

# TDC 4500 PROCESSOR

## MAINTENANCE

### 1. INTRODUCTION

This publication provides information to assist in the installation, troubleshooting and repair of the Processor function of a TDC 4500 Process Control System. The Processor consists of two PWA's (Printed Wiring Assemblies) plugged into card slot positions 5 and 6 of the CPU chassis, which is located in the Central System's cabinet of the system. Virtually all areas of the system are directly or indirectly connected to the Processor through backpanel connections. (See Fig. 3.) The two PWA's of the Processor interconnect through front-edge connectors. Processor connections are summarized by Figs. 1 and 2 of this publication, which include references to pin-connection summaries.

### 2. PRECAUTIONS

The Processor boards contain only logic-level voltages, but to avoid damage to the logic, power must be removed prior to their removal or replacement.

### 3. OPTIONS

The Processor is a functional requirement of the Central Processing Unit (CPU) portion of a TDC4500 system. Other required functions of the CPU include the Memory Bus Controller (MBC), the GENIE\* Bus Controller (GBC) and Memory. The Memory Protect Unit (MPU) and Floating Point are considered optional portions of the CPU.

### 4. REFERENCES

TDC 4500 Processor Theory publication, number ACPU1PROC-T, provides an overall description of the Processor's operation.

TDC 4500 General Description, number PTH-019, provides a general description of the Processor and other areas of the system.

Instruction Reference Manual, publication PTS-046, provides a format description for each of the executable Processor instructions, with references to their interruptibility.

\*Trademark

TDC 4500 Instruction Test, drawing number 51191048, provides an operational test of the Processor.

ATPG Programs - How to Load Them, Run Them, and Use Them, publication ATPG-1, provides assistance in interpreting and operating the Instruction Test.

### 5. COMPONENT LOCATIONS

The Processor is located in card slots 5 and 6 of the CPU chassis of the CPU cabinet.

### 6. TEST EQUIPMENT AND MATERIALS

No special test equipment is required. Due to the complexity of the PWA's, it is recommended that board replacement philosophy be followed for field repair.

### 7. PREVENTIVE MAINTENANCE

Not required for the Processor.

### 8. PERFORMANCE TESTS

Performance of the Processor can be verified by execution of the 4500 Instruction Test, number 51191048. Instructions for operating the test are provided with the punched cards or tape. Reference to the ATPG-1 publication may prove useful. Allow the test to complete the required number of passes.

### 9. ASSEMBLY AND DISASSEMBLY

The only assembly/disassembly possible is the removal and insertion of the PWA's. Remove power during removals or insertions.

### 10. ADJUSTMENTS

The only possible adjustment for the Processor is the pin-selectable timing source at the RPCA1 PWA, which specifies the timing of Processor operations. The 150 ns pin setting is the nominal setting.

PLUA4 PWA Contains Basic Processor Registers,  
CPU Bus Interface and ALU

CPU Bus Provides Common Connection Lines  
That Processor Uses for Communication With:

- MBC
- MEMORY
- GBC
- MPU\*
- Floating Point Hardware (FPH)\*

RPCA1 PWA Provides Basic Firmware  
Control of Processor

\*Options

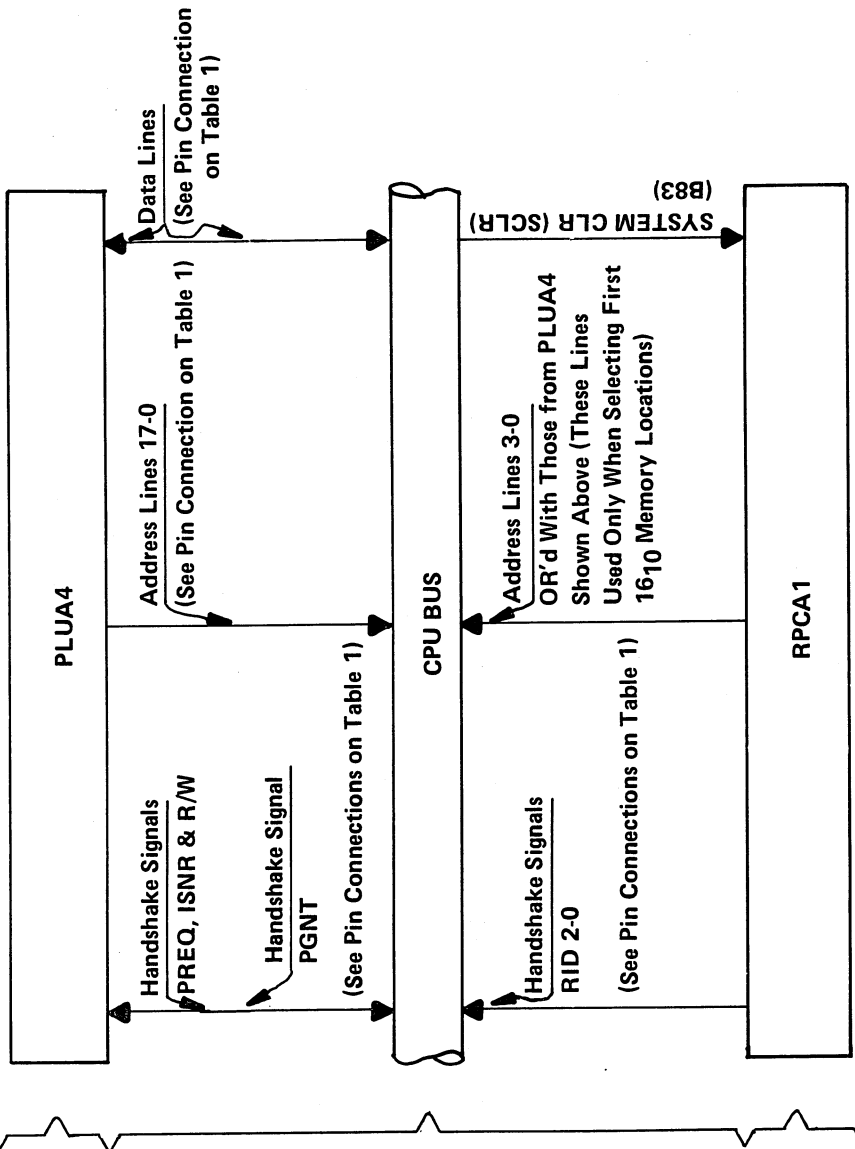


Fig. 1 CPU Bus Connections

## 11. TROUBLESHOOTING

This section provides assistance in isolating faults to a board area. The Processor PWA is not considered field-repairable and replacement of suspected bad boards with known good ones is recommended. The following information is intended to assist in determining the correct PWA to replace.

### 11.1 General Troubleshooting Philosophy

The Processor is the primary director and initiator of system actions. As a general philosophy, the more basic and widespread the failure indications, the more likely the Processor is at fault. However, the firmware control of the Processor performs a basic check during the initialize function and will halt sequencing if it does not pass. This halt will lock out any attempt by the user to communicate through the Programming and Maintenance Console. Should you be able to communicate through the console following an initialization, you can assume a certain amount of firmware control is available.

Within the remainder of the troubleshooting section are illustrations and text references to alarm conditions and traps. These are the most obvious indication of failures.

11.2 provides information on the system alarms and coordinates a flow chart approach to locating the source of an alarm during the power up, initialize, bootstrap load and memory coat functions. (Table 2 indicates the desired response to these functions.) 11.3 supplies information on traps.

For symptoms not supplying the obvious indication of an alarm or trap, it is necessary to use hand loops or the Instruction Test, if operational, for assistance in determining the fault source. Also refer to the Instruction Reference Manual and Processor Theory publications, where necessary. Refer to the Alarm Indications, 11.2, for information on alarm conditions encountered while troubleshooting.

### 11.2 Alarm Indications

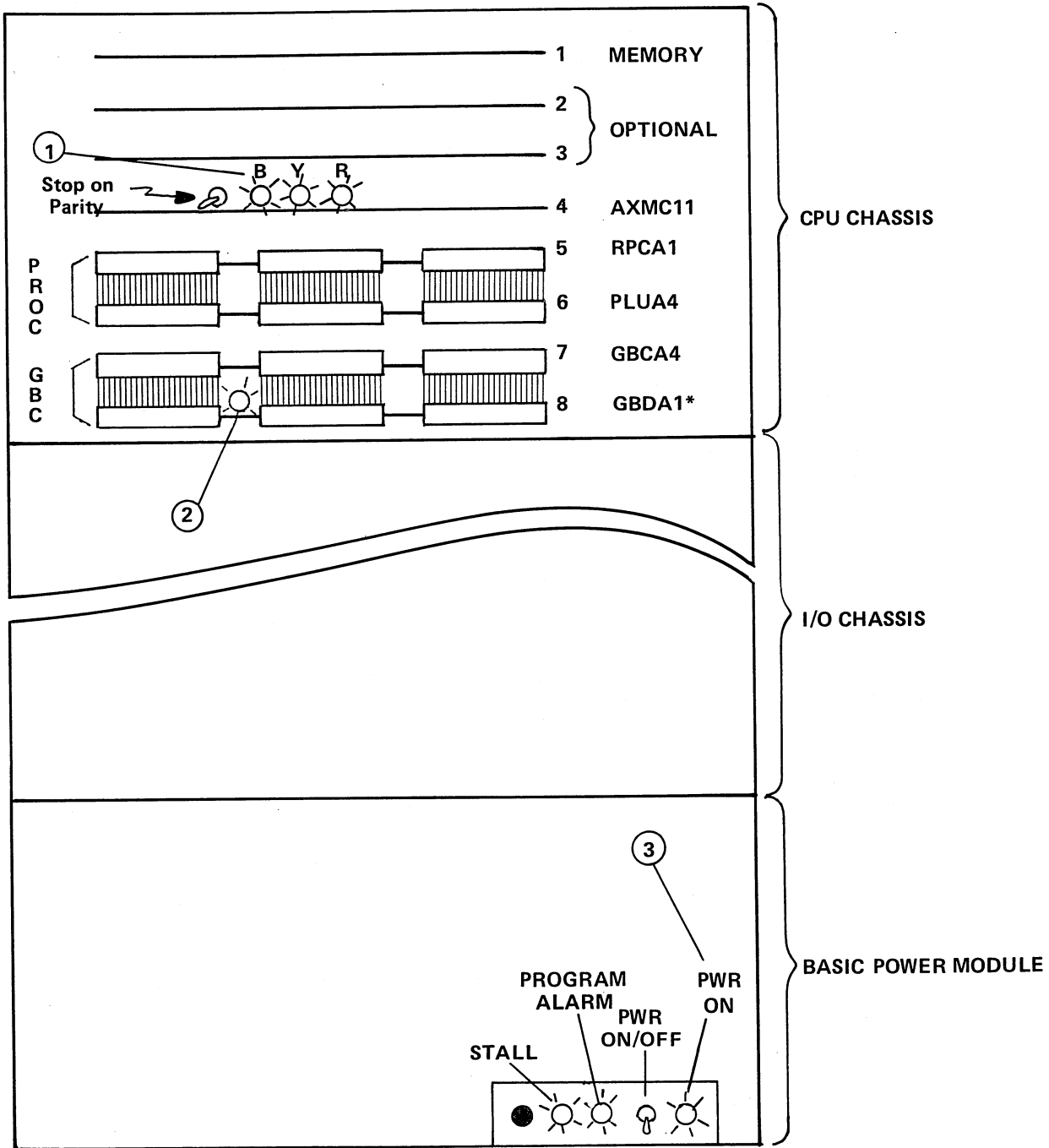
This description of alarm indications refers to those listed on Figs. 3 and 4. Fig. 3 shows their physical locations in the CSU cabinet and Fig. 4 shows their functional interconnections. As indicated by Fig. 4, the Alarm Light at the Programming and Maintenance Console is a result of any of the alarms listed. Several indicators are available at the MBC board, the GBC, and Basic Power Module to specifically indicate which alarm is present.

	FUNCTION	SHOULD CAUSE –	IF NOT –
	POWER-UP	With Battery Backup } Trap to location 22g and perform response based upon (Auto Restart) } macro-instruction there. Subsequent action varies.	See Fig. 7.
	(POWER ON SELECT)	Without Battery } A → -1 (All F's); CS → 0; Backup } PC → End Memory Address - 153g (Start of boot)	See Fig. 7.
See Note if Console Disabled	INITIALIZE (RESET/0) WITH CONSOLE ENABLED	A → 0; CS → 0; PC → End memory address 153g (Start of boot) Console in Halt with I L/O & S L/O lights lit and no alarms.	See Fig. 5.
	MEMORY COAT OPERATION (RESET/7) WITH CONSOLE ENABLED	A → 7g; Bootstrap routine to memory; CS → 0; PC → End memory address - 153g (start of boot). Console in Halt with I L/O & SL/O lights lit and no alarms. Enter pattern in CS register & hit Run to execute. Pattern coats to within 15g locations of memory end (64 K maximum).	See Fig. 6.
	BOOTSTRAP (RESET/1-6) WITH CONSOLE ENABLED	Keypad select function # → A; CS → 0; PC → End memory address - 153g Start of boot). Console in Halt with I /LO & S /LO lights lit and no alarms. Hit Run to execute bootstrap.	See Fig. 6.

Note: An automatic Bulk Bootload performed if a Reset/x function selected while in the Console Disable mode. Console disable clears I L/O, S L/O & Halt following SCLR.

Table 2 Basic Power-Up/Initialize/Bootstrap Functions

**FRONT VIEW**  
(Behind Front Door of CSU Cabinet)



- ① B = Blue Timeout Alarm; Y = Yellow Memory Error; R = Red Memory Error  
(See ACPU1MBC-T, Sh. 8)
- ② Indicates GBC Timeout or Device Controller Alarm.  
(See ACPU1GBC-M)
- ③ PWR ON = +5 V  
(See ACPU1GBC-M for PRG ALM and STALL)

Fig. 3 CPU Alarm Locations

\* AXGBD1 in later processors.

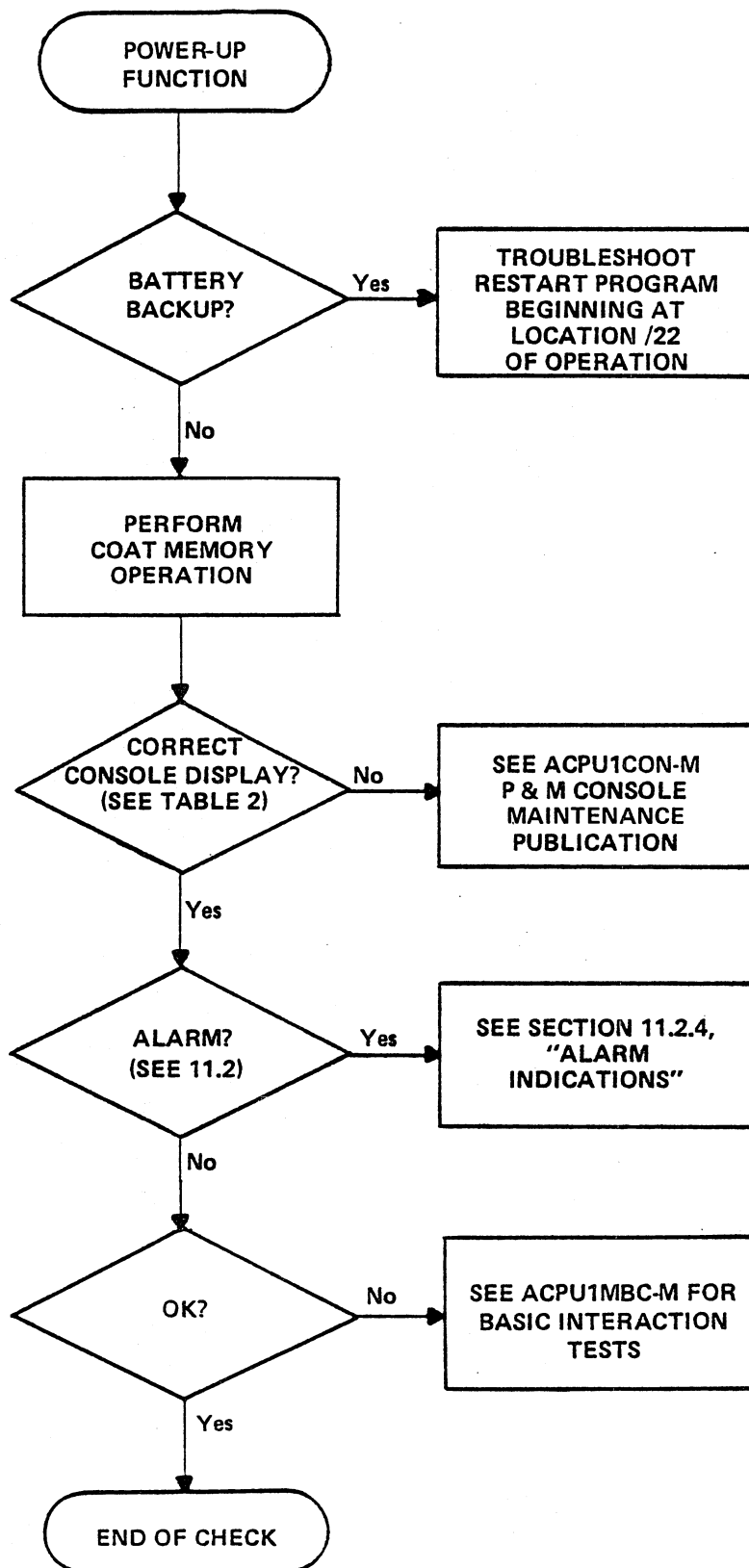


Fig. 7 Power-Up Function Check

### 11.3 Trap Indications

Trap operations provide indications of important events of failures. The following is a list of the trap locations and their cause. Following the causes, in parentheses, are areas most probably at fault when a trap occurs.

<u>Location</u>	<u>Cause</u>
20g	Memory Protect Violation (PROGRAM)
22g	Power Fail Restart (PS/PROC)
23g	Illegal Instruction (PROGRAM/PROC)
24g	Interrupt Watchdog (GBC/PROC)
25g	Memory Error (on CPU Transaction) (MBC)
26g	GENIE Bus Arbitration Error (GBC)
30g	Memory Error (on Interrupt Service) (MBC)
31g	SPB Memory Protect Violation (MPU)

The information saved in locations 21g and 27g is:

<u>Trap</u>	<u>Location 21g</u>	<u>Location 27g</u>
20	Address of last valid instruction, also status information	Not meaningful
22	Undefined	Undefined
23	Illegal Instruction	P Register
24	Not meaningful	P Register
25	Status	P Register
26	Not meaningful	P Register
30	Status	P Register
31	Address of SPB which caused trap	Address generated by SPB

Most traps are discussed in the theory publication for the function to which they apply; Illegal instruction traps are defined here. An illegal instruction is any of the following when Memory Protect is shut off:

Any GEN 1 instruction with bits 11, 12, and 13 set.

Any GEN 1 with bit 8 reset and bits 5 or 6 set and bits 9 or 10 set.

Any GEN 2 with bits 14-0 in the range 00000 through 03777 except IAI2 and LPM.

A Byte instruction referencing a Byte point with bits 22 and 23 set.

Any instruction with op-code 74 and bits 12 and 13 set.

When Memory Protect is enabled, the following instructions will cause an illegal instruction trap:

Any instruction which is illegal with Memory Protect off.

Any external GEN 2.

JND or LPM.

Any Control Bus instruction (op-code 27g)

The illegal instruction is stored in memory location 21g and the Trap instruction in location 23g is executed.

## 12. PARTS

The Processor boards are considered non-repairable in the field and are, therefore, considered replaceable parts:

- PX4000RPCA1\*
  - PX4000PLUA4
  - 51101580-100 - Front-edge PWA connectors
- } Processor PWA's

\* The RPCA1 is further numbered 51102008-XXX where XXX specifies the type/model/quantity of firmware (ROMs) present on the board.

# APPENDIX A

## 4DP3AACPUxxx Model

The following pages contain a listing of the Bootstrap Routine. The routine is loaded by firmware control into the upper memory area following selection of a power-up without battery backup, or the selection of a bootstrap load or memory coat function (Reset with 1-7 from the Programming and Maintenance Console).

### NOTE

The locations shown for the instructions are relative. The last location of the routine will appear in the last memory location. The first instruction (Entry Point) will appear 152g locations from the end of memory.



Bootstrap - Bulk Memory Coater

02070  
02080  
02090  
02100  
02110  
02120  
02130  
02140  
02150  
02160  
02170  
02180  
02190  
02200  
02210  
02220

```

* Command table - Goes at end of memory
*
TABLE
FOR LINKF,0          -1 - POWERFAIL ENTRY
FOR LINKB,0          0 - RESTART (NORMAL REBOOT) ENTRY
FOR CARD,/4002      1 - CARD READER
FOR TAPE,/4003      2 - PAPER TAPE READER
FOR BULK,/4004      3 - PRIMARY STORAGE
FOR BULK,/4005      4 - SECONDARY STORAGE
FOR BADD,/0006      5 - TERTIARY STORAGE
FOR COAT,/0000      6 - RESERVED
FOR COAT,/0000      7 - COAT MEMORY
* (This address must end in /77 or you did something wrong!)
*
*

```

```

01
02
03
04 003767 00704004 00704004
05 003770 01004004 01004004
06 003771 00004002 00004002
07 003772 00604003 00604003
08 003773 01004004 01004004
09 003774 01004005 01004005
10 003775 01004006 01004006
11 003776 03100000 03100000
12 003777 02300000 02300000
13
14
15
16

```

```

* TTL Symbol Analysis
* END

```

```

17 004000 *00000000 *00000000
THERE WERE 063 PAL STATEMENTS IN ERROR IN THE ABOVE ASSEMBLY
003 X INDEX SPECIFICATION ERR.
060 I ILLEGAL OPERAND

```

## 4DP3BACPUxxx Model

The following pages list the Bootstrap Routine for the improved model 4500. This listing should be used when the RPCA1 board is marked 51102008-108 or -109.

The principle difference between standard and improved model 4500s is in the Bootstrap Loader area. Firmware for the BACPU model will, in certain cases as explained below, try to load from an alternative GENIE Bus address if the selected device fails to operate properly. If the second device also fails to work, loading attempts will alternate between the two addresses. Loading attempts halt if either device loads successfully or if the hardware is reset.

The two pairs of alternate addresses are:

4004g	(RESET and 3, or Auto-Reboot)
4005g	(RESET and 4)
4006g	(RESET and 5)
4007g	(RESET and 6)



01	003007	32000006	32000006				
02							
03							
04							
05							
06							
07	003010	41003622	41040012				
08	003011	63000016	63000016				
09	003012	41003624	41040012				
10	003013	32000045	32000045				
11	003014	44000022	44000022				
12							
13	003015	00603770	00640153				
14	003016	32000003	32000003				
15	003017	05000062	05000062				
16	003020	32000007	32000007				
17	003021	14703717	14740076				
18							
19							
20	003022	42000045	42000045				
21	003023	27032400	27032400				
22	003024	00074000	00074000				
23	003025	14000016	14000016				
24							

\* Rootstrap entry point  
 \* SIA 6  
 \* We now must disable power fail recovery routine (until software has completed load or whatever.) Do this by putting linkage to reset in location /16 and putting BRU /16 in 22.  
 \* DLD RSEI  
 \* DST /16 and put in lower memory  
 \* DLD RSEI+2  
 \* STA /45  
 \* SIQ /22  
 \* Complete bootstrap  
 \* LDA TABLE,6  
 \* SIA 3  
 \* SRL 18  
 \* SIA 7  
 \* BRU 60,7  
 \* Constants for System Clear  
 \* RSEI LDR /45 /16 - Perform a  
 \* CON 0,27032400 /17 - a System Reset  
 \* CON 0,00074000 /45 - Pattern to cause reset  
 \* BRU /16 /22 - Powerfail entry  
 \* SLW

00350  
 00360  
 00370  
 00380  
 00390  
 00400  
 00410  
 00420  
 00430  
 00440  
 00450  
 00460  
 00470  
 00480  
 00490  
 00500  
 00510  
 00520  
 00530  
 00540  
 00550  
 00560  
 00570  
 00580

01	TIL	Bootstrap - Card/Tape Loader		00590
02	* A	CARD/TAPE LUADER - Console switches contain relocation address		00600
03	C0	EOL *		00610
04		LDK /20021	Enable code (for paper tape)	00620
05	003026	00003026	start card/tape motion	00630
06	003027	40020021	make word count = "infinity"	00640
07	003030	25320000	initialize column count	00650
08	003031	32000006	Start processing next data record	00660
09		07500000		00670
10		00003032	Initialize checksum to zero	00680
11	003032	05000000		00690
12		00003033	Checksum OK?	00700
13	003033	05004770	No. Checksum error	00710
14	003034	34003034	CK = checksum	00720
15			X4 = last load address	00730
16			X5 = column number	00740
17				00750
18				00760
19	003035	32003006	Yes. Set Checksum to zero	00770
20		00003036	Head number words in record	00780
21	003036	33003000	Does record start past column 72?	00790
22	003037	24537667	Yes. Skip rest of card	00800
23	003040	34003036	Number words = zero?	00810
24	003041	05024654	No. Not end file	00820
25	003042	30003044	Yes. Set end-file flag	00830
26	003043	26501750	(next call to read will skip rest of media)	00840
27			Is data to be relocated?	00850
28	003044	05070713	align number of words	00860
29	003045	05000054	plus one	00870
30	003046	32000006	Get load address	00880
31	003047	33003000	No relocation. Use address as absolute	00890
32	003050	34003052	Yes. Add console switch relocation value	00900
33	003051	11003005	Initial load address	00910
34	003052	32000004	Is count too large to be valid?	00920
35	003053	24637753	Yes. Must have had data	00930
36	003054	34003054	X5 = Column Number	00940
37			No. Read data words and store	00950
38			Get next data word	00960
39	003055	33003000	store data word	00970
40	003056	32400000	and increment address	00980
41	003057	26400001		00990
42				01000
43				01010
44			"Fall into" READ subroutine. We return to last SPR READ + 1	01020
45			to process next data character read. This continues until X6 = 0,	01030
46			when we exit to F0.	01040

01050  
01060  
01070  
01080  
01090  
01100  
01110  
01120  
01130  
01140  
01150  
01160  
01170  
01180  
01190  
01200  
01210  
01220  
01230  
01240  
01250  
01260  
01270  
01280  
01290  
01300  
01310  
01320  
01330  
01340  
01350  
01360  
01370  
01380  
01390  
01400  
01410  
01420  
01430  
01440  
01450

```

* * READ Subroutine - Get 24-bits of data from input device
*
* * X2 = used internally
* * X3 = device number
* * X5 = "Column" counter (incremented by 2)
* * X6 = Number words left in data record
* * X7 = 0-cards/6-paper tape
*
* * HEAD LDX 7,2      Lards = 0 / Tape = 6
* * RI    ERL *
* *      JCR 0,3      At end of media?
* *      BRU F0       Yes. Check for end-file
* *      JDR 0,3      No. Data ready?
* *      BRU R1       No. Wait for it
* *      IN 0,3       Yes. Input next frame
* *      TXH 1000,5   Did we detect zero-length record?
* *      BTR R1       Yes. Skip rest of card
* *      SLL 12,7     No. Process this frame
* *      DRC 12,7     Cards = 12 Paper tape = 1A
* *      STA 0        and accumulate data word in 0
* *      INX -3,2     Done 2 or 4 frames?
* *      BTR R1       cards = -3/tape = 3,0,-3
* *      INX 2,5      No. Head another
* *      ERA CK       Yes. Count 2 "columns"
* *      DMT 6        Compare with checksum
* *      IER 23       Processed entire field?
* *      RTS *+2      Yes. Go test checksum = field checksum
* *      ADD CK       No. Checksum sign = data?
* *      SRL 23       (Restore data + - sign)
* *      ADD 0        Yes. No overflow
* *      STA CK       No. Add Checksum
* *      LDA 0        Isolate sign
* *      LPR 1        and add to data
* *      SLW         and checksum
* *      SLW         update checksum
* *      SLW         Exit with data
*
* * Process data word
* * INX 2,5
* * ERA CK
* * DMT 6
* * HTR E1
* * IER 23
* * ERA CK
* * RTS *+2
* * ADD CK
* * SRL 23
* * ADD 0
* * STA CK
* * LDA 0
* * LPR 1
* * SLW

```

01	003060	16200007	16200007	X	16200007	16200007
02	003061	00003661	00003661		00003661	00003661
03	003062	25360020	25360020		25360020	25360020
04	003063	14040032	14040032		14040032	14040032
05	003064	25360040	25360040		25360040	25360040
06	003065	14003661	14077775		14003661	14077775
07	003066	25350000	25350000		25350000	25350000
08	003067	24536030	24536030		24536030	24536030
09	003068	34003661	34077772		34003661	34077772
10	003069	45702014	45702014		45702014	45702014
11	003070	45705314	45705314		45705314	45705314
12	003071	32000010	32000010		32000010	32000010
13	003072	24237767	24237767	X	24237767	24237767
14	003073	26237775	26237775	X	26237775	26237775
15	003074	30003661	30077764		30003661	30077764
16	003075	26500002	26500002		26500002	26500002
17	003076	10003606	10077707		10003606	10077707
18	003077	06000006	06000006		06000006	06000006
19	003078	30003633	30077732		30003633	30077732
20	003079	05045627	05045627		05045627	05045627
21	003080	10003606	10077703		10003606	10077703
22	003081	34003706	34040002		34003706	34040002
23	003082	11003606	11077701		11003606	11077701
24	003083	05000067	05000067		05000067	05000067
25	003084	11000010	11000010		11000010	11000010
26	003085	11003606	11077676		11003606	11077676
27	003086	32003606	32077675		32003606	32077675
28	003087	00000010	00000010		00000010	00000010
29	003088	35000001	35000001		35000001	35000001
30	003089					
31	003090					
32	003091					
33	003092					
34	003093					
35	003094					
36	003095					
37	003096					
38	003097					
39	003098					
40	003099					
41	003100					



```

01 ITL          Hootstrap = Bulk Loader
02
03 * Restart initial entry = From power failure
04 LINKF -EGL *-GU
05 003726      05004637 05004637
06
07
08 * Entry for manual reset
09 LINKB -EGL *-GN
10 * Bulk memory initial entry
11 BULK -EGL *-GU
12 BULKR FOL *
13 LDK 0,3
14 CBK 0
15 STA CK
16 STA 3
17 ORA JNRK
18 STA JNR
19 LRM 6
20 OUT 0,3
21 LDZ
22 OUT /10,3
23 JNE /4000
24 BRU RULKR
25 OPR 0,3
26 AKA TIMER
27 LDX CK,3
28 JNR
29 BRU ABORT
30 ASS 1
31 JNE 0,3
32 BRU TLOOP
33 BRU 0
34
35 *
36 ABRT 0,3
37 JNR 0,3
38 BRU RULKR
39 BRU *-2
40
41 *
42 LLOUPS EQL 4000000/17
43 TIMER EUL /3777777/LLOUPS
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

01680
01690
01700
01710
01720
01730
01740
01750
01760
01770
01780
01790
01800
01810
01820
01830
01840
01850
01860
01870
01880
01890
01900
01910
01920
01930
01940
01950
01960
01970
01980
01990
02000
02010
02020
02030
02040
02050
02060
02070

```

```

Enter bulk boot w/ talt set

Pick up the bulk controller address
and change to the alternate device.
Save the controller address to be used after
the transfer.
Build a hard JNR instruction in case the
transfer changes X3.
Set transfer length
to 64 words
Set starting address
to 00000
Check if the controller exists.

Initiate transfer of the bootstrap.
Wait 2 seconds for the transfer to complete.
Restore controller address in X3.
JNR serves as a JNR 0 and check for timeout.
Time has expired. JNR stored here.
Transfer OK?
No. Timeout, then abort.
Yes. Enter record just loaded

Abort the controller.
Wait for the controller to come ready again.
When this controller is ready try the other.
Loop until ready.

2000000/8.5 Enough times through the loop
for 2 seconds to expire.

```



```

01 003767 07004005 07004005
02 003770 10004005 10004005
03 003771 00004002 00004002
04 003772 06004003 06004003
05 003773 10004005 10004005
06 003774 10004004 10004004
07 003775 10004007 10004007
08 003776 10004006 10004006
09 003777 41000000 41000000
10
11
12
13
14
15
16
17
18
19
20
21 004000 *00000000 *00000000
THERE WERE 063 PAL STATEMENTS IN ERROR IN THE ABOVE ASSEMBLY
003 X INDEX SPECIFICATION ERR.
060 1 ILLEGAL OPERAND

```

```

* Command table - Goes at end of memory
*
62 DEF /77,1H
61 DEF /77777,0
TDC DEF 0,62,61
*

```

```

TDC LINKF,/4005
TDC LINKR,/4005
TDC CARD,/4002
TDC TAPE,/4003
TDC BULK,/4005
TDC RULK,/4004
TDC RULK,/4007
TDC RULK,/4006
TDC COAT,/0000
-1 - POWERFAIL ENTRY = (BOOT PRIMARY STORAGE)02310
0 - RESTART (NORMAL REBOOT) ENTRY
1 - CARD READER
2 - PAPER TAPE READER
3 - PRIMARY STORAGE
4 - SECONDARY STORAGE
5 - TERTIARY STORAGE
6 - QUATERNARY
7 - CUAL MEMORY
4004 (4005 alternate)02350
4005 (4004 alternate)02360
4006 (4007 alternate)02370
4007 (4006 alternate)02380
02340
02350
02260
02270
02280
02290
02300
02310
02320
02330
02340
02350
02360
02370
02380
02390
02400
02410
02420
02430

```

```

* (This address must end in /77 or you did something wrong!)
*
* TTL Symbol Analysis
* END

```

```

* (This address must end in /77 or you did something wrong!)
*
* TTL Symbol Analysis
* END

```

---

## TABLE OF CONTENTS

1.	INTRODUCTION .....	1
2.	PRECAUTIONS .....	1
3.	OPTIONS .....	1
4.	REFERENCES .....	1
5.	COMPONENT LOCATIONS .....	1
6.	TEST EQUIPMENT AND MATERIALS .....	1
7.	PREVENTIVE MAINTENANCE .....	1
8.	PERFORMANCE TESTS .....	1
9.	ASSEMBLY AND DISASSEMBLY .....	1
10.	ADJUSTMENTS .....	1
11.	TROUBLESHOOTING .....	5
11.1	General Troubleshooting Philosophy .....	5
11.2	Alarm Indications .....	5
11.3	Trap Indications .....	12
12.	PARTS .....	12
	APPENDIX A .....	A-1

# TDC 4500 PROCESSOR

## MAINTENANCE

### 1. INTRODUCTION

This publication provides information to assist in the installation, troubleshooting and repair of the Processor function of a TDC 4500 Process Control System. The Processor consists of two PWA's (Printed Wiring Assemblies) plugged into card slot positions 5 and 6 of the CPU chassis, which is located in the Central System's cabinet of the system. Virtually all areas of the system are directly or indirectly connected to the Processor through backpanel connections. The two PWA's of the Processor interconnect through front-edge connectors. Processor connections are summarized by Figs. 1 and 2 of this publication, which include references to pin-connection summaries.

### 2. PRECAUTIONS

The Processor boards contain only logic-level voltages, but to avoid damage to the logic, power must be removed prior to their removal or replacement.

### 3. OPTIONS

The Processor is a functional requirement of the Central Processing Unit (CPU) portion of a TDC4500 system. Other required functions of the CPU include the Memory Bus Controller (MBC), the GENIE\* Bus Controller (GBC) and Memory. The Memory Protect Unit (MPU) and Floating Point are considered optional portions of the CPU.

### 4. REFERENCES

TDC 4500 Processor Theory publication, number ACPU1PROC-T, provides an overall description of the Processor's operation.

TDC 4500 General Description, number PTH-019, provides a general description of the Processor and other areas of the system.

Instruction Reference Manual, publication PTS-020, provides a format description for each of the executable Processor instructions, with references to their interruptibility.

\* Trademark

TDC 4500 Instruction Test, drawing number 51191048, provides an operational test of the Processor.

ATPG Programs - How to Load Them, Run Them, and Use Them, publication ATPG-1, provides assistance in interpreting and operating the Instruction Test.

### 5. COMPONENT LOCATIONS

The Processor is located in card slots 5 and 6 of the CPU chassis of the CPU cabinet.

### 6. TEST EQUIPMENT AND MATERIALS

No special test equipment is required. Due to the complexity of the PWA's, it is recommended that board replacement philosophy be followed for field repair.

### 7. PREVENTIVE MAINTENANCE

Not required for the Processor.

### 8. PERFORMANCE TESTS

Performance of the Processor can be verified by execution of the 4500 Instruction Test, number 51191048. Instructions for operating the test are provided with the punched cards or tape. Reference to the ATPG-1 publication may prove useful. Allow the test to complete the required number of passes.

### 9. ASSEMBLY AND DISASSEMBLY

The only assembly/disassembly possible is the removal and insertion of the PWA's. Remove power during removals or insertions.

### 10. ADJUSTMENTS

The only possible adjustment for the Processor is the pin-selectable timing source at the RPCA1 PWA, which specifies the timing of Processor operations. The 150 ns pin setting is the nominal setting.

PLUA4 PWA Contains Basic Processor Registers,  
CPU Bus Interface and ALU

CPU Bus Provides Common Connection Lines  
That Processor Uses for Communication With:

- MBC
- MEMORY
- GBC
- MPU\*
- Floating Point Hardware (FPH)\*

RPCA1 PWA Provides Basic Firmware  
Control of Processor

\*Options

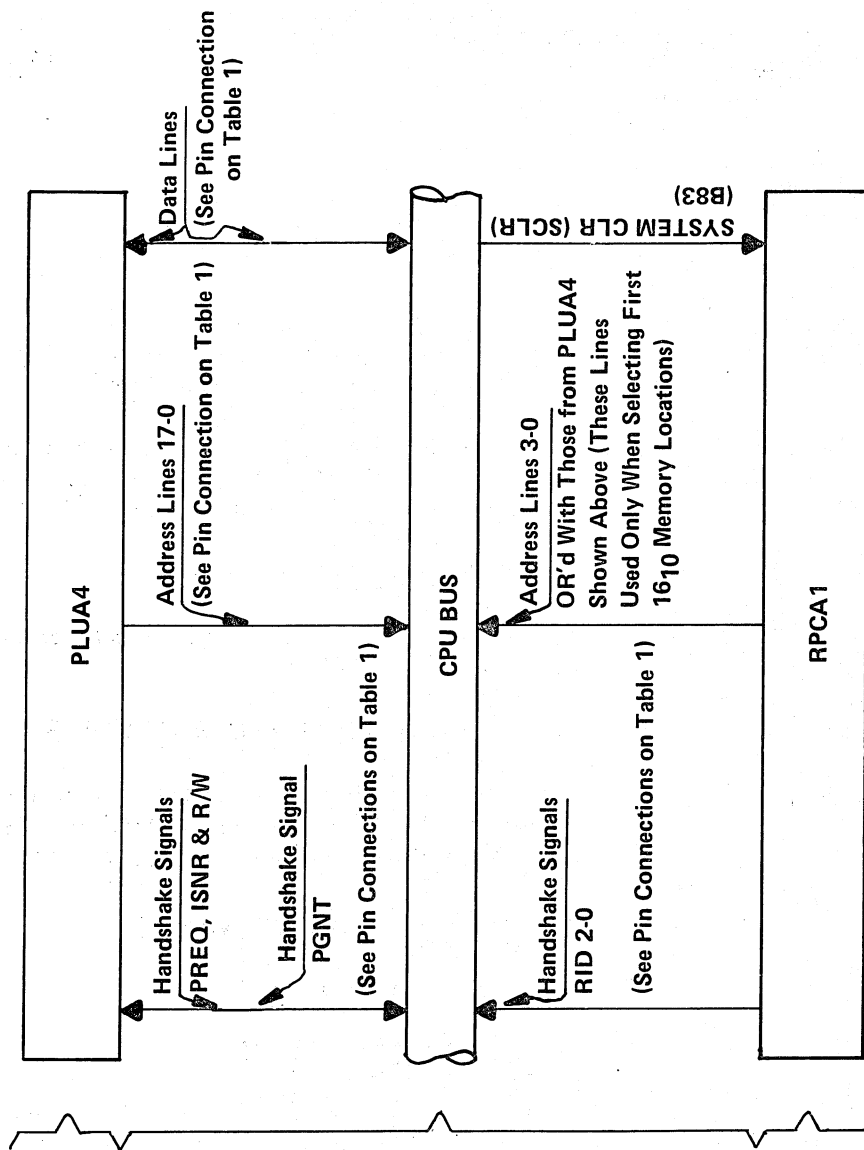


Fig. 1 CPU Bus Connections

**HANDSHAKE CONTROL (See Fig. 1)**

PREQ A34	PGNT A97	ISNR A35	RID2 A08	RID1 A58	RID0 A09	R/W A07
INITIAL ACCESS REQ & GRANT		SYNC REQ	RESPONDER ID			TRANS DIRECT

SIGNAL →  
PIN NUMBER →

**ADDRESS LINES SA17-00 (See Fig. 1)**

SA17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	SA00
A10	A60	A11	A61	A12	A62	A13	A63	A14	A64	A15	A65	A16	A66	A17	A67	A18	A68

SIGNAL →  
PIN NUMBER →

**DATA LINES 23-12 (See Fig. 1)**

DB23	22	21	20	19	18	17	16	15	14	13	DB12
A22	A72	A23	A73	A24	A74	A25	A75	A26	A76	A27	A77

SIGNAL →  
PIN NUMBER →

**DATA LINES 11-00 (See Fig. 1)**

DB11	10	09	08	07	06	05	04	03	02	01	DB00
A28	A78	A29	A79	A30	A80	A31	A81	A32	A82	A33	A83

SIGNAL →  
PIN NUMBER →

Table 1 CPU Bus Signals Generated or Used by Processor

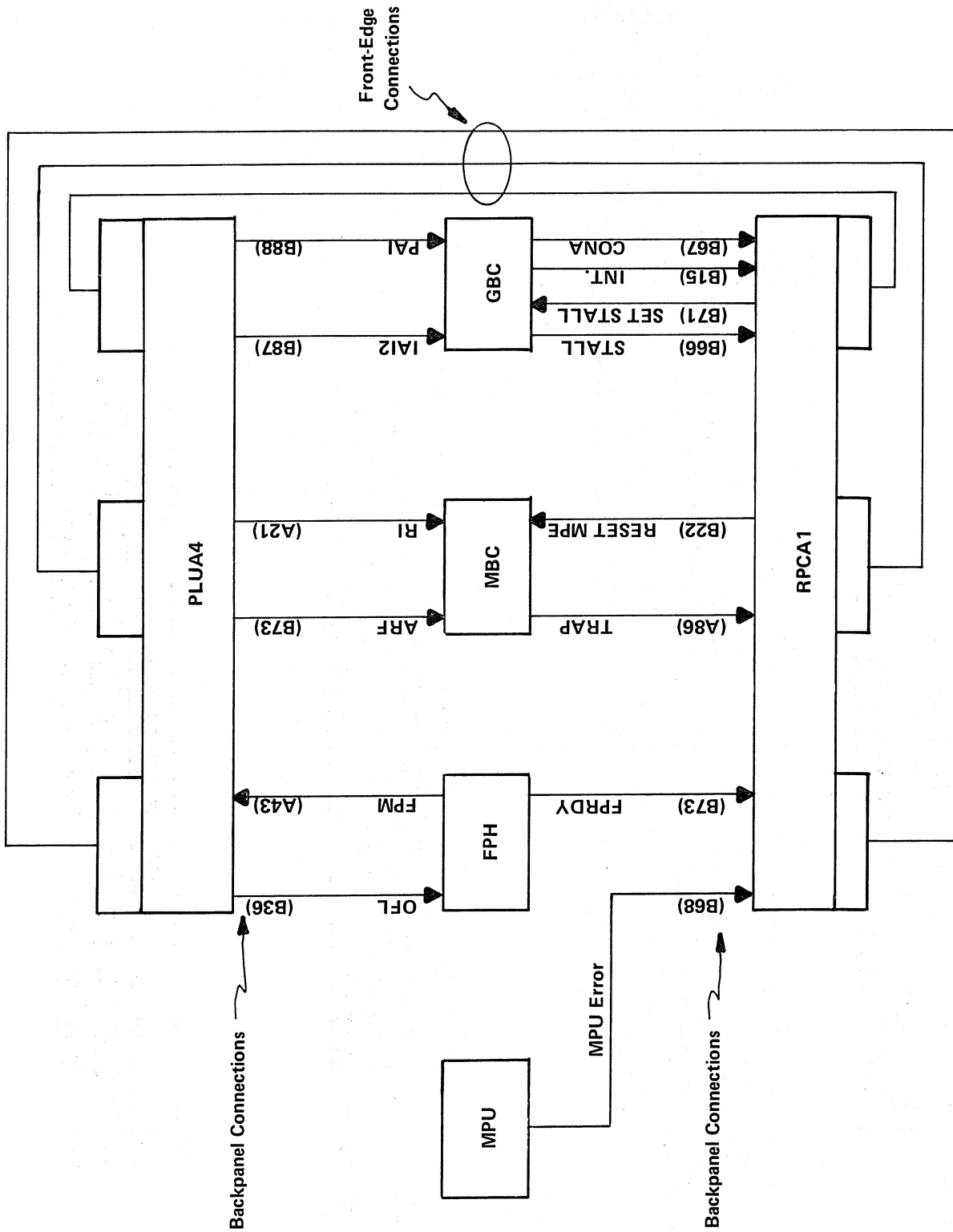


Fig. 2 Direct (Non-CPU-Bus) Connections From Processor

# 11. TROUBLESHOOTING

This section provides assistance in isolating faults to a board area. The Processor PWA is not considered field-repairable and replacement of suspected bad boards with known good ones is recommended. The following information is intended to assist in determining the correct PWA to replace.

## 11.1 General Troubleshooting Philosophy

The Processor is the primary director and initiator of system actions. As a general philosophy, the more basic and widespread the failure indications, the more likely the Processor is at fault. However, the firmware control of the Processor performs a basic check during the initialize function and will halt sequencing if it does not pass. This halt will lock out any attempt by the user to communicate through the Programming and Maintenance Console. Should you be able to communicate through the console following an initialization, you can assume a certain amount of firmware control is available.

Within the remainder of the troubleshooting section are illustrations and text references to alarm conditions and traps. These are the most obvious indication of failures.

11.2 provides information on the system alarms and coordinates a flow chart approach to locating the source of an alarm during the power up, initialize, bootstrap load and memory coat functions. (Table 2 indicates the desired response to these functions.) 11.3 supplies information on traps.

For symptoms not supplying the obvious indication of an alarm or trap, it is necessary to use hand loops or the Instruction Test, if operational, for assistance in determining the fault source. Also refer to the Instruction Reference Manual and Processor Theory publications, where necessary. Refer to the Alarm Indications, 11.2, for information on alarm conditions encountered while troubleshooting.

## 11.2 Alarm Indications

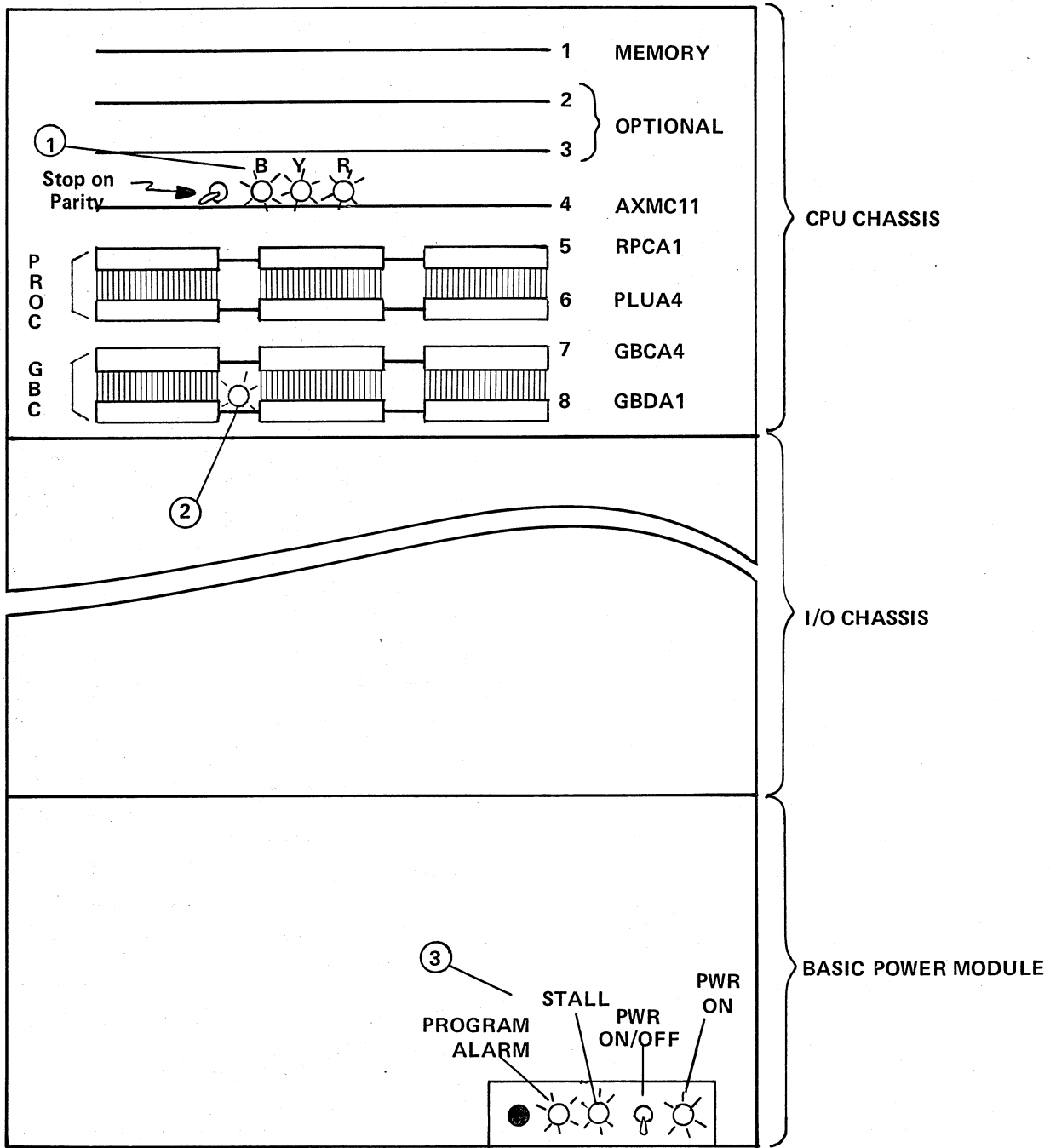
This description of alarm indications refers to those listed on Figs. 3 and 4. Fig. 3 shows their physical locations in the CSU cabinet and Fig. 4 shows their functional interconnections. As indicated by Fig. 4, the Alarm Light at the Programming and Maintenance Console is a result of any of the alarms listed. Several indicators are available at the MBC board, the GBC, and Basic Power Module to specifically indicate which alarm is present.

	FUNCTION	SHOULD CAUSE –	IF NOT –
	POWER-UP	With Battery Backup (Auto Restart) } Trap to location 22g and perform response based upon macro-instruction there. Subsequent action varies.	See Fig. 7.
	(POWER ON SELECT)	Without Battery Backup } A → -1 (All F's); CS → 0; PC → End Memory Address - 153g (Start of boot)	See Fig. 7.
See Note if Console Disabled	INITIALIZE (RESET/0) WITH CONSOLE ENABLED	A → 0; CS → 0; PC → End memory address 153g (Start of boot) Console in Halt with I L/O & S L/O lights lit and no alarms.	See Fig. 5.
	MEMORY COAT OPERATION (RESET/7) WITH CONSOLE ENABLED	A → 7g; Bootstrap routine to memory; CS → 0; PC → End memory address - 153g (start of boot). Console in Halt with I L/O & SL/O lights lit and no alarms. Enter pattern in CS register & hit Run to execute. Pattern coats to within 15g locations of memory end.	See Fig. 6.
	BOOTSTRAP (RESET/1-6) WITH CONSOLE ENABLED	Keypad select function # → A; CS → 0; PC → End memory address - 153g (Start of boot). Console in Halt with I /LO & S /LO lights lit and no alarms. Hit Run to execute bootstrap.	See Fig. 6.

Note: An automatic Bulk Bootload performed if a Reset/x function selected while in the Console Disable mode. Console disable clears I L/O, S L/O & Halt following SCLR.

Table 2 Basic Power-Up/Initialize/Bootstrap Functions

**FRONT VIEW**  
(Behind Front Door of CSU Cabinet)



- ① B = Blue Timeout Alarm; Y = Yellow Memory Error; R = Red Memory Error  
(See ACPU1MBC-M)
- ② Indicates GBC Timeout or Device Controller Alarm.  
(See ACPU1GBC-M)
- ③ PWR ON = +5 V  
(See ACPU1GBC-M for PRG ALM and STALL)

Fig. 3 CPU Alarm Locations



#### NOTE

The Alarm Reset button at the Programming and Maintenance Console clears the MBC Red and Blue Errors and clears the GBC Timeout and Device Controller alarms; it does not clear the Internal GBC Check alarms. SCLR is required to clear all alarm indications.

Refer to the ACPU1GBC-M publication for definitions of the Permit Timer, Watchdog Timer, Stall Timer, Programmable Alarm, GBC Timeout and device controller alarm detections. Refer to the ACPU1MBC-M publication for definitions of the Red and Blue Alarms.

The following descriptions of alarms pertain to specific function selections at the console.

#### 11.2.1 Alarms on Initialize (Fig. 5)

There should be no alarms remaining following an Initialize. Refer to Fig. 4 to determine the source of the alarms and troubleshoot accordingly. Note that Initialize places the console in the Halt mode with the SLO and ILO set. These lockouts should block stall and interrupt watchdog checks at the GBC.

#### 11.2.2 Alarms on Memory Coat (Fig. 6)

The Memory Coat operation is similar to a bootload function since it causes a firmware transfer of a PROM based loader program to upper memory for execution. This program, once initiated by hitting the console's Run button, executes the macro-instructions from memory and can therefore produce alarms if memory accesses do not work properly.

Should the upper memory board or MBC board not respond properly, it is possible to have Red (for data access problems) or Blue (for control access) alarms. The GBC is not activated and no interrupt control or actuation of the stall logic is initiated by this function, so there would be no alarms initiating from the GBC.

#### 11.2.3 Alarms on Bootstrap Load (Fig. 6)

Like the Memory Coat function, a bootstrap load transfers a loader into the upper memory area. Depending on the loader selected, the information loaded from a device will go into the lower memory area (beginning at location 0) or into another area specified by a relocation constant. (The bulk load always loads into lower memory, beginning at location 0.) The card or tape load sequences allow the operator to enter a relocation constant in the console's CS register prior to the transfer and will relocate the load information accordingly.

Prior to hitting Run, the first instruction of the bootstrap program is fetched. Should there be a memory or MBC problem, a Red or Blue alarm could occur, but no GBC action is initiated at this point. Once Run is selected, however, the GBC and a corresponding device controller are accessed to perform the bootstrap load. GBC alarms at this point would be expected if there is a GBC or device controller problem. Note the GBC and MBC alarm lights to determine the specific alarm condition. See Fig. 3 for their location.

The bootstrap load program is reprinted in an appendix of this publication as an aid in troubleshooting.

#### 11.2.4 Alarms on Power Up (Fig. 7)

Refer to Fig. 7 if battery backup is used on the system. For systems without battery backup, the bulk bootload sequence is performed. The program is transferred to upper memory and executed when the Run button is hit. If bulk is available and operational, the program will transfer to lower memory, beginning at location 0.

Alarms prior to Run would indicate a memory or MBC problem.

Prior to hitting Run, only the upper memory area would be accessed. If the bulk transfer takes place, the lower area of memory is accessed to load the program. Control is transferred to location 0 following its load to begin execution of the instructions. Therefore, alarms following hitting of Run could involve the lower memory area.

Attempts to access memory areas outside of the bootstrap program and that loaded by the bootstrap will typically cause Red alarms if the areas are not first coated with a pattern. (The "coat" operation establishes the correct error codes for the memory data.)

**NOTE**

If Console Disabled When Initialize Selected,  
an Auto Bulk Bootload is Performed (see Fig. 6).

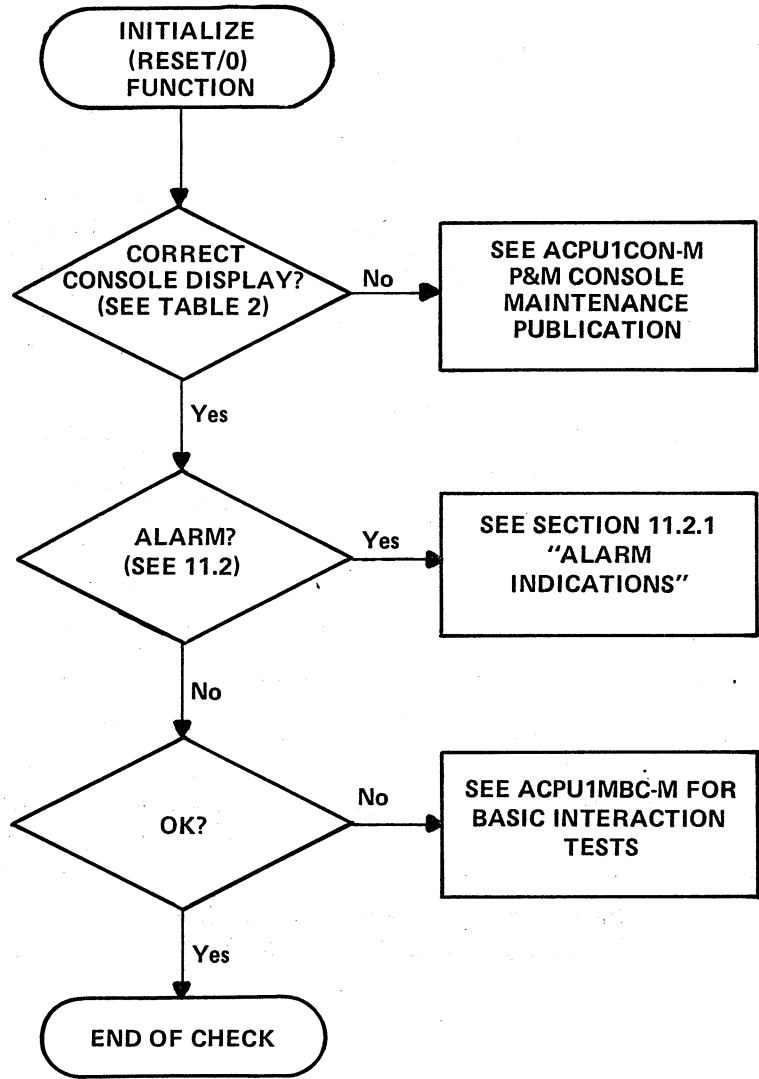


Fig. 5 Initialize Function Check

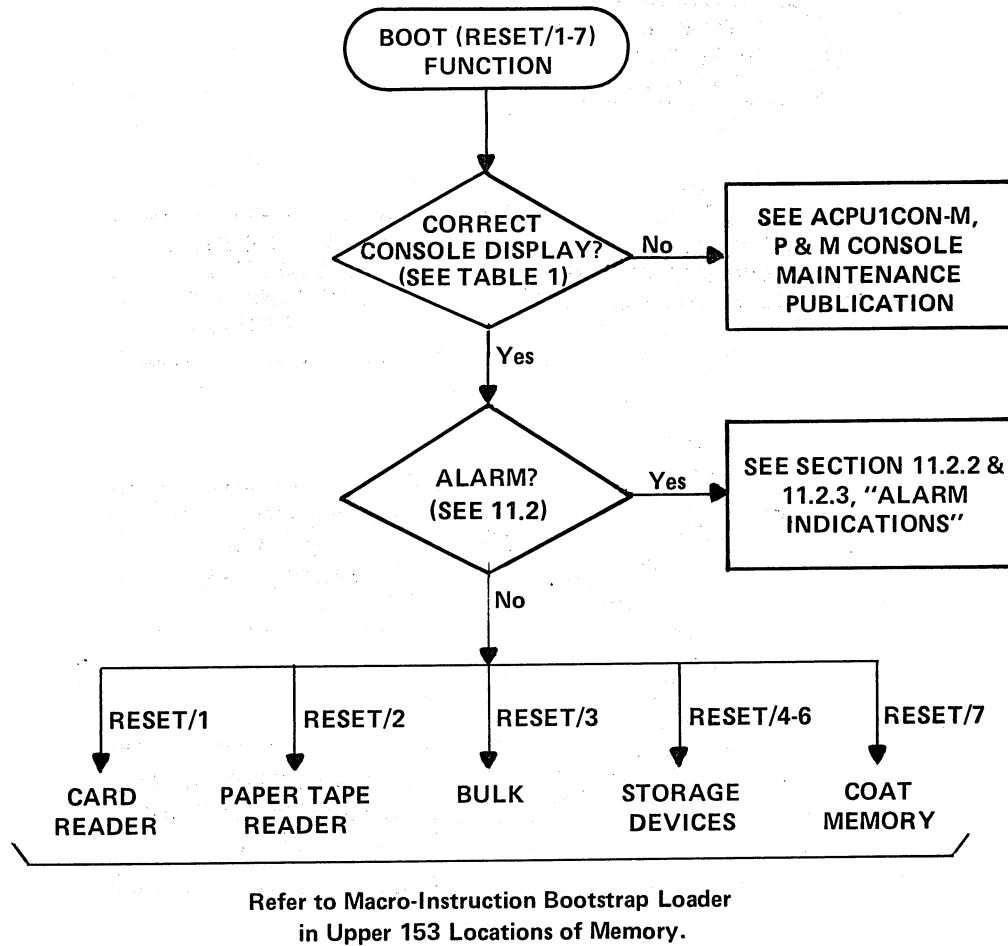


Fig. 6 Bootstrap/Memory Coat Function Check

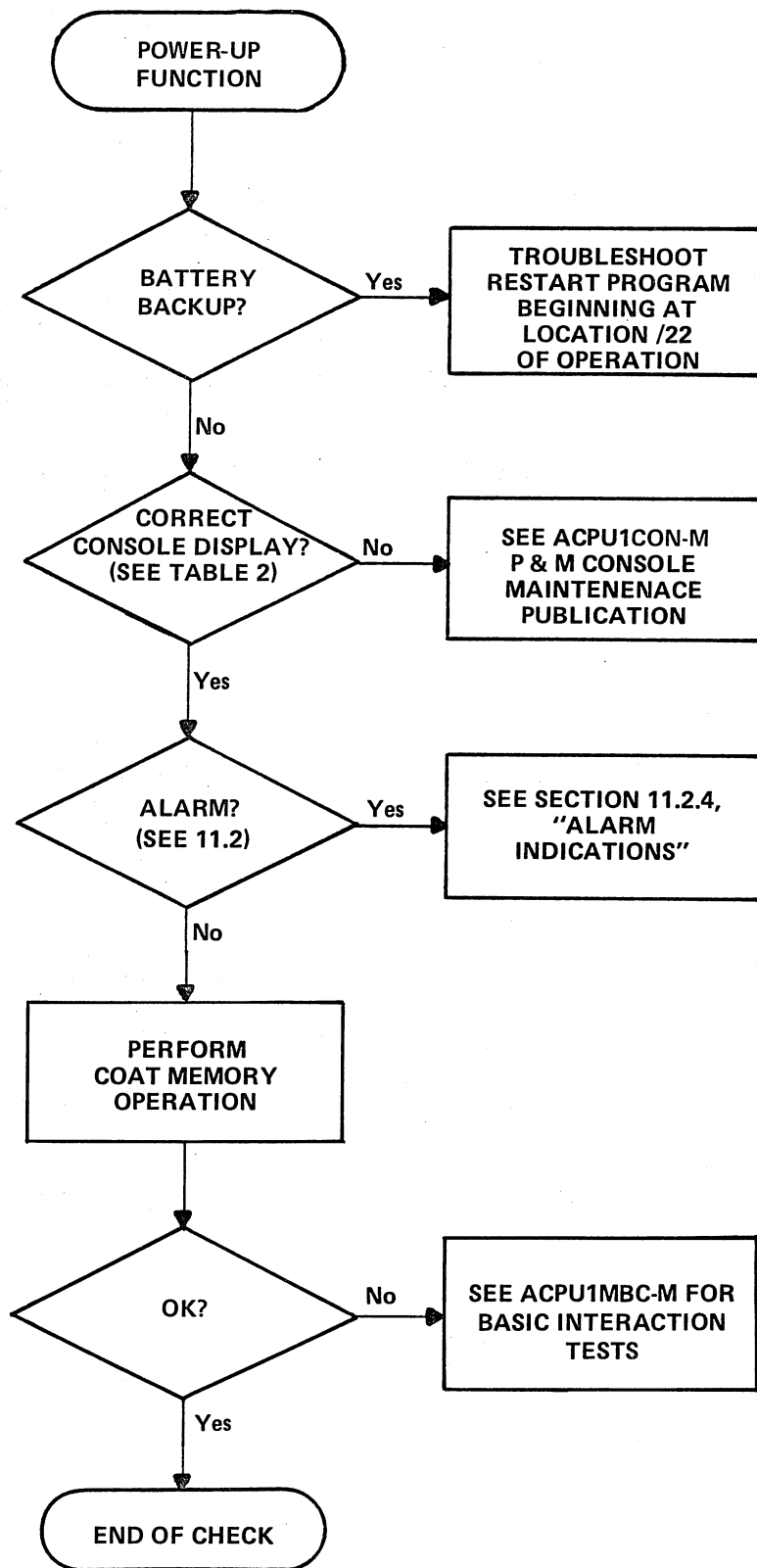


Fig. 7 Power-Up Function Check

### 11.3 Trap Indications

Trap operations provide indications of important events of failures. The following is a list of the trap locations and their cause. Following the causes, in parentheses, are areas most probably at fault when a trap occurs.

<u>Location</u>	<u>Cause</u>
20g	Memory Protect Violation (PROGRAM)
22g	Power Fail Restart (PS/PROC)
23g	Illegal Instruction (PROGRAM/PROC)
24g	Interrupt Watchdog (GBC/PROC)
25g	Memory Error (on CPU Transaction) (MBC)
26g	GENIE Bus Arbitration Error (GBC)
30g	Memory Error (on Interrupt Service) (MBC)
31g	SPM Memory Protect Violation (MPU)

The information saved in locations 21g and 27g is:

<u>Trap</u>	<u>Location 21g</u>	<u>Location 27g</u>
20	Address of last valid instruction, also status information	Not meaningful
22	Undefined	Undefined
23	Illegal Instruction	P Register
24	Not meaningful	P Register
25	Status	P Register
26	Not meaningful	P Register
30	Status	P Register
31	Address of SPB which caused trap	Address generated by SPB

Most traps are discussed in the section where they apply; Illegal instructions traps are defined here. An illegal instruction is any of the following when Memory Protect is shut off:

Any GEN 1 instruction with bits 11, 12, and 13 set.

Any GEN 1 with bit 8 reset and bits 5 or 6 set and bits 9 or 10 set.

Any GEN 2 with bits 14-0 in the range 00000 through 03777 except IAI2 and LPM.

A Byte instruction referencing a Byte point with bits 22 and 23 set.

Any instruction with op-code 74 and bits 12 and 13 set.

When Memory Protect is enabled, the following instructions will cause an illegal instruction trap:

Any instruction which is illegal with Memory Protect off.

Any external GEN 2.

JND or LPM.

Any Control Bus instruction (op-code 27g)

The illegal instruction is stored in memory location 21g and the Trap instruction in location 23g is executed.

## 12. PARTS

The Processor boards are considered non-repairable in the field and are, therefore, considered replaceable parts:

- PX4000RPCA1
  - PX4000PLUA4
  - 51101580-100 - Front-edge PWA connectors
- } Processor PWA's

## APPENDIX A

The following pages contain a listing of the Bootstrap Routine. The routine is loaded by firmware control into the upper memory area following selection of a power-up without battery backup, or the selection of a bootstrap load or memory cost function (Reset with 1-7 from the Programming and Maintenance Console).

### NOTE

The locations shown for the instructions are relative. The last location of the routine will appear in the last memory location. The first instruction (Entry Point) will appear 152<sub>g</sub> locations from the end of memory.

Bootstrap - Executive

```

01 00010
02 00020
03 00030
04 00040
05 00050
06 00060
07 00070
08 00080
09 00090
10 00100
11 00110
12 00120
13 00130
14 00140
15 00150
16 00160
17 00170
18 00180
19 00190
20 00200
21 00210
22 00220
23 00230
24 00240
25 00250
26 00260
27 00270
28 00280
29 00290
30 00300
31 00310
32 00320
33 00330
34 00340
35 00350

* * * * * TTL * * * * * Bootstrap - Executive * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * BOOTSTRAP ROUTINE * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * THIS CODE IS LOADED INTO THE END OF MAIN MEMORY BY THE MICRO-LEVEL
* * * * * BOOTSTRAP ROUTINE. UPON ENTRY THE A-REGISTER CONTAINS A CODE
* * * * * INDICATING THE CAUSE OF THE RE-START CONDITION AS FOLLOWS:
* * * * *
* * * * * -1 - POWERFAIL RECOVERY WITH MAIN MEMORY INVALID
* * * * * 0 - SYSTEM RESET (L CONSOLE DISABLED)
* * * * * 1 - CARD READER
* * * * * 2 - PAPER TAPE READER
* * * * * 3 - BULK (DRUM/DISK)
* * * * * 4 - CONSOLE BOOTSTRAP DEVICE #4
* * * * * 5 - CONSOLE BOOTSTRAP DEVICE #5
* * * * * 6 - CONSOLE BOOTSTRAP DEVICE #6
* * * * * 7 - CONSOLE BOOTSTRAP DEVICE #7
* * * * *
* * * * * OTHER ENTRY CONDITIONS ARE AS FOLLOWS:
* * * * *
* * * * * TSTF, OVFL, PAI, FPM - RESET
* * * * * 0-10 - MAIN MEMORY - UNDEFINED (OR PRESERVED IF NO POWERFAIL
* * * * * J - UNDEFINED
* * * * *
* * * * * 27 DEF /777,15 Pseudo-op definition
* * * * * FOR DEF 0,27,3,0 Pseudo-op definition
* * * * * Q EQL /10 Memory location of Q
* * * * *
* * * * * $BR0MZ-EQL /153 Size of bootstrap
* * * * * LDC /4,000-$BR0MZ-2 Start of bootstrap coding
* * * * * CS BSS 1 Console Switches = Relocation Value
* * * * * CK BSS 1 Checksum
* * * * * SLM
* * * * *
00000010
00000153
240003623 240003623
300000001 300000001
300000001 300000001
003623
003624

```

Bootstrap - Executive

01	003625	32000006	32000006	32000006	
02					
03					
04					
05					
06					
07	003626	41003640	41003640	41040012	
08	003627	63000016	63000016	63000016	
09	003630	41003642	41040012	41040012	
10	003631	32000045	32000045	32000045	
11	003632	44000022	44000022	44000022	
12					
13	003633	00603770	00640135	00640135	
14	003634	32000003	32000003	32000003	
15	003635	05000057	05000057	05000057	
16	003636	32000007	32000007	32000007	
17	003637	14703735	14703735	14740076	
18					
19					
20	003640	42000045	42000045	42000045	
21	003641	27032400	27032400	27032400	
22	003642	00074000	00074000	00074000	
23	003643	14000016	14000016	14000016	
24					

\* Bootstrap entry point  
 \* STA 6  
 \* Command digit from console  
 \* We now must disable power fail recovery routine (until software has completed load or whatever.) Do this by putting linkage to reset in location /16 and putting BRU /16 in 22.  
 \* DLD RSET  
 \* DST /16  
 \* DLD RSET+2  
 \* STA /45  
 \* STQ /22  
 \* Complete bootstrap  
 \* LDA TABLE,6  
 \* STA 3  
 \* SRL 15  
 \* STA 7  
 \* BRU 60,7  
 \* Constants for System Clear  
 \* RSET LDQ /45 /16 - Perform a  
 \* CON 0,27032400 /17 - a System Reset  
 \* CON 0,00074000 /45 - Pattern to cause reset  
 \* BRU /16 /22 - Powerfail entry  
 \* SLM



Bootstrap - Card/Tape Loader

```

01 003676 16200007 X READ LDX 7,2          Cards = 0 / Tape = 6
02 0003677 00003677 EQL 0
03 25360020 25360020 JCB 0,3          At end of media?
04 14040032 14040032 BRU FO          Yes. Check for end-of-file
05 25360040 25360040 JDR 0,3          No. Data ready?
06 14003677 14003677 BRU R1          No. Wait for it
07 25350000 25350000 IM 0,3           Yes. Input next frame
08 24536030 24536030 TXH 1000,5       Did we detect zero-length record?
09 34003677 34003677 BTS R1          Yes. Skip rest of card
10 45702014 45702014 SLL 12,7        No. Process this frame
11 45705314 45705314 DRC 12,7        Cards = 12 Paper tape = 18
12 32000010 32000010 STA Q          Bring in other frames
13 24237767 24237767 X TXH 9,2         and accumulate data word in Q
14 26237775 26237775 X INX -3,2        Done 2 or 4 frames?
15 30003677 30003677 BIR R1          cards = -3/tape = 3,0,-3
16 26500002 26500002 INX 2,5         No. Read another
17 10003624 10003624 ERA CK          Yes. Count 2 "columns"
18 06000006 06000006 DMT 6           Compare with checksum
19 30003658 30003658 BTR E1          Processed entire field?
20 05045627 05045627 TER 23          Yes. Go test checksum = field checksum
21 10003624 10003624 ERA CK          No. Checksum sign = data?
22 34003724 34003724 BRS 0,2         Restore data + - sign
23 11003624 11003624 ADD CK          Yes. No overflow
24 05000067 05000067 SRL 23          No. Add Checksum
25 11000010 11000010 ADD Q          Isolate sign
26 11003624 11003624 ADD CK          and add to data
27 32003624 32003624 STA CK          and checksum
28 00000010 00000010 LDA Q          update checksum
29 35000001 35000001 LPR 1          Exit with data
30 41

```

Card Loader

01	00003732	00003735	00000000	00003735	00000006	01470
02	26536026	00000000	25330000	14003735	14003735	01480
03	24537777	25330000	25050000			01500
04	34003644	32003623	32077664			01510
05	003732	25040000	25040000			01520
06	003733	14003644	14077703			01530
07	003734	14003736	14077774			01540
08						01550
09						01560
10						01570
11	003735		25330000			01580
12	003736		25050000			01590
13	003737		32077664			01600
14	003740		25040000			01610
15	003741		14077703			01620
16	003742		14003736			01630
17						01640
18						01650
19						01660
20						01670
21	003743					01680
22						

```

* TTL Card Loader
* End of media- Did we process end-file record?
FO EQL *
  INX -1002,5      find zero-length record
  TXH 1,5          at beginning of media?
  BTS CO          No. Read another card
                  Yes. Read end-file card. Stop
* Card Reader Initial entry
GD EQL *
CARD -EQL 0
  ABT 0,3          Must equal "GD"
  RCS             Clear device
  STA CS          Get relocation value (+ load/go in 23)
                  and save
  JND             Operator ready?
  BRU CO          Yes. Read 1st media
  BRU GD*1        No. Wait
* Paper tape reader Initial entry
TAPE -EQL 6
  BRU GD          must equal "GD" *6
  SLM            Join card reader path
  
```



Bootstrap - Bulk Memory Coater

01	00000023	25050000	IND	RCS	Put store address *	01910
02	25050000	07640000	LXK IND-1,6	LXK IND-1,6	Store indirect	01920
03	07603761	06640000	STX IND,6	STX IND,6	Stored in 0000?	01930
04	06603762	53077777	STI IND	STI IND	No. Store another word	01940
05	53003762	06077776	DAT IND	DAT IND	Yes. Fall into halt logic	01950
06	06003762	34077776	BTS IND,1	BTS IND,1		01960
07	34003763					01970
08						01980
09						01990
10						02000
11						02010
12						02020
13						02030
14	00000031	14040000	BADD	Invalid entry code		02040
15	14003766		-EOL ←-GO	BRU *	Stop	02050
16			SLW			02060

Bootstrap - Bulk Memory Coater

```

01
02
03
04 003767 00704004 00704004
05 003770 01004004 01004004
06 003771 00004002 00004002
07 003772 00604003 00604003
08 003773 01004004 01004004
09 003774 01004005 01004005
10 003775 01004006 01004006
11 003776 03100000 03100000
12 003777 02300000 02300000
13
14
15
16
17 004000 *00000000 *00000000
THERE WERE 063 PAL STATEMENTS IN ERROR IN THE ABOVE ASSEMBLY
003 X INDEX SPECIFICATION ERR.
060 I ILLEGAL OPERAND

```

```

* Command table - Goes at end of memory
*
*
TABLE
FOR LINK#0
FOR LINK#0
FOR CARD,74002
FOR TAPE,74003
FOR BULK,74004
FOR BULK,74005
FOR BULK,74006
FOR BADD,70000
FOR COAT,70000
-1 - POWERFAIL ENTRY
0 - RESTART (NORMAL REBOOT) ENTRY
1 - CARD READER
2 - PAPER TAPE READER
3 - PRIMARY STORAGE
4 - SECONDARY STORAGE
5 - TERTIARY STORAGE
6 - RESERVED
7 - COAT MEMORY
* (This address must end in /77 or you did something wrong!)
*
*

```

```

* TTL Symbol Analysis
* END

```

```

02070
02080
02090
02100
02110
02120
02130
02140
02150
02160
02170
02180
02190
02200
02210
02220

```

## READER COMMENTS

In order to improve future editions of this and other publications, Honeywell's Process Control Division/Phoenix solicits your comments. You may direct them to the writer through this form, by letter, or telephone. His address and phone number are provided below. Some of the factors contributing to the usefulness of this publication are listed below. Please explain any "no" responses in the COMMENTS section.

Writer: R. Hoover /gc (602) 943-2341 Ext. 516  
 Honeywell Inc.  
 Publications - 130 HVN 364-1 516  
 2222 W. Peoria Ave.  
 Phoenix, Arizona 85029

● Publication No. ACPU1PROC-M

● How is this publication used:

Familiarization	<input type="checkbox"/>	Reference	<input type="checkbox"/>
Training	<input type="checkbox"/>	Maintenance	<input type="checkbox"/>
Other (Explain) _____			

- |   | YES                      | NO                       |
|---|--------------------------|--------------------------|
| ● Does this publication meet your requirements  | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Is the material:  |                          |                          |
| 1) Presented in clear text  | <input type="checkbox"/> | <input type="checkbox"/> |
| 2) Conveniently organized   | <input type="checkbox"/> | <input type="checkbox"/> |
| 3) Adequately detailed  | <input type="checkbox"/> | <input type="checkbox"/> |
| 4) Adequately illustrated   | <input type="checkbox"/> | <input type="checkbox"/> |
| 5) Presented at appropriate technical level   | <input type="checkbox"/> | <input type="checkbox"/> |
| 6) Technically accurate   | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Please provide specific text references (page number, line, etc.) with your comments. |                          |                          |

NAME \_\_\_\_\_ DATE \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY NAME \_\_\_\_\_

AND ADDRESS \_\_\_\_\_

### COMMENTS:

Staple

Communications concerning technical publications should be directed to:

Technical Publications - 130  
Process Control Division/Phoenix  
Honeywell Inc.  
2222 West Peoria Avenue  
Phoenix, Arizona 85029

Fold

Fold

**FIRST CLASS**  
Permit No. 4091  
Phoenix, Arizona

**BUSINESS REPLY MAIL**  
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

**PROCESS CONTROL DIVISION/PHOENIX**  
**HONEYWELL INC.**  
**2222 West Peoria Avenue**  
**Phoenix, Arizona 85029**



Cut Along Line

Attention: Manager, Marketing Communications - 140

Fold

Fold