 1

DF.MLP

To multiprocess, define DF.MLP.

Example: DF.MLP EQU 1

DF.OVL

PS will load the overlays listed in PS.OVLT into central memory.

Example: DF.OVL EQU 1

NOTE

DF.OVL is illegal if DF.PPO is defined.

DF.PPO

If DF.PPO is defined, the program will load into PPO and will not multiprogram or load overlays (PS=1000B).

Example: DF.PPO EQU 1

DF.NIDL

If DF.NIDL is defined, the program will not idle all PPs, just PP10.
Example: DF.NIDL EQU 1

DF.1015

If DF.1015 is defined, PS will select the MAC in PS.RCH and deselect the MAC in PS.DCH. The MAC (1015) code is in MCP+5 in the form 3X00B, where X is the access. WA will be displayed when the access is locked out.
Example: DF.1015 EQU 1

NOTE

When DF.1015 is defined, PS will not display channel status or process sequential storing.

DF.OVMT

To load overlays from tape with no use of central memory, define DF.OVMT and DF.MLP. Overlay calls are made the same as in PS.OVL.

KEYBOARD CONTROLS

1. A keyboard entry of 0221X0222 will activate the breakpoint option if DF.OVMT is not defined.
2. An S will stop the program.
3. A space will start the program.
4. An R will restart the program from location tagged START if DF.MLP is defined.
5. PS displays 400_8 words of PPU memory in four matrices of 10 x 10 words. Memory display occurs whenever the program relinquishes control to PS at address PS.MSG or PS.MSGS. The matrices or fields are independent and their display areas can be set by keyboard entry. To set field 0, type:

   0200 X 20AA (cr), where AA is address/100.

   Similarly, for fields 1, 2, and 3:

   0204 X 20AA (cr)
   0210 X 20AA (cr)
   0214 X 20AA (cr)

6. Enter memory.

   a. Enter AAAA X VVVV (cr), where AAAA is address and VVVV is value.

   b. If format is correct, value VVVV is stored at address AAAA. If format is incorrect, entry is ignored.

   c. If + is substituted for X, sequential storing is activated.

SUMMARY

1. DF.MLP should be defined whenever possible for the following reasons:

   a. Program will run with or without SMM system (auto).

   b. Minimum changes required to make program run on line.

   c. Program may be deadstarted into a PPU from cards.

   d. May display many messages.

   e. May make calls to LDR to do various tasks.

   f. Provides access to the SMM dayfile dump routine, DMP.

2. The program must end with a zero word.

3. Before using a channel, a return jump to PS.RCH+1 should be done.

4. After using a channel, a return jump to PS.DCH+1 should be done.

5. Overlays must not directly reference PS.

6. Direct locations available to user are 20 through 67.

7. Since it takes considerable time to display the 400_8 words of PP memory on the right screen on calls to PS.MSG and PS.MSGS, periodic running messages should be displayed using calls to PS.CPMSG which negates the memory display.

8. To transfer data from one PPU to another, it is necessary for the receiving PPU to be at deadstart conditions or in a similar state. Assuming deadstart conditions, data is transferred by outputting over a channel.

9. The SMM loader can be utilized to load programs into several PPUs and into central memory, provided that none of the programs in the sequence load into PPU-0. That is, PPU-0 programs kill SMM.

6000 DIAGNOSTIC INPUT/OUTPUT PACKAGE (PSIO)

This I/O package is an addition to PS or PSSYS (operating system interface).

The main function of the I/O package is to protect the customer's operating system job processing from error conditions occurring while diagnostics are being run concurrent with system operation.

If channel or equipment conditions are incorrect when an I/O operation is performed, the peripheral processor may hang on the I/O instruction.

The I/O package contains instructions which sense for the proper channel conditions before and after each I/O instruction.

An error condition results in the following steps.

1. Channel deactivated if active.
2. Error message displayed in dayfile.
3. The test is terminated.

Thus, should a serious equipment or channel problem show up during diagnostic runs, the customer's jobs should not be jeopardized by maintenance action.

ASSEMBLY AND USE

Through the use of an *CALL card (that is, *CALL, PSIO), the interface package becomes part of the program, fully channel protected and ready to use.

The assembly of parts or all of the package is conditional upon symbol definition. The following are parts of the package.

1. Output (using A register)
2. Output from memory
3. Input (using A register)
4. Input to memory

Each routine used requires the following symbol definitions.

1. Output via A register: DF.WR
2. Output from memory: DF.WRM
3. Input via A register: DF.RD
4. Input to memory: DF.RDM

If any routines are used, the channel number must be inserted into the package I/O instructions.

1. The first address of the channel table is PS.CTBL.
2. The table ends with a zero entry.

To use the I/O package, do a return jump to the routine name +1 with parameters properly set.

1. Connect routine (assumes 6681 selected).
   a. Name: PS.CON.
   b. Input parameter: A = connect code
   c. Output parameter: none
   d. Special functions: checks 6681 status for reject or transmission parity error
2. Function routine (assumes equipment connected).
   a. Name: PS.FNC
   b. Caution to user
      1) After the function code is sent out, the 6681 status is checked. If reject is up, an error message is issued and the test is aborted.
      2) To prevent a reject, the programmer has the following two options.
         a) Jump to his own wait not busy routine before entering the function routine.
         b) Allow PS.FNC to wait a specified number of seconds for a not busy condition before executing the function by placing the maximum wait time in seconds in memory location PS.MAXDL. If busy status is still up at the end of that time, an error message is issued and the test is aborted.

3) The first method is the preferred method, since the programmer can time-share the channel while his equipment is busy. The PS.FNC routine cannot release the channel while waiting because of reconnecting complications.

c. Input parameters

1) A = function code.

2) Memory location PS.RJOK≠O if a reject is expected.

3) Memory location PS.RJOK=O if function should not reject (PS.RJOK initially = O).

4) Memory location PS.MAXDL = number of seconds to wait for a not busy condition before executing function.

5) Memory location PS.MAXDL = 0 if no wait before function is desired (PS.MAXDL initially = 0).

d. There are no output parameters.

e. Special functions

1) Check 6681 status for transmission parity error.

2) Check 6681 status for reject (PS.CROK for CON).

a) If reject is up and PS.RJOK is not set, abort test.

b) If PS.RJOK≠O, exit routine whether or not a reject occurred.

3. Output via A register (assumes equipment connected).

a. Name: PS.WR

b. Input parameters:

1) First address of output stored in memory location PS.FWAO.

2) A = word count

c. Output parameter: A = equipment status

d. Special function: check 6681 status for transmission parity error.

4. Output from memory (assumes equipment connected)

a. Name: PS.WRM

b. Same as for PS.WR.

c. Same as for PS.WR.

d. Same as for PS.WR.

5. Input via A register (assumes equipment connected)

   a. Name: PS.RD

   b. Input parameters:

      1) First address for input stored in memory location PS.FWAI

      2) A = word count

   c. Same as for PS.WR.

   d. Same as for PS.WR.

6. Input to memory (assumes equipment connected)

   a. Name: PS.RDM

   b. Input parameters:

      1) Same as for PS.RD.

      2) Same as for PS.RD.

      3) Memory location PS.RDMD = 0 for a 1500 read.

      4) Memory location PS.RDMD = 1400_8 for an EOR read.

      5) PS.RDMD preset to 0.

   c. Output parameters:

      1) A = equipment status

      2) Memory location PS.SAVEA = content of A after read

   d. Same as for PS.WR.

7. Equipment status (assumes equipment connected)

   a. Name: PS.EST

   b. There are no input parameters.

   c. Output parameter: A = equipment status

   d. Special function: check status for reserved by other channel, bit 11. If set, abort test with error code of RS.

8. 6681 status (assumes 6681 selected)

   a. Name: PS.CST

   b. There are no input parameters.

   c. Output parameter: A = 6681 status

   d. Special function: check bit 2 for transmission parity error.

9. 6681 function (usually a select or deselect 6681)

   a. Name: PS.81FX

   b. Input parameter: A = function code

   c. No output parameters or special functions.

   d. No output parameters or special functions.

10. Convert 12-bit quantity to display codes.

    This is not an I/O routine, but it is needed in the I/O package for error processing. If the user is pressed for storage, 30_8 locations can be saved by using this routine.

    a. Name: PS.DISC

    b. Input parameter: A = convert quantity (XXYY0)

    c. Output parameters:

       1) A register (lower 12 bits) = display codes for XX.

       2) PS.D3 (temporary direct storage location) = codes for YY.

PSIO ERROR MESSAGES

The following general form is the main dayfile error message.

C=xxEy, kk Cssss Ezzzz, P=xxxx

xx = Channel
y = Equipment
kk = Error condition
ssss = Last 6681 status
zzzz = Last equipment status
xxxx = Address +2 in main program which referenced the routine in which the failure was detected

Error Conditions (kk)

CR - Connect Reject

6681 status shows bit 0 and/or 1 set after a connect attempt.

1. The error message is sent to the dayfile.
2. The test is terminated.

TP - Transmission Parity Error

If bit 2 of 6681 status is set anytime it is read, the following occurs.

1. The error message is sent to the dayfile.
2. The test is terminated.

IE - The channel was inactive; it should have been active and empty.

1. The error message is sent to the dayfile.
2. The test is terminated.

FE - The channel was full and active; it should have been empty and active.

1. The channel is deactivated.
2. Same as steps a and b for transmission parity error.
3. Same as steps a and b for transmission parity error.
4. Same as steps a and b for transmission parity error.

EF - The channel was empty and active; it should have been full and active. (Same action as in FE.)

IF - The channel was inactive; it should have been active and full. (Same action as in IE.)

FI - The channel was active and full; it should have been inactive. (Same action as in FE.)

EI - The channel was active and empty; it should have been inactive. (Same action as in FE.)

FR - A function was rejected. (Same action as in TP.)

RS - Bit 11 of equipment status was set, reserved by other channel. (Same action as in TP.)

SAMPLE PS CALLING SEQUENCE AND DECK STRUCTURE

The following example shows proper PS calling sequence and desired deck structure.

Card			Reference Number
	IDENT	MTX.PS.FWA	1
	PERIPH		
	TITLE	MTX 6000/3X2X/60X TAPE TEST	
PS.EQTYP	EQU	1524B	2
DF.RFL	EQU	1	3
DF.OVL	EQU	1	4
DF.MLP	EQU	1	5
DF.ANY	EQU	1	6
DF.WR	EQU	1	7
DF.WRM	EQU	1	8
DF.RD	EQU	1	9
DF.RDM	EQU	1	10
*CALL,PS			11
MCP	DATA	12B	12
MCP1	DATA	0	13
MCP2	DATA	1305B	14
MCP3	DATA	0	15
MCP4	DATA	0	16
MCP5	DATA	0	17
MCP6	DATA	2000B	18
MCP7	DATA	0	19
MCP10	DATA	1777B	20
MCP11	DATA	0	21
START	LJM	STARTX	22
*CALL,PSIO			23
STARTX	(PROGRAM BODY)		24
	DATA	0	25
	END		26

Reference Number	Explanation
1	MTX is program name. PS.FWA is a tag within PS which sets origin address = 0.

Reference Number	Explanation
2	Used by system. PS.EQTYP must be set to the value of the hardware type. In example, 1524B is display code for MT.
3	Used by system. DF.RFL definition allows usage of optional routine PS.RFL.
4	DF.OVL defines the program as a segmented program. It should, of course, only be defined if MTX has segments.
5	Permits multiprogramming under SMM.
6	May be defined if PSIO is called.
7 through 10	Permits the four I/O routines in PSIO to be assembled.
11	Causes UPDATE to insert COMMON deck (PS) into program. This can only be done if program is on a library tape which has the COMMON deck (PS). A *CALL,PSSYS would be substituted for the system interface.
12 through 21	SMM parameter words. Refer to CPC controlled CPU programs.
22	START is initial entry point to program.
23	Causes UPDATE to insert COMMON deck PSIO into program.
24	Main program body.
25	Program must terminate with a zero.
26	End of program.

CPC (SMM SYSTEM) CONTROLLED CPU PROGRAMS

1. Program must ORG at location 1.
2. A binary of the program must be produced by COMPASS in order to obtain the necessary tables required by the SMM loader.
3. The required P, RA, and FL settings will be inserted into the input exchange package at CM location 400 (600 for CPU1) upon completion of the load call for the test.
4. All registers may be changed by use of keyboard entries. (Refer to CPC entries.) A breakpoint routine is also available when using the SMM system display and CPU control program (CPC).
5. Messages may be displayed by EXC or CPC.
   a. Under EXC control, the following format must be used.
      1) Write MS G00 0000 00XX XXXX in RA+1. XXXXXX is the FWA of a display code message which terminates with a 12-bit word of 0000 and does not exceed five central memory words.
      2) Message will be displayed by EXC until a new message takes its place in RA+1.
      3) EXC clears the upper two bytes of RA+1 when the message has been accepted.
   b. Under CPC control, the following format must be used.
      1) Write MS GO 0000 00XX XXXX in RA+1. XXXXXX is the FWA of a display code message which terminates with a 12-bit word of 0000 and each new message or line must have a ++ before it starts.

         Example:
         ```
         FWA   DATA  H*SET PARAMETERS*
               DATA  H*++LOCATION 3=0002 TO STOP ON ERR*
               DIS   ,*++LOCATION 4=0000 TO LOOP ON ERR*
         ```

      2) The format used is 64 characters per line.
      3) The Y coordinates available are from 7640 to 7100, and any X coordinate may be used.

4) The message is displayed whenever the I display is selected under CPC.

5) RA+1 is not cleared when the message is recognized by CPC.

c. Whenever error messages are used, the programmer must reset all registers (used to set RA+1) back to their original value before stopping the program.

PPU CONTROLLED CPU PROGRAMS (NO SMM CONTROL)

If a PPU program is to control execution of the CPU, the programmer must understand the CPU exchange jump.

EXCHANGING INTO EXECUTION

1. A 16-word exchange package must be set up in central memory with P, RA, FL, RAECS, FLECS, and EM registers set for program execution.
2. This package will be exchanged with the current contents of the CPU registers upon execution of an exchange jump instruction.

EXCHANGING OUT OF EXECUTION

1. Before exchanging out, a 16-word exchange package must be set up so that the CPU actually quits running.
2. Stopping the CPU can be accomplished by setting the FL register to 0000 or by setting the P and RA registers to an address where 02000 AAAAA have been stored with A = address of 02000 jump.

If the PPU program is loaded by the SMM system loader, the following information is made available.

1. Central memory location 1.

   00BB TTYY PPEE 000N 00UU

   BB = The number of central memory banks in the system (2 through 100)

   TYPE = The type of central processor - 6XXX, 7XXX, or 17X

N = 0 if only CPU0 is available.
1 if CPU0 and CPU1 are available.

UU = The number of PPUs available in the system (5 through 20)

2. Present RA and FL in CM location 2.

If the PPU program uses the SMM loader to load the CPU program, the following format should be used.

1. The first word of the PPU program must be nonzero. (This is accomplished by defining DF.MLP and calling PS.)
2. The CPU program must follow the rules described in CPC controlled CPU programs.
3. Central memory location 2 should be set to the desired RA, FL, and CPU number before making the loader call.

   CM location 2 = 00FL FLFL 00RA RARA 000C
   FL = Field length
   RA = Relative address
   C = 0-CPU0, 1-CPU1

4. Write the following information into central memory locations.
   a. 101 = 0001 XXXX 0000 0000 0000
      XXXX = 0000 - rewind. If nonzero, do not rewind before loading program.
   b. 102 = MMNN EE00 0000 0000 0000
      MNE = the name of the CPU program
   c. Refer to CPC controlled CPU programs for information on exchange package.
5. Upon completion of the load, location 101 will be cleared.
6. If any additional PPU or CPU programs are to be loaded, follow the same call rules.
   a. The last PPU program loaded must have a 0000 as its first word in order to drop the SMM system.
7. If no more programs are to be loaded and a PPU program that starts with 0000 has not been loaded, send the following call to drop the SMM system.

a. Write 0005 0000 0000 0000 0000 in central memory location 101.

b. Wait until central memory location 101 clears and start execution of the program.

PPU CONTROLLED CPU PROGRAMS (WITH SMM SYSTEM (CPC) DISPLAY AND CONTROL)

1. All rules in programming CPU tests apply.
2. The SMM system (CPC) CPU control may be turned off by setting CM location 100 = CCPP CC00 0000 0001 0000
3. The SMM system (CPC) CPU control may be turned on by setting CM location 100 = CC PP CC00 0000 0000 0000

PARAMETERS FOR PERIPHERAL TESTS UNDER SMM OR SCOPE

PARAMETER SETTINGS FOR SMM

Address 1500 =	XXX1	Repeat subcondition
=	XXX2	Stop on error
=	XXX4	Stop at end of section
=	XX1X	Stop at end of test
=	XX2X	6681 mode I
Address 1501 =	1XXX	Repeat test
=	2XXX	Repeat section
=	4XXX	Repeat condition
Address 1502 =	CCEE	CC = Channel number EE = Equipment number
Address 1503 =	0XXX	Unit selection (if any)
Address 1504 =	0000	User option.
Address 1505 =	0000	MAC access code 3X00
Address 1506 =	X000	6681 select code
Address 1507 =	XXXX	Section flags for individual tests (if needed, use 1510 first)

Address 1510 = XXXX Example: First section - bit 0 of 1510

Second section - bit 1 of 1510

:

13th section - bit 1 of 1507, etc.

Address 1511 = XXXX Section flags (if needed)

PARAMETER SETTINGS FOR SCOPE

The operator then turns on the desired sense switches (if any) and types:

X.GO. (X is the CP number)

1. To stop on all errors, turn on SENSE SWITCH 1.
2. To stop at end of section, turn on SENSE SWITCH 2.
3. To stop at end of test, turn on SENSE SWITCH 3.
4. To repeat the entire test, turn on SENSE SWITCH 4.
5. To repeat a single section, turn on SENSE SWITCH 5.
6. To repeat condition, turn on SENSE SWITCH 6.

PARAMETERS FOR CPU TESTS UNDER SMM OR SCOPE

The following are standard parameter entries for PPU controlled CPU programs with no SMM system control.

1. Parameters are set by entering SXX and cleared by a CXX.
2. Entries: XX

   SE - Stop on error
   SC - Stop on section condition
   SS - Stop at end of section
   ST - Stop at end of test
   RC - Repeat condition
   RS - Repeat section
   RT - Repeat test
   P0 - Test CPU0
   P1 - Test CPU1

The following are standard parameters for CPU tests with SMM system control.

1. Central memory locations of the parameters within the program.

   2 = 00-00 - Use program supplied number to generate random numbers.
   = XX-XX - Use XX-XX to generate random numbers.
   3 = 00-00 - Use central simulator (SMM).
   = 00-01 - Use PP simulator (SCOPE).
   4 = 00-XX - Use XX as the length of the random loop.
   5 = 00-00 - Send message to SMM display or SCOPE dayfile.
   00-01 - Loop on failure.
   00-02 - Stop on error.
   00-04 - Stop at end of section.
   00-10 - Stop at end of test.
   00-20 - Repeat section.
   00-40 - Repeat test.
   00-100 - Optimize error result.
   6 = XX-XX - Run section XX corresponding to bits set.

DESIGN SPECIFICATION

RESTRICTIONS

1. Test must run under SMM and SCOPE.
2. The interface PS must be used for peripheral equipment tests.
3. The interface PSIO should be used tor 3000 equipments.
4. The programmer is responsible for building periodic communication with the system into his program. It is necessary to maintain communication so that operator control of the program is possible.
5. I/O channels are to be shared by all system programs. This must hold true of diagnostics or the system will be degraded. Before a channel may be used, it must be requested from the system. It also must be periodically returned to the system so that channel requests by other programs are honored.

6. It is important that diagnostics terminate properly so that PPU, equipment, and channels are left in good shape for the next program which uses them. Before a program terminates, it must deselect the 6681, ensure the channel is deactive, empty, returned to the system, and do a long jump to the interface at PS.END. The interface will return equipment and PPU to system.

SECTION STRUCTURING

1. Sections should be structured so as to allow any section to be executed by itself or following any other section. Section must be repeatable and, therefore, conditions must be initialized before each pass. The routine for selecting sections should allow for the addition of more sections in the future. Sections should call on common subroutines as often as possible.
2. One section should be singled out as an interrupt test although other sections may also use interrupts.
3. At least one section should check buffered data transfers.
4. Conditions must be repeatable whether or not an error occurred. Each condition should be a complete entity in itself. All initial conditions must be included in the loop. Channel and nonchannel considerations must be taken into account.
5. The sections must be written such that an R entry can be done at any time. After the R entry and a space, the test must restart and execute correctly.

STATUS

1. A full status word compare must always be used. Status should be taken and comparison of the bits made; check for those bits that are expected to change and for those bits that are not expected to change.
2. Status operation should be done in a subroutine rather than in line code.
3. Both the channel and equipment status should be checked when applicable.

4. When test is in wait mode because of positioning, I/O, timing, etc., the program should be monitoring the full status word, not only the single condition bit. Abnormal and other erroneous status conditions could be missed if this check is omitted.

HANGING

1. The test must ensure that the I/O channel is protected from hanging up. Some form of timeout, along with status checking (full word) should be used.
2. To prevent hanging the system, the test must not hang on a channel instruction (that is, FJM *,0). All rejects should use some form of timeout or retry counts along with status checking.

RETRY CONSIDERATIONS

The test should have two selectable modes of operations, where applicable.

1. Report all errors and allow normal repeat condition.
2. When detecting a data or parity error, repeat the read or write operation N times before reporting error. If retry corrected the error, update the number of retries required.

The test can repeat an erroring operation N times before reporting the malfunction. This is usually done only on reading and writing of data. The number of times the retry procedure was used must be updated and reported when the test is completed.

ERROR REPORTING

All available pertinent information should be reported in the error display. This information should be gathered at the closest possible time to the moment of the erroring condition. The following are examples of error information to report.

1. Channel conditions, all status
2. Equipment conditions, all status

3. Data written and data read
4. Timing
5. Retries
6. Address at which error was detected
7. Contents of pertinent registers
8. Interrupts selected and expected
9. Critical locations in test
10. Test name, channel, equipment, unit, and section

TIMING CONSIDERATIONS

RUNNING TIME

1. Total time to execute the entire test with the prestored parameters should be available in the test writeup.
2. Total time to execute each section of the test should be available in the test writeup.

TIMING OF OPERATIONS

1. Timing must take into consideration the differences in configurations (that is, 841's or 821's).
2. For accurate timing in critical portions of the test, it may be necessary to run this portion by itself (that is, not with other tests).

MACROS

A macro is a sequence of codes that may be called by a single instruction. In the use of macros, there are generally only two restrictions: the macro must be defined prior to the actual program code and the maximum number of parameters used by the macro is 63.

The following is a simple macro followed by a call of that macro.

```
LNRJ    MACRO     (Y,Z)
        LDN       Y
        RJM       Z+1
        ENDM
```

Using the above macro, enter the A register with a 6 and do a return jump to a subroutine called READ:

```
LNRJ    6,READ
```

In this case, the value 6 has replaced the first parameter Y, and the location READ has replaced the second parameter Z. Without the use of the macro, the code above would have been written as the following.

```
LDN     6
RJM     READ+1
```

Obviously, the macro LNRJ must be used more than once to be of any real benefit. Each additional call will result in a savings of one line on the listing and one less card to punch. This is not impressive. However, if this macro was called 400 times in the course of a program, it would shorten a listing by approximately 200 lines and save an equal number of cards. This savings will increase proportionately to the number of macros used, the number of operations performed, and the frequency with which they are called.

The following are three macros, followed by a macro call to one which will also use the other two.

```
LNSD        MACRO     (Q,T)          Load no address/store direct macro.
            LDN       Q
            STD       T
            ENDM
RJZJ        MACRO     (C,D)          Return jump/zero jump macro.
            RJM       C
            ZJN       D
            ENDM
CONNECT     MACRO     X
            LNSD      0,CE           Call macro LNSD.
            LDM       TABLE-1,B6
            NJN       *+3
            LJM       X
            RJZJ      CONN+1,*-9     Call macro RJZJ.
            ENDM
```

Now, a call to the macro CONNECT is done.

```
CONNECT     ZIP
```

This one call would then result in the following instruction generation.

```
LDN         0             Enter A with 0.
STD         CE            Store at location CE.
LDM         TABLE-1,B6    Load memory.
NJN         *+3           Jump forward three if A is not 0.
LJM         ZIP           Jump to ZIP if A=0.
RJM         CONN+1        Return jump to CONN.
ZJN         *-9           Jump reverse 9 if A=0.
```

In the example above, it would require approximately two calls to the macro CONNECT to reach a break-even point so far as lines of print and number of cards saved is concerned. Each successive call will eliminate six lines of code. Add to that the fact that the macros LNSD and RJZJ are still available for other use, and the advantages become more obvious.

The following example helps demonstrate the versatility of the macro. It is an illustrative example, and as such, would not be practical.

```
SUPER     MACRO     (S,T,V,W,X,Y)
          S         T
          V         W
          X         Y
          ENDM
```

Now, a call to the macro SUPER is done.

```
SUPER     [LDM,(ZIP,B6),STD,ZAP,ZJN,ZOP]
```

This would result in the following instructions being generated.

```
LDM       ZIP,B6
STD       ZAP
ZJN       ZOP
```

A second call to the macro SUPER may be the following.

```
SUPER     (LPN,10B,NJN,ZUP,LJM,ZEP)
```

Again, this would result in the following instructions being generated.

```
LPN       10B
NJN       ZUP
LJM       ZEP
```

An entire program could be written using just one macro such as SUPER. The final listing and the cards used would be just one-third of that which was written without the macro. Also, during the writing of a program, it is necessary to obtain new listings which reflect changes and/or additions which have been made in the development of that program. Through the use of macros, the turn-around time required for such assemblies and the listings can be greatly reduced. If at any time an expansion of the macros is desired, a LIST card may be inserted into the program and the actual machine instructions will be generated.

Following are two LIST card examples.

```
    LIST      L,R,M
or  LIST      L,R,G
```

The preceding description and examples should help the customer engineer to better understand the layout and application of macros. The macro is a valuable programming tool, and as such, will be used more in the future.

SMM CATALOGS

To obtain a catalog of the deadstart tape, load program LST into a PP and run. Catalog will be displayed on the display. Listing is available with DMP.

Example:

```
AUTO. cr
1.DMP,1206. cr (1206 = channel and equipment)
LST.
1.DIS.
F.
```

To obtain an UPDATE, assemble any program on the OLDPL tape.

INPUT/OUTPUT PACKAGE DIAGNOSTIC (PSIO)

OPERATIONAL PROCEDURE

RESTRICTIONS

The I/O package (for 6681 only) is an addition to PSSYS (operating system interface) or PS. The assembly of parts or all of the package is conditional upon symbol definition. The following are parts of the package.

- Output (using A register)
- Output from memory
- Input (using A register)
- Input to memory

PSIO is a common deck and may be called by other programs, after first calling the interface to be used. Also, each routine used requires the following symbol definitions.

1.	Output via A register:	DF.WR	EQU	1
2.	Output from memory:	DF.WRM	EQU	1
3.	Input via A register:	DF.RD	EQU	1
4.	Input to memory:	DF.RDM	EQU	1

If any of routines are used, the channel number must be inserted into the package I/O instructions.

1. The first address of the channel table is PS. CTBL.
2. The table ends with a zero entry.
3. Suggested code:

```
            LDN     0
            STD     D1 (temporary location)
STARTX      LDM     PS.CTBL,D1
            ZJN     STARTY
            STD     D2
            LDI     D2
            SCN     77B
            LMD     D3      D3 = channel in lower six bits
            STI     D2
            AOD     D1
            UJN     STARTX
STARTY      ....
```

LOADING PROCEDURE

Not applicable.

PARAMETERS

To use the I/O package, do a return jump to the routine name + 1 with parameters properly set.

Connect Routine (Assumes 6681 Selected)

1. Name: PS.CON
2. Input parameter: A = connect code
3. There are no output parameters.
4. Special functions: checks 6681 status for reject or transmission parity error. (Refer to CR and TP.)
5. If connect rejects are expected, set PS.CROK to nonzero.

Function Routine (Assumes Equipment Connected)

1. Name: PS.FNC
2. Caution to user
   a. After the function code is sent out, 6681 status is checked. If reject is up, an error message is issued and the test is aborted (under PSSYS), not aborted under PS.
   b. To prevent a reject, the programmer has the following options.
      1) Jump to his own wait not busy routine before entering the function routine.
      2) Allow PS.FNC to wait a specified number of seconds for a not busy condition before executing the function by placing the maximum wait time in seconds in memory locations PS.MAXDL. If busy status is still up at the end of that time, an error message is issued and the test is aborted (under PSSYS), not aborted under PS.

c. The first method is the preferred method, since the programmer can time-share the channel while his equipment is busy. The PS.FNC routine cannot release the channel while waiting because of reconnecting complications.

3. Input Parameters

   a. A = function code

   b. Memory location PS.RJOK≠O if a reject is expected

   c. Memory location PS.RJOK=O if function should not reject (PS.RJOK initially = O)

   d. Memory location PS.MAXDL = number of seconds to wait for a not busy condition before executing function

   e. Memory location PS.MAXDL = O if no wait before function is desired (PS.MAXDL initially = O)

4. There are no output parameters.

5. Special functions

   a. Check 6681 status for transmission parity error. (Refer to TP.)

   b. Check 6681 status for reject.

      1) If reject is up and PS.RJOK is not set, abort test. (Refer to FR.)

      2) If PS.RJOK≠O, exit routine whether or not a reject occurred.

Output Via A Register (Assumes Equipment Connected)

1. Name: PS.WR

2. Input parameters:

   a. First address of output stored in memory location PS.FWAO

   b. A = word count

3. Output parameter: A = equipment status

4. Special function: check 6681 status for transmission parity error. (Refer to TP.)

Output from Memory (Assumes Equipment Connected)

1. Name: PS.WRM
2. Same as for PS.WR. (Refer to output via A register.)
3. Same as for PS.WR. (Refer to output via A register.)
4. Same as for PS.WR. (Refer to output via A register.)

Input Via A Register (Assumes Equipment Connected)

1. Name: PS.RD
2. Input parameters:
   a. First address for input stored in memory location PS.FWAI
   b. A = word count
3. Same as for PS.WR. (Refer to output via A register.)
4. Same as for PS.WR. (Refer to output via A register.)

Input to Memory (Assumes Equipment Connected)

1. Name: PS.RDM
2. Input parameters:
   a. Same as for PS.RD. (Refer to input via A register.)
   b. Same as for PS.RD. (Refer to input via A register.)
   c. Memory location PS.RDMD = O for a 1500 read
   d. Memory location PS.RDMD = 1400_8 for an EOR read
   e. PS.RDMD preset to 0
3. Output parameters:
   a. A = equipment status
   b. Memory location PS.SAVEA = content of A after read
4. Same as for PS.WR. (Refer to output via A register.)

Equipment Status (Assumes Equipment Connected)

1. Name: PS.EST
2. There are no input parameters.
3. Output parameter: A = equipment status
4. Special function: check status for reserved by other channel, bit 11. If set, abort test with error code of RS. (Refer to RS.)

6681 Status (Assumes 6681 Selected)

1. Name: PS.CST
2. There are no input parameters.
3. Output parameter: A = 6681 status
4. Special function: Check bit 2 for transmission parity error. (Refer to TP.)

6681 Function (Usually a Select or Deselect 6681)

1. Name: PS.81FX
2. Input parameter: A = function code
3. No output parameters or special functions
4. No output parameters or special functions

Convert 12-Bit Quantity to Display Codes

This is not an I/O routine, but it is needed in the I/O package for error processing. If the user is pressed for storage, 30_8 locations can be saved by using this routine.

1. Name: PS.DISC
2. Input parameter: A = convert quantity (XXYY)
3. Output parameters:
   a. A register (lower 12 bits) = display codes for XX
   b. PS.D3 (temporary direct storage location) = codes for YY

MESSAGES

NORMAL MESSAGES

Not applicable.

ERROR MESSAGES

The following is the general form of the main dayfile error message.

```
C=xxEy, kk Cssss Ezzzz, P=xxxx
```

xx = Channel
y = Equipment
kk = Error condition (error code)
ssss = Last 6681 status
zzzz = Last equipment status
xxxx = Address + 2 in main program which referenced the routine in which the failure was detected

Error Conditions (Error Codes)†

CR - Connect Reject

6681 status shows bits 0 and/or 1 set after a connect attempt.

1. The error message is sent to the dayfile.
2. The equipment is turned off in the operating system equipment status table.
3. The test is terminated.

TP - Transmission Parity Error

If bit 2 of 6681 status is set anytime it is read, the following occur.

1. The error message is sent to the dayfile.
2. The equipment is turned off in the operating system equipment status table.
3. The test is terminated.

†The test is not terminated when under PS usage.

IE - The channel was inactive; it should have been active and empty.

1. The error message is sent to the dayfile.
2. The equipment is turned off in the operating system equipment status table.
3. The test is terminated.

FE - The channel was full and active; it should have been empty and active.

1. The channel is deactivated.
2. Refer to steps 1 through 3 in TP.

EF

The channel was empty and active; it should have been full and active. (Same action as in FE.)

IF

The channel was inactive; it should have been active and full. (Same action as in IE.)

FI

The channel was active and full; it should have been inactive. (Same action as in FE.)

EI

The channel was active and empty; it should have been inactive. (Same action as in FE.)

FR

A function was rejected. (Same action as in TP.)

RS

Bit 11 of equipment status was set, reserved by other channel. (Same action as in TP.)

When an equipment is turned off in the operating system equipment status table, the following message is issued to the dayfile.

EQUIP xx TURNED OFF BY DIAG OPERTN

xx is the number of the equipment relative to the start of the EST.

DESCRIPTION

GENERAL

1. The main function of the I/O package is to protect the customer's operating system job processing from error conditions occurring while diagnostics are being run concurrent with system operation.
2. If channel or equipment conditions are incorrect when an I/O operation is performed, the peripheral processor may hang on the I/O instruction.
3. The I/O package contains instructions which sense for the proper channel conditions before and after each I/O instruction.
4. An error condition results in the following steps.
   a. Channel deactivated if active.
   b. Error message displayed in dayfile. (Refer to error messages.)
   c. The test equipment is turned off in the operating system equipment status table and the test terminated.
   d. Under PS. wait for operator action.
5. Thus, should a serious equipment or channel problem show up during diagnostic runs, the customer's jobs should not be jeopardized by maintenance action.

SUBROUTINE DETAILS

Connect (PS.CON)

1. Error if channel is active.
2. Mode I connect:
   a. Execute the connect function.
   b. Error if active does not drop in 100 microseconds.

3. Mode II connect:
   a. Select 6681 for connect (1000 function).
   b. Error if channel is active.
   c. Activate channel.
   d. Error if channel is inactive.
   e. Error if channel is full.
   f. Output the connect code.
   g. Error if full does not drop in 100 microseconds.
   h. Error if channel is inactive.
   i. Deactivate channel.
   j. Error if channel is active.
4. If 6681 status says transmission parity error, send dayfile message and terminate test. (Refer to TP.)
5. If 6681 status says reject, send dayfile message, and terminate test. (Refer to CR.) Check PS.CROK before error message.

Function (PS.FNC)

1. If PS.MAXDL = x, x ≠ O, wait a maximum of x seconds for a not busy condition before trying to execute the function.
2. Refer to steps 1 through 4 in connect operation.
3. If function rejects and PS.RJOK flag = O, send dayfile message and terminate test. (Refer to FR.)

Output via A register (PS.WR)

This output routine should be used in preference to PS.WRM, because channel conditions can be checked before each byte goes out. The minimum time between bytes is 16 microseconds.

1. Save word count.
2. Set up first and last byte addresses.
3. Error if channel is active.
4. Select 6681 for a 1600 output.
5. Error if channel is active.

6. Activate channel.
7. Error if channel is full.
8. Error if channel is inactive.
9. Update byte address for output. If done, go to 13.
10. Output 12-bit byte.
11. Error if channel remains full for 1 second.
12. Go to step 8.
13. Deactivate channel.
14. Error if channel is active.
15. If 6681 status says transmission parity error, send dayfile message and terminate test.
16. Set A register = equipment status and exit.

Output from Memory (PS.WRM)

1. Save word count.
2. Insert first output address.
3. Error if channel is active.
4. Select 6681 for a 1600 output.
5. Error if channel is active.
6. Activate channel.
7. Error if channel is inactive.
8. Error if channel is full.
9. Word count to A register.
10. Initiate output.
11. Error if channel remains full for 1 second after last byte goes out.
12. Refer to steps 13 through 16 for PS.WR.

Input Via A Register (PS.RD)

This input routine should be used in preference to PS.RDM, because channel conditions can be checked before and after each byte comes in. The minimum time between bytes is 16 microseconds.

1. Save word count.
2. Set up first and last word addresses.
3. Error if channel is active.
4. Select 6681 for a 1500 input.
5. Error if channel is active.
6. Activate channel.
7. Error if channel is inactive.
8. Error if channel does not go full in 1 second.
9. Input and store 12-bit byte.
10. Update byte address for input. If not done, go to 7.
11. Error if channel is inactive.
12. Deactivate channel.
13. Error if channel is active.
14. If 6681 status says transmission parity error, send dayfile message and terminate test.
15. Set A register = equipment status and exit.

put to Memory (PS.RDM)

1. Save word count.
2. Insert first input address.
3. Error if channel is active.
4. Select 6681 for input:
   a. If memory location PS.RDMD=O, use 1500 read.
   b. If memory location PS.RDMD=1400_8, use end of record read.
5. Error if channel is active.
6. Activate channel.
7. Error if channel is inactive.
8. Error if channel does not go full in 1 second.
9. Word count to A register.
10. Initiate input.

11. Save A register content after read complete.
12. Error if channel is full after last byte comes in.
13. If A≠0, go to 16. (EOR should have deactivated the channel.)
14. Error if channel is inactive.
15. Deactivate channel.
16. Error if channel is active.
17. If 6681 status says transmission parity error, send dayfile message and terminate test.
18. Set A register = equipment status and exit.

Equipment Status (PS.EST)

1. Error if channel is active.
2. Request equipment status via 6681 by a 1300 function.
3. Error if channel is active.
4. Activate channel.
5. Error if channel is inactive.
6. Error if channel is empty.
7. Input and save status.
8. Error if channel is not empty in 100 microseconds.
9. Error if channel is inactive.
10. Deactivate channel.
11. Error if channel is active.
12. Error if bit 11 is set (reserved by other channel).
13. Set A register = equipment status and exit.

6681 Status (PS.CST)

1. Error if channel is active.
2. Request 6681 status by a 1200 function.
3. Same as for PS.EST, steps 3 through 11.
4. If 6681 status says transmission parity error, send dayfile message and terminate test.
5. Set A register = 6681 status and exit.

6681 Function (PS.81FX)

1. Insert function code.
2. Error if channel is active.
3. Execute function.
4. Error if channel is active.

Error Processing

1. Convert last equipment status into display code.
2. Convert last 6681 status into display code.
3. Insert error condition into display.
4. Convert channel and equipment into display code.
5. Convert address + 2 (in main program which referenced failing routine) to display code.
6. If channel is active, deactivate it.
7. Send message to dayfile (RJM MSG+1).
8. All error conditions terminate the test.
   a. Deactivate the channel if active.
   b. Check for match of channel and equipment with equipment status table.
   c. If match, set off bit in EST (bit 23).
   d. Send dayfile message EQUIP xx TURNED OFF BY DIAG OPERTN (RJM PS.MSG+1).
   e. End test by LJM PS.END (for PSSYS.).
   f. Return to program for PS.

Suggestions for Usage

1. Connect routine code:

```
CON       LJM    *
EQUIPN    LDC    *  Equipment (EXXX)
          RJM    PS.CON+1
          UJN    CON
```

2. Equipment status routine code:

```
STATUS    LJM    *
          RJM    CON+1  Ensure connect
          RJM    PS.EST+1
          STD    STAT
          UJN    STATUS
```

Storage Used

The basic package, exclusive of the routines listed, is the following.

PS.WR: 66_8 locations
PS.WRM: 65_8 locations
PS.RD: 65_8 locations
PS.RDM: 77_8 locations

Processing Constant Busy Pulse on Status Lines

1. Most peripheral tests require waiting for an equipment to go not busy before continuing.
2. Most wait not busy routines release the channel to the operating system and request it again before the next status check. Normally, this is satisfactory, since the equipment will drop busy in a short time.
3. However, it is possible for the equipment to generate a false nonterminating busy pulse, and then hang.
4. Thus, a good policy is to allow the equipment plenty of time to go not busy (for example, 500-1000 release channels), and if it is still busy, then issue an error message and abort the test.
5. Peripheral tests using the 6000 diagnostic I/O package may place an error code in the A register (for example, 0223,BS) and do an LJM to PS.ERAC2 to process such an error.

RUNNING OPTIONS AND PARAMETERS FOR 3000 PERIPHERAL TESTS UNDER SMM

A

CALLING TEST

Type:	X.MNE.	
or:	X.MNE,CCOE,	where CC = channel and E = equipment
or:	MNE.	
or:	MNE,CCOE,	where CC = channel and E = equipment

PARAMETER SETTINGS

Address 1000 or 1500	= XXX1	Repeat subcondition
	= XXX2	Stop on error
	= XXX4	Stop at end of section
	= XX1X	Stop at end of test
	= XX2X	6681 mode I
Address 1001 or 1501	= 1XXX	Repeat test
	= 2XXX	Repeat section
	= 4XXX	Repeat condition
Address 1002 or 1502	= CCEE	CC = Channel number EE = Equipment number
Address 1006 or 1506	= X000	6681 select code
Address 1007 or 1507	= XXXX	Section flags for individual tests (sections 13 through 24)
1010 or 1510	= XXXX	Example: First section - bit 0 of 1010 or 1510 Second section - bit 1 of 1010 or 1510 . . . Twelfth section - bit 11 of 1010 or 1510, etc.
Address 1011 or 1511	= XXXX	Section flags for sections 25 through 36.

RESTARTING AFTER STOP

1. Hitting the spacebar will make the program continue sequentially.
2. To restart the test completely, type R. or type 0221 W 1012, or 1512. 1512 or 1012 is the starting address for most tests.

STOPPING THE TEST

1. In addition to the options for stopping, the operator may type an S and the program will stop.
2. The test must be prestarted by typing R or 0221 W XXXX, because the operator has stopped the sequential flow of the program.

TO SET BREAKPOINT

1. Enter location 0076 with breakpoint address and type 0221 W 0222.
2. When the program stops with B displayed, set 0076 to the next breakpoint and space.

NOTE

It is impossible to breakpoint at the same address twice in succession.

RUNNING OPTIONS FOR SCOPE PERIPHERAL PROGRAMS

CALLING TEST

PPU Programs

MNE,XXXX.

Test MNE is called and address MCP+10 is set to XXXX.

MNE(ZZZZXX,XXYYYY).

Test MNE is called and address MCP+10 is set to YYYY, address MCP+7 is set to XX00, and address MCP+11B is set to ZZZZ.

MNE(ZZXX,YYYY).

Test MNE is called and address MCP+10 is set to YYYY, address MCP+7 is set to XX00, and MCP+11 is set to 00ZZ.

MNE.

Call test with all preset parameters selected.

CPU Programs

MNE,W,X,Y,Z.

Test MNE is called. Arguments W, X, Y, and Z are placed in RA+2, 3, 4, and 5, respectively. Up to 50 arguments may be entered, depending on how many arguments test is set up for.

OPTIONS

1. All tests have an initial parameter stop:

   The operator then turns on the desired sense switches (if any) and types:

   GO, or X.GO. (X is the CP number).

   To stop on all errors, turn on SENSE SWITCH 1.
   To stop at end of section, turn on SENSE SWITCH 2.
   To stop at end of test, turn on SENSE SWITCH 3.
   To repeat the entire test, turn on SENSE SWITCH 4.
   To repeat a single section, turn on SENSE SWITCH 5.
   To repeat a condition, turn on SENSE SWITCH 6.

SMM AND SCOPE PARAMETER REFERENCE TABLE

STOP/REPEAT FLAGS

Standard Parameter	Set Condition	Standard Tags	SMM Address	SCOPE Sense Switch
Repeat Subcondition	XXX1	MCP	1000 or 1500	----
Stop on Error	XXX2	MCP	1000 or 1500	1
Stop End of Section	XXX4	MCP	1000 or 1500	2
Stop at End Test	XX1X	MCP	1000 or 1500	3
Repeat Test	1XXX	MCP+1	1001 or 1501	4
Repeat Section	2XXX	MCP+1	1001 or 1501	5
Repeat Condition	4XXX	MCP+1	1001 or 1501	6

OTHER PARAMETERS

Standard Parameter	Parameter Position	Standard Tag	SMM Address	SCOPE Interface
Channel Number	PPXX	MCP+2	1002 or 1502	EST entry
Equipment Number	XXPP	MCP+2	1002 or 1502	EST entry
6681 Number	PXXX	MCP+6	100 or 150	EST entry
Test Section Flags 1 through 12	PPPP	MCP+10	1010 or 1510	Test Call Parameter
Test Section Flags 13 through 24	PPPP	MCP+7	1007 or 1507	Test Call Parameter

Standard Parameter	Parameter Position	Standard Tag	SMM Address	SCOPE Interface
Relative Address/100	OOPP	PS.RA	N/A	Through PP Resident
Field Length/100	OOPP	PS.FL	N/A	Through PP Resident
Control Point Address	OoopPP	PS.CPA	N/A	Through PP Input Register
Logical Equipment Number	OOPP	PS.LUN	N/A	EST Entry Ordinal

CROSS REFERENCE OF PPU NUMBERING SCHEMES USED BY DIFFERENT TESTS

B

AUTO DDD DMP(LP) PC1	CH2 CHT IP1 PCM PM1 PMM PSP SSP	CH1 MAP PRW SCOPE DMP (DISP)	SCOPE D.S. DUMP	Physical Channel Numbering	Physical PPU Numbering
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
10	10	8	8	10	10
11	11	9	9	11	11
				12	
				13	
				14 (RTC)	
				15 (INT. REG)	
				16 (SCR)	
12	20	10	A	20	0
13	21	11	B	21	1
14	22	12	C	22	2
15	23	13	D	23	3
16	24	14	E	24	4
17	25	15	F	25	5
20	26	16	G	26	6
21	27	17	H	27	7
22	30	18	I	30	10
23	31	19	J	31	11
				32	
				33	

GLOSSARY C

The contents of this appendix consists of abbreviations and minimum definitions of key words found in this volume.

AUTO	Multiprogramming mode.
CED	Deadstart diganostic.
CEL	Deadstart display routine.
CEQ	Deadstart diagnostic call for CED as it appears on OLDPL tape.
CM	Central memory.
COMPASS	Comprehensive assembly language; basic language used in SMM diagnostics.
CPU	Central processor unit.
(CR)	Carriage return.
CPC	Central processor control; display and CPU control reside in PPU 10.
DDD	Dump routine; dump to tape or to printer.
DMP	CM and PPU to printer dump routine; operates under auto.
D.S. or D/S	Deadstart.
DSCL	Deadstart loader card.
EM	Emergency maintenance.
ENS	Loader program that allows only PPU diagnostics to be loaded with no keyboard entry; loads an octal record (diagnostic).
EXC	Executive, list of CPU and CM diagnostics run from a PPU under multiprogramming mode.
FL	Field length CM.
FLX	Field length ECS.
FWA	First-word address.
FWA-1	First-word address minus one word.
LDR	Loader main program routine; resides in PPU 0.
MA	Monitor address exchange package.
MASS STO	Mass storage.
MNE	Mnemonic of instruction set.

OLDPL	Old program library; the update tape.
PPU	Peripheral processor unit.
PTL	Loader card used to deadstart tapes on channel 4, 5, 6, or 7.
RA	Relative address CM.
RAX	Relative address ECS.
SMM	System maintenance monitor.
50 TABLE	Identifies record as a CPU record.
70 TABLE	Name of the record and its contents. (Refer to system software manual for description.)

COMMENT SHEET

MANUAL TITLE CDC 6000/CYBER 70/CYBER 170 SMM (Volume 1 of 3)
Reference Manual

PUBLICATION NO. 60160600 REVISION A

FROM: NAME: _____

BUSINESS ADDRESS: _____

CUT ALONG LINE

AA3419 REV. 7/75 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD

FOLD

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

Litho in U.S.A.