

SERIES-III PL/M-86 X19B COMPILATION OF MODULE CXUBOOT
 OBJECT MODULE PLACED IN :F1:CX300T.OBJ
 COMPILER INVOKED BY: PLM86.86 :F1:CX300T.P86 XREF PRINT(:F5:CX300T.LST)

```

/*****
*
*   TITLE: CXU$BOOT
*
*   ABSTRACT: MODULE CXU$BOOT CONTAINS THE ROUTINES WHICH MAKE UP THE CXU
*             PROM BOOTSTRAP. THE FUNCTIONS PERFORMED ARE THOSE OF THE TPS SYSBOOT
*             SLAVE PROCESSOR
*
*****/

1      CXU$BOOT: DO;

/* .BOOTSTRAP COMMAND LITERALS */
2      1  DECLARE IDLE      LITERALLY '0',
          SYNC$CMD        LITERALLY 'OFFH',
          NOT$BUSY        LITERALLY '0',
          MAX$CMD          LITERALLY '3',

          CXU$INITIALIZED LITERALLY '1234H',
          CXU$READY        LITERALLY '0ABH',
          CXU$BOOTED       LITERALLY '0ABCDH',
          CXU$BOOT$ERROR   LITERALLY '0EEH';

3      1  DECLARE COM$BLOCK STRUCTURE(
          CMD          BYTE,
          RESP         BYTE,
          PTR          POINTER,
          MISC         WORD,
          BOOT$FLAG    WORD,
          BOOT$ERR     WORD) AT (04E3B0H);

4      1  CXU_POINTER_FIXUP: PROCEDURE(IN_PTR) POINTER EXTERNAL;
5      2  DECLARE IN_PTR POINTER;
6      2  END CXU_POINTER_FIXUP;

7      1  LONG$CALL: PROCEDURE (PTR$TO$CALL) EXTERNAL;
8      2  DECLARE PTR$TO$CALL POINTER;
9      2  END LONG$CALL;

10     1  CXU$SUICIDE: PROCEDURE (ERROR) PUBLIC;
11     2  DECLARE ERROR WORD;
12     2  DECLARE AND_DIE LITERALLY 'GOTO EAT_LEADEN_DEATH';
/* FOR NOW CONSISTENCY CHECK IS A SUICIDE ROUTINE */
/* SOME ERRORS TRAPPED MAY BE RECOVERABLE */

13     2  /* RETURN ERROR CODE TO MASTER BOOT */
          COM$BLOCK.BOOT$ERR= ERROR;

```

```

14 2      COMSBLOCK.RESP= CXU$BOOT$ERROR;
15 2      /* TURN ON LED ERROR LIGHT */
        OUTPUT(0262H)= 70;
16 2      DISABLE;
17 2      EAT_LEADEN_DEATH:
        AND_DIE;
18 2      END CXU$SUICIDE;

        /*****
        /*
        /*          CXU BOOT ENTRY          */
        /*
        /*****

19 1      CXU_BOOT_MAIN:  PROCEDURE PUBLIC;

20 2      DECLARE (BUFFER$PTR,DEST$PTR,TEMP$PTR) POINTER;
21 2      DECLARE (BOOT$CMD, I) BYTE;
22 2      DECLARE COPY$LEN WORD;

23 2      DISABLE;

24 2      OUTPUT(262h)= 60h; /* turn on both LEDs */

        /* WAIT WHILE SPU MAPS CSU MEMORY (ABOUT 2 SEC.) */
25 2      DO I=1 TO 500;
26 3          CALL TIME(250);
27 3      END;

28 2      OUTPUT(262h)= 00h; /* turn off both LEDs */

29 2      COMSBLOCK.CMD = IDLE;
30 2      COM$BLOCK.RESP = NOT$BUSY;

        /* WAIT FOR SYNCHRONIZE COMMAND */
31 2      DO WHILE COM$BLOCK.CMD <> SYNC$CMD;
32 3          CALL TIME(100); /* WAIT 10 MSEC */
33 3      END;
34 2      COMSBLOCK.CMD = IDLE;
35 2      COMSBLOCK.RESP = NOT$BUSY;

        /* CONTROL TRANSFERS OUT THIS LOOP VIA A TRANSFER CONTROL COMMAND */
        /* FROM MASTER SPU. ERROR EXIT IS ROUTINE CXU$SUICIDE. ERROR */
        /* RECOVERY IS VIA SYSTEM RESET. */

36 2      DO WHILE 1;
        /* CALL ROUTINE BASED ON COMMAND RECEIVED */
37 3      IF (COMSBLOCK.CMD >= 0) AND (COMSBLOCK.CMD <= MAX$CMD)
38 3      THEN DO;

39 4          BOOT$CMD= COMSBLOCK.CMD;
40 4          DO CASE COMSBLOCK.CMD;

```

```

/*          IDLE COMMAND          */
41  5          DOSNOTHING: CALL TIME(50);
/*          SAVE POINTER IN COM. BLOCK AS POINTER TO COPY BUFFER          */
42  5          LOAD$PTR:  BUFFER$PTR= COM$BLOCK.PTR;
/*          MOVE CONTENTS OF BUFFER AT BUFFER$PTR TO LOCATION POINTED          */
/*          TO BY COM$BLOCK.PTR. COM$BLOCK.MISC IS COPY LENGTH          */
43  5          COPY:  DO;
44  6              DEST$PTR= COM$BLOCK.PTR;
45  6              COPY$LEN= COM$BLOCK.MISC;
/*          CXU_POINTER_FIXUP CHANGES THE MULTIBUS WINDOW          */
/*          TEMP$PTR= CXU_POINTER_FIXUP(BUFFER$PTR);          */
46  6              CALL MOV$(TEMP$PTR, DEST$PTR, COPY$LEN);
47  6          /*          RESTORE MULTIBUS WINDOW TO LOOK AT COMMAND BLOCK          */
/*          OUTPUT(2COH)= 0EH;          */
48  6              END;
49  6          /*          TRANSFER CONTROL TO ADDRESS IN COM$BLOCK.PTR          */
/*          IN GENERAL CONTROL WILL NOT RETURN TO THIS MODULE          */
50  5          TRANSFER:
51  6              DO;
52  6                  COM$BLOCK.BOOT$FLAG= CXUSBOOTED;
53  6                  CALL LONG$CALL(COM$BLOCK.PTR);
54  5              END; /* DO CASE */
55  4          END; /* IF THEN */
56  3          IF BOOT$CMD <> IDLE
57  3              THEN DO;
58  4                  COM$BLOCK.CMD = IDLE;
59  4                  COM$BLOCK.RESP = NOT$BUSY;
60  4              END;
61  3          END;
62  2          END CXU_BOOT_MAIN;
63  1          END CXUSBOOT;
```

CROSS-REFERENCE LISTING

DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
12			AND_DIE LITERALLY 'GOTO EAT_LEADEN_DEATH' IN PROC (CXUSUICIDE) 17
21	0008H	1	BOOTCMD BYTE IN PROC (CXU_BOOT_MAIN) 39* 56
20	0000H	2	BUFFERPTR POINTER IN PROC (CXU_BOOT_MAIN) 42* 46
3	4HE3B0H	10	COMBLOCK STRUCTURE AT ABSOLUTE
	0000H	1	CMD BYTE 29* 31 34* 37 39 40 58*
	0001H	1	RESP BYTE 14* 30* 35* 59*
	0002H	2	PTR POINTER 42 44 52
	0004H	2	MISC WORD 45
	0006H	2	BOOTFLAG WORD 51*
	0008H	2	BOOTERR WORD 13*
43	00E1H		COPY LABEL IN PROC (CXU_BOOT_MAIN)
22	0006H	2	COPYLEN WORD IN PROC (CXU_BOOT_MAIN) 45* 47
	0000H		CXUBOOT PROCEDURE STACK=0000H
2			CXUBOOTED LITERALLY '0ABCDH' 51
2			CXUBOOTERROR LITERALLY '0EEH' 14
2			CXUINITIALIZED LITERALLY '1234H'
2			CXUREADY LITERALLY '0ABH'
10	0000H	37	CXUSUICIDE PROCEDURE PUBLIC STACK=0004H
19	0025H	311	CXU_BOOT_MAIN PROCEDURE PUBLIC STACK=0006H
4	0000H		CXU_POINTER_FIXUP PROCEDURE POINTER EXTERNAL(0) STACK=0000H 46
20	0002H	2	DESTPTR POINTER IN PROC (CXU_BOOT_MAIN) 44* 47
41	00C1H		DONOTHING LABEL IN PROC (CXU_BOOT_MAIN)
17	001EH		EAT_LEADEN_DEATH LABEL IN PROC (CXUSUICIDE) 17
11	0004H	2	ERROR WORD IN PROC (CXUSUICIDE) PARAMETER AUTOMATIC 11 13
21	0009H	1	I BYTE IN PROC (CXU_BOOT_MAIN) 25* 25 27
2			IDLE LITERALLY '0' 29 34 56 58
5	0000H	2	IN_PTR POINTER IN PROC (CXU_POINTER_FIXUP) PARAMETER 5
42	00D2H		LOADPTR LABEL IN PROC (CXU_BOOT_MAIN)
7	0000H		LONGCALL PROCEDURE EXTERNAL(1) STACK=0000H 52
2			MAXCMD LITERALLY '3' 37
			MOV8 BUILTIN 47
2			NOTBUSY LITERALLY '0' 30 35 59
			OUTPUT BUILTIN 15* 24* 28* 48*
8	0000H	2	PTRTOCALL POINTER IN PROC (LONGCALL) PARAMETER 8
2			SYNCCMD LITERALLY 'OFFH' 31
20	0004H	2	TEMPPTR POINTER IN PROC (CXU_BOOT_MAIN) 46* 47
			TIME BUILTIN 26 32 41
50	011CH		TRANSFER LABEL IN PROC (CXU_BOOT_MAIN)

MODULE INFORMATION:

CODE AREA SIZE = 015CH 348D
 CONSTANT AREA SIZE = 0004H 4D
 VARIABLE AREA SIZE = 000AH 10D
 MAXIMUM STACK SIZE = 0006H 6D
 157 LINES READ
 0 PROGRAM WARNINGS
 0 PROGRAM ERRORS

END OF PL/M-86 COMPILATION