```
ASSIGN M:CI,(FILE,INITIAL,:DooCI)
METASYM CI,L0,CN
    .SS R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15
    .SS SR1,SR2,SR3,SR4,D1,D2,D3,D4,$
    .END
```

```
CONV
        440/BAL         447/BAL         474=EQU
CONV1
        476/BNEZ        479=EQU
CONV2
        481=EQU         490/BDR
CORED
         91/REF         240/LW
CRNDD
        416/STO*        418/TIO*        508=DATA
CSED$PATH
        107/REF         187/STW
DA
        152/LI          152/LI          360/LI          402/LI          409/LI          415/LI          421/LI
        434/LI
DCT1
        105/REF         358/LH
DEBUG
         19=SET         142/DO          372/DO
DECFPSD
         47/REF         329/XPSD
EXIT
        408/BNE         429/BEZ         539=EQU
FIVES
        249/LW          311=DATA
FIXOVPSD
         45/REF         320/XPSD
FLTFPSD
         46/REF         321/XPSD
HGP
         75/REF         161/LH
HLTPSD
        346/XPSD        351=IPSD
IA
        351/IPSD        352/IPSD        353/IPSD
INITIAL
        113/DEF         123=LPSD        347/DATA        563/END
```

```
                *M*   INITIAL  MONITOR INITIALIZATION ROUTINE - RESIDENT
                *P*
                *P*   NAME:     INITIAL
                *P*
                *P*   PURPOSE:  INITIALIZATION ROUTINE FOR RESIDENT MONITOR
                *P*
                *P*   DESCRIPTION:  (1) SETS UP TRAP AND INTERRUPT LOCATIONS
                *P*                  (2) BALS TO BOOTSUBR FOR FURTHER INITIALIZATION
                *P*                        AND/OR TAPE BOOT PROCESS
                *P*                  (3) SETS UP THE HARDWARE WRITE LOCKS AND MAP
                *P*                  (4) BUILDS MONITOR JIT
                *P*                  (5) INITIATES ALLOCAT AND GHOST1
                *P*                  (6) STARTS UP SYSTEM BY EXITING TO SCHEDULER
                *P*
                      PCC       0
```

```
17                          DEF     INITIAL:        MODULE NAME
18          01 00000    INITIAL: EQU     $
19          00000000    DEBUG   SET     0           SET TO 1 TO INCLUDE ABS PATCHER
20          00000002    BC      EQU     2
21          00000001    BITS    SET     1           GET BIT31TO0 AND MASKS
22                          SYSTEM  UTS
23          00000000    R0      EQU     0
24          00000001    R1      EQU     1
25          00000002    R2      EQU     2
26          00000003    R3      EQU     3
27          00000004    R4      EQU     4
28          00000005    R5      EQU     5
29          00000006    R6      EQU     6
30          00000007    R7      EQU     7
31          00000008    R8      EQU     8
32          00000009    R9      EQU     9
33          0000000A    R10     EQU     10
34          0000000B    R11     EQU     11
35          0000000C    R12     EQU     12
36          0000000D    R13     EQU     13
37          0000000E    R14     EQU     14
38          0000000F    R15     EQU     15
```

```
40                      *  REFS AND DEFS

42                         REF     NOPPSD          TRAP 40
43                         REF     UNIMPPSD        TRAP 41
44                         REF     STKLPSD         TRAP 42
45                         REF     FIXOVPSD        TRAP 43
46                         REF     FLTFPSD         TRAP 44
47                         REF     DECFPSD         TRAP 45
48                         REF     PSD$T46         TRAP 46 WATCHDOG TIMER
49                         REF     IPT47           TRAP 47 INTER-PROCESSOR TRAP
50                         REF     CAL1PSD         TRAP 48
51                         REF     CAL2PSD         TRAP 49
52                         REF     CAL3PSD         TRAP 4A
53                         REF     CAL4PSD         TRAP 4B
54                         REF     PSD$T4C         TRAP 4C HARDWARE ERROR TRAP
55                         REF     PSD$T4D         TRAP 4D REG EXCEPTION TRAP
56                         REF     LEE20           TRAP 4E XDLT ENTRY
57                         REF     POWRON          TRAP 50
58                         REF     POWROFF         TRAP 51
59                         REF     TINC            INTR 54 CLOCK 3 PULSE
60                         REF     S:CLOCK4        INTR 55 CLOCK 4 PULSE (          SPACE
61                         REF     PSD$I56         INTR 56 MEM PE INTR OR PFI
62                         REF     PSD$I57         INTR 57 MEM FLT INTR
63                         REF     CLK3PSD         INTR 5A CLOCK3 ZERO
64                         REF     CLK4PSD         INTR 5B CLOCK4 ZERO
65                         REF     IOPSD           INTR 5C
66                         REF     OCPSD           INTR 5D

68                         REF     MONORG          START OF RESIDENT CODE

70                         REF     BOOTHLT         EMERGENCY HALT (BOOT OC INTR)

72                         REF     MONINIT         ENTRY POINT TO BOOTSUBR

74                         REF     PSA$END         SET UP FOR RECOVERY
75                         REF     HGP             SET UP FOR RECOVERY
76                         REF     BOOTSBAND       SET UP FOR RECOVERY
```

```
 78                    REF     MB:GPT          USED TO REL GHOST1 SW SPACE
 79                    REF     SL:CORE         USED TO REL GHOST1 SW SPACE
 80                    REF     T:SGRNU         USED TO REL GHOST1 SW SPACE
 81                    REF     MB:GAM4         USED TO REL GHOST1 SW SPACE

 83                    REF     S:GJOBTBL       USED TO START ALLOCAT & GHOST1
 84                    REF     MING            USED TO START ALLOCAT & GHOST1
 85                    REF     T:GJOBSTRT      USED TO START ALLOCAT & GHOST1

 87                    REF     SPOOL           INITIALIZED TO ZERO

 89                    REF     T:SE            ENTRY TO SCHEDULER

 91                    REF     CORED           USED TO SET UP WRITE LOCKS
 92                    REF     PPSTART         USED TO SET UP WRITE LOCKS
 93                    REF     JIT             USED TO SET UP WRITE LOCKS
 94                    REF     LKIMG           USED TO SET UP WRITE LOCKS
 95                    REF     PB:HVA          USED TO SET UP WRITE LOCKS
 96                    REF     IOLOW           USED TO SET UP WRITE LOCKS
 97                    REF     IOHIGH          USED TO SET UP WRITE LOCKS
 98                    REF     TSTACK          USED TO BUILD MONITOR JIT

100                    REF     :BIG            USED TO SET UP MAP

102                    REF     SYSVERS         SET UP FROM X'2B'

104                    REF     OB:BOPTX        DCTX OF OC FOR ERROR MSG
105                    REF     DCT1            IO ADDR OF OC FOR ERROR MSG

107                    REF     CSED$PATH       CLEARED TO ZERO FOR FLT HANDLERS

109                    REF     LSWAP           NO. OF SWAPPERS IN SYSTEM
110                    REF     NSWAP           CELL CONTAINING NO. OF SWAPPERS IN S
111                *,*                         INITIALLY SET AT ONE.

113                    DEF     INITIAL         ENTRY POINT AND MODULE BASE
```

```
114
115
116
                DEF     LMA             ENTRY TO RELOAD WRT LOCKS,MAP & ACC
                DEF     UMOV#           INSTR FILLED IN BY BOOTSUBR
                DEF     TOPROOT         TOP OF RESIDENT CODE
```

```
                        *F*  NAME:  INITIAL
                        *F*
                        *F*  PURPOSE:  INITIAL ENTRY POINT FOR STARTUP OF SYSTEM
                        *F*            DURING ANY TYPE OF BOOT OR RECOVERY

01 00000  05000000      INITIAL LPSD,0  INITPSD
01 00001                INIT2   RES
01 00001  32200004      A       LW,2    S7TST
01 00002  35200040      A       STW,2   X'40'
                        *
                        *O*     MESSAGE: SYSTEM REQUIRES SIG9 OR X560
                        *O*
                        *O*     MEANING: THIS PO TAPE CANNOT BE BOOTED ON A SIGMA 6/7
                        *O*              BECAUSE IT NEEDS HALFWORD MAP FORMAT
                        *
01 00003  67000005      A       EXU     S7INST          CHECK IF SIG7 IN BIG MAP MODE
01 00004  222FFFA0      N       LI,2    MONORG=X'60'
01 00005  32100082      A       LW,1    MTWO            'MTW,0  0' INSTRUCTION
01 00006  35140051      A       STW,1   X'5F',2
01 00007  64200006      A       BDR,2   *-1
01 00008  22100020      A       LI,1    X'5F'-X'3F'
01 00009  32220098      A       LW,2    XPSDS-1,1       SET TRAPS, INTERRUPTS, ETC
01 0000A  35220031      A       STW,2   X'3F',1
01 0000B  64100009      A       BDR,1   *-2
          00000000              DO      DEBUG
                        *S*     XPSD,10 TRP51           DO WE WANT TO PATCH
                                FIN
01 0000C  22102030      A       LI,1    X'2030'         ARM/ENABLE I/O,CLOCK3,OC
01 0000D  6A101200      A       AIO,1   ARM*ENABLE
01 0000E  6AB00000      X       BAL,11  MONINIT
01 0000F  6AB00049      A       BAL,11  LMA
01 00010  22000001      N       LI,0    TSTACK+1        BUILD MONITOR JIT
01 00011  221001FE      N       LI,1    JIT+512-TSTACK-2
01 00012  25100010      A       SLS,1   16
01 00013  22200000      N       LI,2    DA(TSTACK)-DA(JIT-2)
01 00014  1505FFFE      N       STD,0   JIT-2,2
01 00015  18000000      A       SD,0    0
```

```
155  01 00016  642000014          BDR,2   $-2
156  01 00017  329000000 X        LW,9    PSA$END          REL SWAP SPACE FOR GH=1
157  01 00018  219001480 A        CI,9    X'1480'          DON'T ALLOCATE LAST
158  01 00019  691000018          BL      $+2              4 GRAMS
159  01 0001A  209FFFF8 A        AI,9    -8
160  01 0001B  359000000 X        STW,9   BOOTSBAND         INITIALIZE BOOTSBAND
161  01 0001C  527000001 N        LH,7    HGP+1             FROM END OF PSA
162  01 0001D  557000000 X        STH,7   BOOTSBAND         INSERT DCT INDEX
163  01 0001E  722000000 X        LB,2    MB'GPT           GRAN PER TRACK
164  01 0001F  252000001 A        SLS,2   1                SECTORS PER TRACK
165  01 00020  228000000 A        LI,8    0
166  01 00021  368000002 A        DW,8    2
167  01 00022  222000000 A        LI,2    0                SWAPI OF FIRST SWAPPER
168  01 00023  723000000 X        LB,3    MB'GAM4
169  01 00024  259600001 A        SLS,9   1,3
170  01 00025  309000008 A        AW,9    8
171  01 00026  327000000 X        LW,7    SL'CORE
172  01 00027  207000003 A        AI,7    3
173  01 00028  2570047E A        SAS,7   -2
174  01 00029  25700402 A        SAS,7   2
175  01 0002A  209FFFFE A        AI,9    -2
176  01 0002B  32F000009 A        LW,15   9
177  01 0002C  6AB000000 X        BAL,11  T'SGRNU          REL SL'CORE GRAN FOR GH1
178  01 0002D  691000002A         BCS,1   $-3
179  01 0002E  33E000000 X        MTW,-2  BOOTSBAND
180  01 0002F  647000002A         BDR,7   $-5
181       C1 00030          NOSREL  EQU     $
182  01 00030  320000000 X        LW,0    SYSVERS
183  01 00031  350000028 A        STW,0   X'28'
184  01 00032  120000BA           LD,0    IOXPSDS          REPLACE XPSDS
185  01 00033  1500005C A        STD,0   X'5C'            FOR I/O AND OC
186  01 00034  222000000 A        LI,R2   0                LET TRAP HANDLERS
187  01 00035  352000000 X        STW,R2  CSED$PATH        KNOW INITIAL IS DONE
188  01 00036  120000004 N        LD,0    S'GJOBTBL+4      START UP ALLOCAT
189  01 00037  6AA000000 X        BAL,10  T'GJOBSTRT
190  01 00038  120000000 F        LD,0    S'GJOBTBL+MING+MING
191  01 00039  6AA000000 X        BAL,10  T'GJOBSTRT       START UP THE STARTER GHOST
```

```
   01 0003A    22100001 N          LI,R1     LSWAP+1             NO. OF SWAPPERS IN SYSTEM
   01 0003B    35100000 X          STW,R1    NSWAP
   01 0003C    221038F0 A          LI,1      X'38F0'             ARM AND ENABLE: PANEL,IO,
                              *                                     C4Z,C3Z,C4P,C3P,C2P
   01 0003D    70200000 X          BIF,S7    WD                  FOR SIG7, JUST MPI
   01 0003E    69800043
   01 0003F    221034F0 A          LI,R1     X'34F0'             FOR SIG9, JUST MFI
   01 00040    70200000 X          BIF,S9    WD
   01 00041    69400043
   01 00042    22103CF0 A          LI,R1     X'3CF0'             FOR X560, MFI AND PFI
         01 00043             WD   EQU       $
   01 00043    6D101200 A          :WD,1     ARM$ENABLE
   01 00044    22100080 A          LI,1      X'80'               FIRE CLOCK3 ZERO
   01 00045    6D101700 A          :WD,1     TRIGGER
   01 00046    22100000 A          LI,1      0
   01 00047    B5100000 X          STW,1     *SPOOL              END CHAIN
   01 00048    68000000 X          B         T:SE
                              *
                              *F*       NAME:     LMA
                              *F*
                              *F*       PURPOSE: TO RELOAD WRITE LOCKS AND MAP AND ACCESS CODES
                              *F*
                              *F*       DESCRIPTION: CALLED DURING INITIALIZATION AND
                              *F*                    CALLED AFTER POWER FAIL/RESTORE OR AFTER
                              *F*                    PARITY ERROR TO RELOAD THE HARDWARE REGISTERS
                              *F*
                              *D*       NAME:     LMA
                              *D*
                              *D*       CALL:     BAL,11 LMA
                              *D*
                              *D*       REGISTERS: ALL REGISTERS VOLATILE
                              *
         01 00049             LMA       EQU       $
   01 00049    6AA00058            BAL,R10   LOADWL              GO LOAD WRITE LOCKS
   01 0004A    6AA0004C            BAL,R10   LOADMAP             LOAD MAP AND ACCESS CODES
   01 0004B    E800000B A          B         *R11
         01 0004C             LOADMAP   EQU       $
```

```
227  01 0004C  223000040 A          LI,3    64
228  01 0004D  253000000 N          SLS,R3  ;BIG              SHIFT FOR BIG MAP MODE
229  01 0004E  322000096            LW,2    MAPINIT
230  01 0004F  6A4000E8E            BAL,4   LOOP
231  01 00050  6F080000 N           LDMAP,0 0
232  01 00051  302000097            AW,2    MAPINC
233  01 00052  322000098 LACBF     LW,2    CJOBACC
234  01 00053  223000010 A          LI,3    16
235  01 00054  6A4000E8E            BAL,4   LOOP
236  01 00055  6F040000 A           LPC,0
237  01 00056  222FFFFF A           LI,R2   -1
238  01 00057  E8000000A A          B       *R10
239  01 00058                 LOADWL  RES
240  01 00058  323000000 X          LW,3    CORED             FIRST SET ALL TO ZERO
241  01 00059  253000073 A          SLS,3   -9-4
242  01 0005A  6A4000E8E            BAL,4   LOOP
243  01 0005B  222000000 A          LI,2    0
244  01 0005C  6F020000 A           LLOCKS,0
245  01 0005D  321000C7A            LW,1    WLZAP             ZAP WRITE LOCK BUFFER
246  01 0005E  610000258            MBS,0   BA(MAPINIT)
247  01 0005F  221000000 A          LI,1    0                 NOW SET MON DATA TO 01
248  01 00060  224FFFFF N           LI,4    PPSTART-1
249  01 00061  322000095            LW,2    FIVES
250  01 00062  6A600007B            BAL,6   LWL
251  01 00063  221000000 N          LI,1    JIT               SET JIT TO 01
252  01 00064  224001FF N           LI,4    JIT+511
253  01 00065  6A600007B            BAL,6   LWL
254  01 00066  221001FF N          LI,1    PPSTART+511       SET MON PURE PROCEDURE TO 11
255  01 00067  224FFF98 N           LI,4    TOPROOT-1-256     IF SPOOL BUFFER, DONT PROTECT IT
256  01 00068  222FFFFF A           LI,2    -1
257  01 00069  6A600007B            BAL,6   LWL
258  01 0006A  221000200 N          LI,1    JIT+512           SET UMOV TO 11
259  01 0006B  225000000 A     UMOV#  LI,5    0                 FILLED IN BY BOOTSUBR
260  01 0006C  724A00000 X          LB,4    PB!HVA,5
261  01 0006D  254000009 A          SLS,4   9
262  01 0006E  204FFFFF A           AI,4    -1
263  01 0006F  6A600007B            BAL,6   LWL
```

```
264  01 00070  70200000 X          BIF,S7S9 N0I0WL
     01 00071  69C00076
265  01 00072  22100000 N          LI,1     I0L0W          SET I0 BUFFERS T0 0 F0R 560
266  01 00073  224FFFFF N          LI,4     I0HIGH=1
267  01 00074  22200000 A          LI,2     0
268  01 00075  6A60007B            BAL,6    LWL
269  01 00076                 N0I0WL RES
270  01 00076  22000000 N          LI,0     LKIMG
271  01 00077  3210001C N          LW,1     Y08
272  01 00078  6B020000 A          LL0CKS,0                INT0 THE HARDWARE REGISTERS
273  01 00079  E8000000A A          B        *10            AND RETURN
274                            *
275                            *
276  01 0007A  20000000 N     WLZAP GEN,8,24 3P,BA(LKIMG)
277                            *
278                            *
279  01 0007B                 LWL   RES
280                            *      SET WRITE L0CK IMAGE  F0R PAGES (1) THRU (4)
281                            *      T0 (2) (32 BITS)
282  01 0007B  25100077 A          SLS,1    =9
283  01 0007C  2230000F A          LI,3     X'F'           GET DISP INT0 W0RD
284  01 0007D  4B300001 A          AND,3    1
285  01 0007E  233FFFFE A          MI,3     =2             (C0MPLEMENTED)
286  01 0007F  32360020 N          LW,3     MASKS+32,3     INITIAL MASK T0 USE
287  01 00080  2510007C A          SLS,1    =4             INITIAL W0RD DISP
288  01 00081  25400573 A          SAD,4    =9=4           N0W GET MASK F0R LAST ST0RE
289  01 00082  25500064 A          SLS,5    =28
290  01 00083  235FFFFE A          MI,5     =2
291  01 00084  3A5A001E N          LCW,5    MASKS+30,5
292  01 00085  205FFFFF A          AI,5     =1
293  01 00086  31400001 A     LWL1 CW,4     1
294  01 00087  691C0000 A          BL       0,6            RETURN IF D0NE
295  01 00088  69300008A           BNE      $+2
296  01 00089  4B300005 A          AND,3    5              SET FINAL MASK (MAY ALS0 BE FIRST 0N
297  01 0008A  47220000 X          STS,2    LKIMG,1
298  01 0008B  20100001 A          AI,1     1
299  01 0008C  223FFFFF A          LI,3     =1             INTERMEDIATE MASK
```

```
300  01 0008D  68000086          B       LWL1
301                              *
302                              *
303                              *
304  01 0008E  22100001 A  LOOP  LI,1     1
305  01 0008F  22000002 A        LI,0     2
306  01 00090  75100001 A        STB,1    1
307  01 00091  67080000 A        EXU      0,4
308  01 00092  67080001 A        EXU      1,4
309  01 00093  64300008F  LOOP5  BDR,3    LOOP*1
310  01 00094  68080002 A        B        2,4
311  01 00095  55555555 A  FIVES  DATA     X'55555555'
312  01 00096  00010203 N  MAPINIT GEN,15,1,6,1,7,1,1   0,1=:BIG,0,1=:BIG,0,1=:BIG,1
313                        MAPINC  GEN,5,1,7,1,1,6,1,7,1,1,1   0,1=:BIG,0,1=:BIG,:BIG,0,1=:BIG,;
314  01 00097  04040404 N                                      0,1=:BIG,:BIG,0
315  01 00098  BFFFFFFF A  CJOBACC DATA     X'BFFFFFFF'
316  01 00099              TOPROOT RES
317  01 00099  0FA00000 X  XPSDS   XPSD,10   NOPPSD
318  01 0009A  0FA00000 X        XPSD,10   UNIMPPSD
319  01 0009B  0FA00000 X        XPSD,10   STKLPSD
320  01 0009C  0FA00000 X        XPSD,10   FIXOVPSD
321  01 0009D  0FA00000 X        XPSD,10   FLTFPSD
322  01 0009E  0FA00000 X        XPSD,10   DECFPSD
323  01 0009F  0F000000 X        XPSD,0    PSD$T46
324  01 000A0  0FA00000 X        XPSD,10   IPT47              INTER-PROCESSOR TRAP
325  01 000A1  0FA00000 X        XPSD,10   CAL1PSD
326  01 000A2  0FA00000 X        XPSD,10   CAL2PSD
327  01 000A3  0FA00000 X        XPSD,10   CAL3PSD
328  01 000A4  0FA00000 X        XPSD,10   CAL4PSD
329  01 000A5  0F000000 X        XPSD,0    PSD$T4C
330  01 000A6  0F000000 X        XPSD,0    PSD$T4D
331  01 000A7  0F800000 X        XPSD,8    LEE20             LEES WATERING HOLE
332  01 000A8  FF000000 N        DATA      JIT+255**24       CJOB
333  01 000A9  0FA00000 X        XPSD,10   POWRON
334  01 000AA  0FA00000 X        XPSD,10   POWROFF
335  01 000AB  33000000 A        MTW,0     0
336  01 000AC  33000000 A        MTW,0     0                 CLOCK2
```

```
337  01 000AD  33F00000 X          MTW,-1    TINC
338  01 000AE  B3100000 X          MTW,1     *S:CLOCK4          ACCOUNTING CLOCK (CLOCK4)
339  01 000AF  0F000000 X          XPSD,0    PSD$I56
340  01 000B0  0F000000 X          XPSD,0    PSD$I57
341  01 000B1  33000000 A          MTW,0     0                  CLOCK1
342  01 000B2  33000000 A  MTWO    MTW,0     0                  CLOCK2
343  01 000B3  0FA00000 X          XPSD,10   CLK3PSD
344  01 000B4  0FA00000 X          XPSD,10   CLK4PSD
345  01 000B5  0FA000C2            XPSD,10   AIOPSD             INITIAL IO INTERRUPT XPSD
346  01 000B6  0FA000BC            XPSD,10   HLTPSD             BOOT RESTART HALT
347  01 000B7  00000000 A          DATA      0,INITIAL          MON START ADR FOR DISCBOOT
     01 000B8  00000000
348                                BOUND     8
349  01 000BA  0F800000 X  IOXPSDS XPSD,8    IOPSD
350  01 000BB  0F800000 X          XPSD,8    OCPSD
351  01 000BC  00000000 N  HLTPSD  :PSD      RES,(IA,BOOTHLT)
                 00000000
                 00000000
                 00000000
352  01 000C0  00000001 N  INITPSD :PSD      (IA,INIT2)
                 00000000
353  01 000C2  00000000 N  AIOPSD  :PSD      RES,(IA,AIOSRV)
                 00000000
                 000000DE
                 00000000
354  01 000C6  05000358 N  S7MSG   GEN,8,24,32  5,BA(S7TXT),31
                 0000001F
355  01 000C8  00000000 A  S7ERR   DATA      0,0,$+2,0
     01 000C9  00000000 A
     01 000CA  000000CC
     01 000CB  00000000 A
356  01 000CC  22100002 A          LI,1      OC                 OC OP LEBEL INDEX
357  01 000CD  72120000 X          LB,1      OB:BOPTX,1         DCTX OF OC
358  01 000CE  52120000 X          LH,1      DCT1,1
359  01 000CF  201FAF00 A          AI,R1     =X'C100'           MAKE INTO SIGMA7 ADDRESS
360  01 000D0  22000063            LI,0      DA(S7MSG)          TELL OPERATOR AND QUIT.
361  01 000D1  CC000001 A          SIO,0     *1
```

```
     01 000D2      680000D2               B          $
     01 000D3      680000D1               B          $-2
     01 000D4      0FA000C8     S7TST     XPSD,10    S7ERR
     01 000D5      28000000 N   S7INST    GEN,5,1,26    5,!BIG,0
     01 000D6      15       A   S7TXT     DATA,1     X'15',X'15','S','Y'
     01 000D6 1    15       A
     01 000D6 2    E2       A
     01 000D6 3    E8       A
     01 000D7      E2E3C5D4 A             TEXT       'STEM REQUIRES SIG9 OR X560'
     01 000D8      40D9C5D8 A
     01 000D9      E4C9D9C5 A
     01 000DA      E240F2C9 A
     01 000DB      C7F940D6 A
     01 000DC      D940F7F5 A
     01 000DD      F6F04040 A
     01 000DE      6E000000 A   AIOSRV    AIO,0      0
     01 000DF      740000E1               STCF       S9AIOCC             SAVE AIO CC
     01 000E0      0EB000C2               LPSD,X'B'  AIOPSD              RETURN
     01 000E1      00000000 A   S9AIOCC   DATA       0
                00000000                  DO         DEBUG
                            *S* ******************
                            *S* *   B O O T   T I M E   P A T C H   R O U T I N E
                            *S* *
                            *S* *         PATCH CP-V MONITOR ROOT AND BOOT TIME ROUTINE
                            *S* *         WITH ABSOLUTE PATCHES. PATCH FORMAT IS:
                            *S* *
                            *S* *         COL.1          COL.9
                            *S* *
                            *S* *         LOC.           VALUE
                            *S* *
                            *S* *                   WHERE: LOC     =5 HEX DIGIT ADDR
                            *S* *                          VALUE   =8 HEX DIGIT VALUE
                            *S* *
                            *S* *   ENTRY  :        BY EXECUTING TRAP LOC. X'51' WHICH CONTAINS XPSD
                            *S* *   EXIT   :         BY LPSD TO RESULT OF ENTRY XPSD
                            *S* *
                            *S* *
```

```
391                    *S* *         PATCHES ARE TERMINATED BY  A  EOD CARD, AND ARE
392                    *S* *         READ FROM THE CARD READER
393                    *S* *
394                    *S* *         PATCHES ARE PRINTED ON THE LINE PRINTER
395                    *S* *
396                    *S* ********************
399                    *S* PACHER    EQU       $                   ENTRY-------
400                    *S*           LCI       0                   SAVE REGS.
401                    *S*           PSM,R0    STK
402                    *S*           LI,0      DA(PTHMSG)          ASK IF THEY WANT TO PATCH
403                    *S*           SIO,0     *TYNDD
404                    *S*           TIO,0     *TYNDD
405                    *S*           BCS,12    $-1
406                    *S*           LB,R1     ADDR                GET ANSWER
407                    *S*           CI,R1     'Y'                 IS IT YES
408                    *S*           BNE       EXIT                B IF NOT
409                    *S*           LI,R0     DA(PRNTEST)
410                    *S*           SIO,R2    *LPNDD              EJECT A PAGE
411                    *S*           BCS,CC    $
412                    *S*           TIO,R4    *LPNDD
413                    *S*           BCS,CC    $-1
414                    *S* READCR    EQU       $
415                    *S*           LI,0      DA(READ)            READ A CARD
416                    *S*           SIO,R2    *CRNDD
417                    *S*           BCS,CC    $
418                    *S*           TIO,R4    *CRNDD              WAIT FOR
419                    *S*           BCS,CC    $-1                 I/O DONE
420                    *S* PRNTOK    EQU       $
421                    *S*           LI,R0     DA(PRINT)
422                    *S* PRNTOK0   EQU       $
423                    *S*           SIO,R2    *LPNDD              PRINT CARD
424                    *S*           BCS,CC    $                   STOP SOMETHING IS WRONG
425                    *S*           TIO,R4    *LPNDD              WAIT FOR
426                    *S*           BCS,CC    $-1                 I/O TO FINISH
427                    *S* PRNTOK1   EQU       $
428                    *S*           LW,R1     PACHBUF
429                    *S*           BEZ       EXIT                B IF END OF PATCHING
```

```
430                    *S*          CW,R1    L(' EOD')          NO----CHECK FOR
431                    *S*          BNE      PATCH              END OF PATCHES
432                    *S*          LI,R1    0                  WE ARE DONE, SO SET
433                    *S*          STW,R1   PACHBUF            PATCHING FLAG
434                    *S*          LI,R0    DA(PRNTEST)
435                    *S*          B        PRNTOKO            EJECT PAGE
436                    *S* *-------------------------
437                    *S* PATCH    EQU      $
438                    *S*          LI,R8    0                  FLAG FOR 'LOC' CONVERS.
439                    *S*          LD,R6    LOC                GET LOC FIELD
440                    *S*          BAL,R9   CONV               CONVERT TO HEX
441                    *S* *
442                    *S* *  RETURN WITH R10 = LOC.
443                    *S* *
444                    *S*          STW,R10  ADDR
445                    *S*          LI,R8    1                  VALUE CONVERS.
446                    *S*          LD,R6    VALUE              VALUE FIELD
447                    *S*          BAL,R9   CONV               CONVERT TO HEX
448                    *S* *
449                    *S* *  RETURN WITH R10 = VALUE
450                    *S* *
451                    *S*          STW,R10  *ADDR              PATCH MONITOR
452                    *S*          B        READCR             GET NEXT CARD
453                    *S* *
454                    *S* *
455                    *S* *
457                    *S* ***************************
458                    *S* *
459                    *S* *  C O N V  R O U T I N E
460                    *S* *
461                    *S* *       PART OF BOOTPATCH ROUTINE. CONVERTS EIGHT (8)
462                    *S* *       EBC CHARACTERS TO 8 HEX CHARACTERS
463                    *S* *
464                    *S* *  ENTRY : R6-R7  = 8 EBC CHAR.
465                    *S* *           R8    = 0 CONVERT LOC
466                    *S* *                 = 1 CONVERT VALUE
467                    *S* *
```

```
468                     *S* *  EXIT  : R10    = 8 HEX CHAR (CONVERTED)
469                     *S* *
470                     *S* *  ENTERED BY :    BAL,R9
471                     *S* *
472                     *S* ***************************
474                     *S* CONV     EQU      $            ENTRY.........
475                     *S*          CI,R8    0
476                     *S*          BNEZ     CONV1              NO--LOC CONV.
477                     *S*          BR,R7    L(' 000')          YES--5 CHAR LOC BECOMES
478                     *S*          SCD,R6   =24                8 CHAR
479                     *S* CONV1    EQU      $
480                     *S*          LI,R15   8                  8 DIGITS
481                     *S* CONV2    EQU      $
482                     *S*          LB,R11   R6                 GET CHAR
483                     *S*          SLD,R6   8                  TO NEXT CHAR
484                     *S*          CI,R11   '0'
485                     *S*          BGE      $+2
486                     *S*          AI,R11   X'39'              CONVERT
487                     *S*          AI,R11   -'0'               EBC DIGIT
488                     *S*          SCS,R11  =4
489                     *S*          SLD,R10  4
490                     *S*          BDR,R15  CONV2              GET NEXT CHAR.
491                     *S*          B        *R9                ------- RETURN
492                     *S* ***************************
494                     *S* ***************************
495                     *S* *  D A T A  I N F O
496                     *S* ***************************
498                     *S* ***********
499                     *S*          BOUND    8
500                     *S* NEWPAGE  TEXT     '1   '             TOP OF FORM
501                     *S* PRNTBUF  TEXT     'A   '             UPSPACE ONE LINE
502                     *S* PACHBUF  RES      20                 INPUT BUFFER FOR PATCHES
503                     *S* ********************
504                     *S* CC       EQU      X'C'               CONDITION CODE FOR I/O CHECK
505                     *S* LOC      EQU      PACHBUF+0          WD-0 IN BUF OF LOC (2 WDS TOTAL)
506                     *S* VALUE    EQU      PACHBUF+2          WD-2 IN BUF OF VALUE (2WDS TOTAL)
507                     *S* ******
```

```
508                     *S* CRNDD   DATA    0003                CR ADDR
509                     *S* LPNDD   DATA    0002                LP ADDR
510                     *S* TYNDD   DATA    0001                TY ADDR
511                     *S* ******
512                     *S* ADDR    PZE     0                   WHERE TO PATCH
513                     *S* ********
514                     *S*         BOUND   8
515                     *S* STKSZ   EQU     16
516                     *S* STK     PZE     *+1
517                     *S*         GEN,16,16 STKSZ,0
518                     *S*         RES     STKSZ
519                     *S* ******
521                     *S* *******
522                     *S*         BOUND   8
523                     *S* READ    EQU     *                   IOCD TO READ A CARD
524                     *S*         GEN,8,24  X'06',BA(PACHBUF)
525                     *S*         GEN,8,24  X'00',20*4
526                     *S* ******
527                     *S*         BOUND   8
528                     *S* PRINT   EQU     *                   IOCD TO PRINT A CARD
529                     *S*         GEN,8,24  X'05',BA(PRNTBUF)
530                     *S*         GEN,8,24  X'00',21*4
531                     *S* ******
532                     *S*         BOUND   8
533                     *S* PRNTEST EQU     *                   IOCD TO DETERMINE PRINTER TYPE
534                     *S*         GEN,8,24  X'05',BA(NEWPAGE)       AND EJECT PAGE
535                     *S*         GEN,8,24  X'00',1*4
536                     *S* ******
537                     *S* ************
538                     *S* ****************
539                     *S* EXIT    EQU     *
540                     *S*         LCI     0
541                     *S*         PLM,R0  STK
542                     *S*         LPSD,0  TRP51               RETURN
543                     *S* ***************************
544                     *S*         BOUND   8
545                     *S* TRP51   DATA    0,0,PACHER,0
```

```
546                     *S* *
547                     *S* *
548                     *S*         BOUND    8
549                     *S* *
550                     *S* PTHMSG  GEN,8,24 X'05',BA(BUSH)
551                     *S*         GEN,8,24 X'20',BUSHSZE
552                     *S*         GEN,8,24  X'86',BA(ADDR)
553                     *S*         GEN,8,24 X'00',4
554                     *S* *
555                     *S* *
556                     *S* BUSH  TEXT  ' IF YOU WANT TO PATCH, ANSW, YES  '
557                     *S* BUSHSZE EQU     BA($)-BA(BUSH)
558                     *S* *
559                     *S* *
560                     *S* PATCHEE RES    50
561                     *S* *
562                         FIN
563          01 00000       END     INITIAL

CONTROL SECTION SUMMARY: 01 000E2   PT 0
```

```
*    SYMBOL VALUES
     AIOPSD/01 000C2        AIOSRV/01 000DE        ANSPROC/00000000       BITS/00000001
     CJOBACC/01 00098       DCBPROC/00000000       DEBUG/00000000         DISCBPROC/00000000
     FIVES/01 00095         HLTPSD/01 000BC        INITPSD/01 000C0       INIT2/01 00001
     IOXPSDS/01 000BA       LACBF/01 00052         LOADMAP/01 0004C       LOADWL/01 00058
     LOOP/01 000BE          LOOP5/01 00093         LWL/01 0007B           LWL1/01 00086
     MAPINC/01 00097        MAPINIT/01 00096       MONPROC/00000000       MPBITS/00000000
     MTWO/01 000B2          M1/00000001  S         M10/0000000A  S        M11/0000000B  S
     M12/0000000C  S        M13/0000000D  S        M14/0000000E  S        M15/0000000F  S
     M16/00000010  S        M17/00000011  S        M18/00000012  S        M19/00000013  S
     M2/00000002  S         M20/00000014  S        M21/00000015  S        M22/00000016  S
     M23/00000017  S        M24/00000018  S        M25/00000019  S        M26/0000001A  S
     M27/0000001B  S        M28/0000001C  S        M29/0000001D  S        M3/00000003  S
     M30/0000001E  S        M31/0000001F  S        M32/00000020  S        M4/00000004  S
     M5/00000005  S         M6/00000006  S         M7/00000007  S         M8/00000008  S
     M9/00000009  S         NOIOWL/01 00076        NOSREL/01 00030        OC/00000002
     R0/00000000            R1/00000001            R10/0000000A           R11/0000000B
     R12/0000000C           R13/0000000D           R14/0000000E           R15/0000000F
     R2/00000002            R3/00000003            R4/00000004            R5/00000005
     R6/00000006            R7/00000007            R8/00000008            R9/00000009
     S69PROC/00000001       S7ERR/01 000C8         S7INST/01 000D5        S7MSG/01 000C6
     S7TST/01 000D4         S7TXT/01 000D6         S9AIOCC/01 000E1       UFLAGS/00000000
     UTSPROC/00000001       WD/01 00043            WLZAP/01 0007A         XPSDS/01 00099
     X1/00000001  S         X10/00000005  S        X100/00000009  S       X1000/0000000D  S
     X2/00000002  S         X20/00000006  S        X200/0000000A  S       X2000/0000000E  S
     X4/00000003  S         X40/00000007  S        X400/0000000B  S       X4000/0000000F  S
     X8/00000004  S         X80/00000008  S        X800/0000000C  S       X8000/00000010  S
     Y0001/00000011  S      Y0002/00000012  S      Y0004/00000013  S      Y0008/00000014  S
     Y001/00000015  S       Y002/00000016  S       Y004/00000017  S       Y008/00000018  S
     Y01/00000019  S        Y02/0000001A  S        Y04/0000001B  S        Y08/0000001C  S
     Y1/0000001D  S         Y2/0000001E  S         Y4/0000001F  S         Y8/00000020  S
*    EXTERNAL DEFINITIONS
     INITIAL/01 00000       INITIAL:/01 00000      LMA/01 00049           TOPROBT/01 00099
     UMOV#/01 0006B
*    PRIMARY REFERENCES
     BOOTHLT        BOOTSBAND      BT31T80        C$CPU          CAL1PSD        CAL2PSD        CAL3PSD
     CAL4PSD        CLK3PSD        CLK4PSD        CORED          CSED$PATH      DCT1           DECFPSD
```

```
        FIXAVPSD        FLTFPSD         HGP             IOHIGH          IOLOW           IOPSD           IPT47
        JIT             LEEPO           LKIMG           LSWAP           MASKS           MB:GAM4         MB:GPT
        MING            MONINIT         MONORG          NOPPSD          NSWAP           OB:BOPTX        OCPSD
        PB:HVA          POWROFF         POWRON          PPSTART         PSA$END         PSD$I56         PSD$I57
        PSD$T4C         PSD$T4D         PSD$T46         S:CLOCK4        S:GJOBTBL       SL:CORE         SPOOL
        STKLPSD         SYSVERS         T:GJOBSTRT      T:SE            T:SGRNU         TINC            TSTACK
        UNIMPPSD        !BIG            !B560           !B9
* NO SECONDARY REFERENCES
* NO UNDEFINED SYMBOLS
*    ERROR SEVERITY LEVEL: 0
* NO ERROR LINES
```