```
                                    ;23625          ;--------------------------------;
                                    ;23626  =100    ;BRANCH ON 1.PART AND FINISH-BITS OF STATE
                                    ;23627          ;100-----------------------------;
                                    ;23628          ALU_0(A),R[R15]_ALU,            ;  CLEAR R15
U 0EA4, 0003,003C,0180,FAF8,0000,01B0 ;23629        J/DIVC0                         ;  READ DIVIDEND INTO ID-REGISTERS
                                    ;23630          ;101-----------------------------;
                                    ;23631          VA_R[SP],SC_K[.30],             ;  INITIALIZE VA AND ID-POINTER
U 0EA5, 0000,003C,79F3,FA70,0284,6854 ;23632        Q_0,J/DIV.R4                    ;  AND COUNTER
                                    ;23633          ;T10-----------------------------;
                                    ;23634          ;NOT PART OF INITIAL BRANCH,AS "10" SHOULD NOT OCCUR
                                    ;23635  DIV.R5: ID(SC)_D,VA_VA+4,               ;  STO  IN ID-REGISTER
                                    ;23636          SC_SC+T,ALU_0+Q+1,Q_ALU,        ;  INCREMENT COUNTER AND POINTER
U 0EA6, 001F,0110,01C0,3403,0080,C854 ;23637        Z?,J/DIV.R4
                                    ;23638          ;111-----------------------------;
                                    ;23639  =111    ALU_RC[T5],D_ALU.LEFT2,
U 0EA7, 0870,0038,0180,F928,C000,0C46 ;23640        J/DIVS2                         ;  WRITE LAST BYTE OF QUOTIENT
                                    ;23641  =;END   ;--------------------------------;
                                    ;23642  =0      ;BRANCH ON ALU Z-BIT
                                    ;23643          ;0-------------------------------;
                                    ;23644  DIV.R4: D[LONG]_CACHE.WCHK,             ;  READ LONGWORD OFF THE STACK
                                    ;23645          ALU_Q-K[.3],CLK.UBCC,           ;  CLOCK COUNTER
U 0854, 0019,2000,0D80,5000,0010,0EA6 ;23646        J/DIV.R5
                                    ;23647          ;1-------------------------------;
                                    ;23648          D_R[R4],CLK.UBCC,BYTE,          ;  GET QUOTIENT LENGTH
U 0855, 0800,973C,0180,FA20,0010,0606 ;23649        STATE3-0?,J/DIV.R6              ;  TEST FOR INTERRUPTS
                                    ;23650  =;END   ;--------------------------------;
                                    ;23651  =011*   ;BRANCH ON DIV.INTRPT. BIT OF STATE
                                    ;23652          ;--------------------------------;
                                    ;23653  DIV.R6: ;  ENTER HERE AFTER READING DIVISOR AND DIVIDEND FROM MEMORY
                                    ;23654          ;011*----------------------------;
                                    ;23655          ALU_D-K[.1],D_ALU,
U 0606, 0819,0000,0580,F800,0000,0CBB ;23656        J/DIV.R7
                                    ;23657          ;111*----------------------------;
U 060E, 0000,003C,0180,F800,0000,01B9 ;23658        J/DIVC1                         ;  RETURN AFTER AN INTERRUPT
                                    ;23659  =;END   ;--------------------------------;
U 0CBB, 0000,003C,01C0,FA28,0000,0CBC ;23660  DIV.R7: Q_R[R5]                        ;  GET QUOTIENT-ADDRESS
                                    ;23661          ;--------------------------------;
                                    ;23662          ALU_Q-K[.1],R[R5]_ALU,          ;  DECREMENT QUOTIENT-ADDRESS
U 0CBC, 0019,2000,0580,FAA8,0000,0C42 ;23663        J/DIVSAV                        ;  JOIN ROUTINE TO WRITE QUOTIENT-BYTE
                                    ;23664
                                    ;23665  .LIST           ;Re-enable full listing
```

```
                                        ;23666  .TOC    'RPPR.MIC''
                                        ;23667  .TOC    'Revision 0.4''
                                        ;23668  ;        P. R. Guilbault
                                        ;23669
;23670  .NOBIN
;23671  .TOC    ''      Revision History''
;23672
;23673  ; 00    Start of history
;23674
                                        ;23675  .BIN
                                        ;23676  .NOLIS          ;Disable listing of PCS code for quickie assemblies
```

```
                                ;23677  .TOC     "       RET, PUSHR, and POPR  : RET"
                                ;23678
                                ;23679  ;       RET - RETURN FROM CALLG/CALLS SUBROUTINE - ENTER FROM A FORK.
                                ;23680
                                ;23681  088:    ;-------------------------------;
U 008B, 0000,C03C,3DF0,2E68,0200,0CC0   ;23682  RET:    LAB_R[FP], VA_LA, Q_ID[PSL]     ; LOAD VA FROM FP AND GET PSL
                                ;23683
                                ;23684  ; ***********************************************
                                ;23685  ; * Patch no. 010, PCS 008B trapped to WCS 114A *
                                ;23686  ; ***********************************************
                                ;23687
                                ;23688          ;-------------------------------;
U 0CC0, 0019,2024,C1C0,F803,0000,0CC1   ;23689          VA_VA+4, Q_Q.ANDNOT.K[.FFFF]    ; BUMP VA TO PSW/MASK, CLR PSL<L>
                                ;23690
                                ;23691          ;-------------------------------;
                                ;23692          D[LONG]_CACHE,                  ; READ IN PSW/MASK
U 0CC1, 0078,0038,8380,4180,0000,0CC2   ;23693          RC[TO]_R[.3FF].LEFT2, SI/MUL-   ; RC[TO] GETS 0FFF
                                ;23694
                                ;23695          ;-------------------------------;
                                ;23696          ALU_D.0XT[BYTE]+Q, D_ALU, Q_D,  ; FORM NEW PSL IN D, Q<31:16> GETS MASK
U 0CC2, 081F,9814,65E0,F803,0084,6B8D   ;23697          VA_VA+4, SC_K[.10], TC0,F803,0000,0CC1  ;23689          VA_VA+4, Q_Q.ANDNOT.
                                ;23690
                                ;23691          ;-------------------------------;
                                ;23692          D[LONG]_CACHE,                  ; READ IN PSW/MASK
U 0CC1, 0078,0038,8380,4180,0000,0CC2   ;23693          RC[TO]_R[.3FF].LEFT2, SI/MUL-   ; RC[TO] GETS 0FFF
                                ;23694
                                ;23695          ;-------------------------------;
                                ;23696          ALU_D.0XT[BYTE]+Q, D_ALU, Q_D,  ; FORM NEW PSL IN D, Q<31:16> GETS MASK
U 0CC2, 081F,9814,65E0,F803,0084,6B8D   ;23697          VA_VA+4, SC_K[.10], D.BYTES?    ; SC GETS SHIFT CT, TEST PSW<15:8>=0
                                ;23698
                                ;23699  =1101
                                ;23700          ;0-------------------------------;
                                ;23701          ID[PSL]_D, D_DAL.SC, VA_VA+4,   ; SAVE NEW PSL, GET MASK IN D
U 0B8D. 0D00,0D3C,3D87,3C03,0000,0462   ;23702          SS_0&SD_0, Q31?, J/RET.0        ; SET UP SS TO GET MASK<15>, TEST MASK<15>
                                ;23703
                                ;23704          ;1-------------------------------;
U 0B8F, 0000,003C,0180,F800,0000,0106   ;23705          J/RSVOPR                        ; NASTY MAN - PAPA SPANK!
                                ;23706
                                ;23707  =01*                                    ; PSL<31>=0 (NOT COMPATIBILITY MODE)
                                ;23708          ;0-------------------------------; MASK<15>=0
                                ;23709  RET.0:  D_D.AND.RC[TO], STATE_D(EXP),   ; D=MASK<11:0>, STATE=MASK<14:7>,
                                ;23710          SC_D(EXP)(A),                   ; SC = MASK<14:7> ALSO
U 0462, 0811,0034,0180,F903,1498,6CC3   ;23711          VA_VA+4, CLK.UBCC, J/RET.1      ; SET Z ON MASK<11:0>
                                ;23712
                                ;23713          ;1-------------------------------; MASK<15> = 1
                                ;23714          D_D.AND.RC[TO], STATE_D(EXP),   ; D=MASK<11:0>, STATE=MASK<14:7>,
                                ;23715          SD_NOT.SD, SC_D(EXP)(A),        ; SET SD TO MASK<15>, SC = MASK<14:7>
U 0466, 0811,0034,0183,F903,1498,6CC3   ;23716          VA_VA+4, CLK.UBCC, J/RET.1      ; SET Z ON MASK<11:0>
```

```
                                    ;23717                ;-------------------------------;
                                    ;23718  RET.1:  SS_SD, FE_SC.ANDNOT.K[.7F],     ; ISOLATE MASK<14> IN FE<7> & SET EALU.Z
                                    ;23719          CLR.UBCC, VA_VA+4, RC[TO]_LA,   ; SAVE FP FOR MPOP, VA --> 1ST SAVED REG
U 0CC3, 0000,013C,5982,F983,0114,4858 ;23720        Z?                              ; TEST IF MASK<11:0> ZERO
                                    ;23721
                                    ;23722  =0
                                    ;23723          ;0-------------------------------; MASK NOT ZERO
                                    ;23724          Q_RC[TO], SC_K[ZERO],           ; SET UP FOR MPOP
U 0858, 0210,0C39,19C0,F900,0084,6EC4 ;23725        D_D.RIGHT2, MUL?, CALL[MPOP]    ; AND CALL IT
                                    ;23726
                                    ;23727          ;1-------------------------------; MPOP RETURNS HERE, OR MASK=0
                                    ;23728          LC_RC[TO], Q_K[.14],            ; Q=FIXED STACK FRAME LEN, LC=BUMPED FP
U 0859, 0018,1238,21C0,F900,1400,4EAA ;23729        STATE_STATE.ANDNOT.FE, EALU?    ; CLEAR STATE<7>, TEST MASK<15:14>
                                    ;23730
                                    ;23731  =*010                                   ; TEST OLD SP ALIGNMENT (EALU.N=0)
                                    ;23732          ;00------------------------------; MASK<15:14> = 01
                                    ;23733          ALU_Q+LC+1, R[SP]_ALU, Q_ALU,   ; SP = BUMPED FP + STACK DELTA + 1
U 0EAA, 0011,3610,01C0,FAF0,0200,0EB3 ;23734        VA_ALU, STATE7-4?, J/RET.4      ; TEST FOR CALLS/CALLG
                                    ;23735
                                    ;23736          ;01------------------------------; MASK<15:14> = 11
U 0EAB, 0019,2014,0DC0,F800,0000,0EAE ;23737        Q_Q+K[.3]                       ; RE-UNALIGN THE STACK DELTA
                                    ;23738
                                    ;23739          ;10------------------------------; MASK<15:14> = 00
                                    ;23740  RET.2:  Q_Q+LC, R[SP]_ALU, VA_ALU,      ; SP = BUMPED FP  +  STACK DELTA
U 0EAE, 0011,3614,01C0,FAF0,0200,0EB3 ;23741        STATE7-4?, J/RET.4              ; TEST FOR CALLS/CALLG
                                    ;23742
                                    ;23743          ;11------------------------------; MASK<15:14> = 10
U 0EAF, 0019,2014,09C0,F800,0000,0EAE ;23744        Q_Q+K[.2], J/RET.2              ; RE-UNALIGN THE STACK DELTA
                                    ;23745
                                    ;23746  =*011                                   ; BRANCH ON STATE<6>
                                    ;23747          ;0-------------------------------; STATE<6>=0 - CALLG
U 0EB3, 0F18,0014,0180,F800,0200,0CC8 ;23748  RET.4:  VA_LA+K[.8], D_0, J/RET.6      ; LA STILL HAS FP, REMEMBER?
                                    ;23749
                                    ;23750          ;1-------------------------------; STATE<6>=1 - CALLS
U 0EB7, 0000,803C,0180,4000,0000,0CC4 ;23751        D[BYTE]_CACHE                   ; GET ARGUMENT COUNT
                                    ;23752
                                    ;23753          ;-------------------------------;
U 0CC4, 081B,8014,0580,F800,0000,0CC5 ;23754        D_D.OXT[BYTE]+K[.1]             ; D = ARG COUNT + 1
                                    ;23755
                                    ;23756          ;-------------------------------;
U 0CC5, 0118,0014,0180,F800,0200,0CC8 ;23757        D_D.LEFT2, VA_LA+K[.8]          ; D = LENGTH OF ARG LIST, VA--> SAVED AP
```

```
                                    ;23758  ;HERE TO FINISH RET BY RESTORING AP, FP, AND PC
                                    ;23759
                                    ;23760          ;-------------------------------;
U 0CC8, 001D,0014,0180,42F0,0000,0CC9 ;23761  RET.6:  R[SP]_D+Q, D[LONG]_CACHE        ; BUMP SP OVER ARG LIST, GET SAVED AP
                                    ;23762
                                    ;23763          ;-------------------------------;
                                    ;23764          R[AP]_D,LONG,                   ;RESTORE ARG POINTER
U 0CC9, 0001,003C,0180,FAE3,0000,0CCA ;23765        VA_VA+4                         ;GET ADDR OF FP
                                    ;23766
                                    ;23767          ;-------------------------------;
U 0CCA, 0000,003C,0180,4000,0000,0CCC ;23768        D[LONG]_CACHE                   ;GET SAVED FP
                                    ;23769
                                    ;23770          ;-------------------------------;
                                    ;23771          Q_D,                            ;SAVE FP UNTIL ALL MEMORY OPS COMPLETE
U 0CCC, 0000,003C,01E0,F803,0000,0CCD ;23772        VA_VA+4                         ;GET ADDR OF PC
                                    ;23773
                                    ;23774          ;-------------------------------;
U 0CCD, 0000,003C,0180,4000,0000,0CD0 ;23775        D[LONG]_CACHE                   ;SAVED PC
                                    ;23776
                                    ;23777          ;-------------------------------;
U 0CD0, 2001,003C,0180,F801,4200,0CD1 ;23778        PC&VA_D,FLUSH.IB                ;START INSTRUCTION BUFFER
                                    ;23779
                                    ;23780          ;-------------------------------;
                                    ;23781          R[FP]_Q,LONG,                   ;RESTORE SAVED FP,
U 0CD1, 0001,203C,0180,62EC,0000,0062 ;23782        LOAD.IB,PC_PC+1,J/IRD           ;GO DO NEXT INSTR
                                    ;23783
                                    ;23784  ; ************************************************
                                    ;23785  ; * Patch no. 068, PCS 0CD1 trapped to WCS 118E *
                                    ;23786  ; ************************************************
```

```
                                        ;23787  .TOC     ""      RET, PUSHR, and POPR  : PUSHR''
                                        ;23788
                                        ;23789  ;HERE TO DO PUSH REGISTERS INSTRUCTION
                                        ;23790  ;D CONTAINS MASK
                                        ;23791
                                        ;23792  289:    ;-------------------------------;
                                        ;23793  PUSHR:  D_D.OXT[WORD].ANDNOT.K[.8000],  ;STRIP MASK TO BITS 14-0
                                        ;23794          RC[T2]_ALU, CLK.UBCC,           ;SAVE FOR LATER, SET Z FOR NBYTES
U 0289, 081B,4025,4580,F990,0010,0CD3   ;23795          CALL,J/NBYTES                   ;GO GET NUMBER OF BITS IN MASK*4
                                        ;23796
                                        ;23797  299:    ;-------------------------------;RETURN HERE WITH NUMB OF BYTES TO PUSH
U 0299, 001C,2000,8D80,F800,0284,6CD2   ;23798          VA_LA-D, SC_K[.1F]              ;LOAD LAST ADDR TO PUSH
                                        ;23799
                                        ;23800          ;-------------------------------;
                                        ;23801          SC_SC-FE,                       ;SC GETS NUMBER OF FIRST REG TO PUSH
U 0CD2, 0000,003C,0180,1000,0080,A870   ;23802          TEST.TB.WCHK                    ;DOES TB RECOGNIZE THIS ADDRESS?
                                        ;23803
                                        ;23804  =0      ;-------------------------------;CALL SITE FOR MPUSH
                                        ;23805          INHIBIT.IB,                     ;PREVENT IB ACCESS FROM MESSING TB TEST
                                        ;23806          D_R(SC),                        ;LOAD FIRST REG TO PUSH INTO D
U 0870, 0800,1D3D,0180,0868,0000,005D   ;23807          TB.TEST?,CALL,J/PUSHR.1         ;VERIFY WRITABILITY
                                        ;23808
                                        ;23809          ;-------------------------------;RETURN HERE FROM MPUSH
                                        ;23810          CLR.IB.OPC,PC_PC+1,             ;GET RID OF THIS INSTRUCTION
U 0871, C000,003C,0180,F804,4000,0062   ;23811          J/IRD                           ;GO TO NEXT
                                        ;23812
                                        ;23813  =011*01 ;-------------------------------;TB HIT AND NOT ACCESS VIOLATION
U 005D, 0019,2000,0D80,F800,0200,0874   ;23814  PUSHR.1:VA_Q-K[.3], J/PUSHR.2           ;Q HAS SP-1 SO VA GETS SP-4
                                        ;23815
                                        ;23816          ;-------------------------------;TB MISS OR ACCESS VIOLATION
                                        ;23817          RC[PC.SV]_PC,PC_VA,             ;SAVE REGISTERS
                                        ;23818          ID[D.SV]_D,D_Q,                 ; IN RC & ID TEMPS
U 005F, 0C14,0039,B980,3DE1,0000,0D12   ;23819          CALL,J/TBF.W                    ;WHILE FILLING IN TRANSLATION BUFFER
                                        ;23820
                                        ;23821  =111*11 ;-------------------------------;RETURN HERE IFF ACCESS VALID
                                        ;23822          D_Q,Q_ID[Q.SV],                 ;FINISH REGISTER RESTORE
U 007F, 0C00,003C,BDF0,2C00,0000,005D   ;23823          J/PUSHR.1                       ;NOW GO START STORING
                                        ;23824
                                        ;23825  =0
                                        ;23826          ;-------------------------------;
                                        ;23827  PUSHR.2:CACHE_D[LONG],                  ;WRITE FIRST REGISTER
U 0874, 0000,003D,0180,3270,0000,0CD9   ;23828          LAB_R[SP], CALL, J/PSHSP        ;PUSH SP BY 4 BYTES
                                        ;23829
                                        ;23830          ;-------------------------------;
U 0875, 0010,0038,01C0,F910,0000,0889   ;23831          Q_RC[T2], J/MPUSH2              ;GET MASK, JUMP INTO MPUSH
```

```
                                        ;23832  .TOC    "       RET, PUSHR, and POPR  : POPR"
                                        ;23833
                                        ;23834  ;       POP REGISTERS INSTRUCTION - ENTER AT B-FORK WITH MASK IN D.
                                        ;23835
                                        ;23836  28A:    ;-------------------------------;
                                        ;23837  POPR:   D_D.OXT[WORD].ANDNOT.K[.8000],  ; KILL BIT 15 IN MASK
                                        ;23838          STATE_AMX.EXP, RC[T1]_ALU,      ; SAVE MASK - STATE<7> GETS MASK<14>
U 028A, 0816,4025,4580,F988,1418,6CD3   ;23839          CLK.UBCC, CALL[NBYTES]          ; SET Z ON MASK, GO COUNT BITS
                                        ;23840
                                        ;23841  29A:    ;-------------------------------; RETURN - D=COUNT*4, LA=R[SP], Q=LA-1
                                        ;23842          STATE_STATE.ANDNOT.K[.7F],      ; CLEAR EXTRANEOUS BITS FROM STATE
U 029A, 001D,2014,5980,F800,1604,4880   ;23843          VA_Q+D                          ; CALC LAST ADDR TO POP
                                        ;23844
                                        ;23845  =0
                                        ;23846          ;0-------------------------------; MASK NOT NULL
U 0880, 0000,003C,01C0,0180,0000,0881   ;23847          RC[TO]_LA, Q_LA, TEST.TB.RCHK   ; GET SP FOR MPOP, TEST READABILITY
                                        ;23848
                                        ;23849          ;-------------------------------;
U 0881, 0810,1D38,0180,0908,0000,0098   ;23850          D_RC[T1], INHIBIT.IB, TB.TEST?  ; IS THE LAST BYTE READABLE?
                                        ;23851
                                        ;23852  =011*00
                                        ;23853          ;00-------------------------------; ALL IS OK
                                        ;23854  POPR.1: VA_LA, SC_K[ZERO], D_D.RIGHT2,  ; SET UP VA AND SC FOR MPOP,
U 0098, 0200,0C3D,1980,F800,0284,6EC4   ;23855          MUL?, CALL[MPOP]                ; AND CALL IT
                                        ;23856
                                        ;23857          ;01-------------------------------; RETURN FROM MPOP (TB.TEST CANT COME HERE)
                                        ;23858          CLR.IB.OPC, PC_PC+1, LC_RC[TO], ; CLEAR INST FROM IB, LATCH NEW SP,
U 0099, C000,163C,0180,F904,4000,0710   ;23859          STATE7-4?, J/POPR.3             ; SEE IF SP WAS ONE OF THE REGS POPPED.
                                        ;23860
                                        ;23861          ;10-------------------------------; ACCESS VIOLATION
U 009A, 0000,003C,0180,F800,0000,009B   ;23862          J/POPR.2                        ; FAKE A MISS TO SAVE SPACE
                                        ;23863
                                        ;23864          ;11-------------------------------; TB MISS
                                        ;23865  POPR.2: RC[PC.SV]_PC, PC_VA, ID[D.SV]_D,; REFILL THE TB AND RETEST
U 009B, 0C14,0039,B980,3DE1,0000,0D10   ;23866          D_Q, CALL[TBF.R]                ; RETURN[20] IF NO FAULT
                                        ;23867
                                        ;23868  =111*11 ;-------------------------------; RETURN FROM TBF.R
U 00BB, 0C00,003C,BDF0,2C00,0000,0098   ;23869          D_Q, Q_ID[Q.SV], J/POPR.1       ; FINISH RESTORING STATE & CONTINUE
                                        ;23870
                                        ;23871  =0***
                                        ;23872          ;0-------------------------------; SP WAS NOT POPPED
U 0710, 0010,0038,0180,FAF0,0000,0718   ;23873  POPR.3: R[SP]_LC                         ; RESTORE SP TO ITS BUMPED VALUE
                                        ;23874
                                        ;23875          ;1-------------------------------; SP WAS POPPED
U 0718, F80C,003B,01F1,F857,139B,6000   ;23876          IRD                             ; THEN WE ARE DONE!
```

```
                                ;23877  ;HERE IS SUBROUTINE TO RETURN THE NUMBER OF BYTES TO BE PUSHED OR POPPED
                                ;23878  ; ENTER WITH MASK IN D
                                ;23879
                                ;23880          ;-------------------------------;
                                ;23881  NBYTES: D_D.,MAP,                        ;MOVE MASK BITS TO 31-16 FOR COUNTING
                                ;23882          SC_K[.10].ALU,                  ;PUT 16 IN SC FOR LATER SHIFTING
U OCD3, 0B18,0138,6580,F800,018E,6884   ;23883          FE_SHF.VAL, Z?          ;HIGHEST BIT GOES TO FE, CHECK IF 0
                                ;23884
                                ;23885  =0      ;-------------------------------;
U 0884, 0000,003C,3580,3C00,0000,0CD4   ;23886          ID[VECTOR]_D, J/NBYTE1  ;SEND MASK BITS TO BE COUNTED
                                ;23887
                                ;23888          ;-------------------------------;
U 0885, C000,003C,0180,F804,4000,0062   ;23889          CLR.IB.OPC, PC_PC+1, J/IRD      ;NULL MASK MEANS NULL OPERATION
                                ;23890
                                ;23891          ;-------------------------------;
                                ;23892  NBYTE1: Q_ID[VECTOR],D_0,               ;READ BACK THE BIT COUNT
U OCD4, 0F00,003C,35F0,2E70,0000,0CD5   ;23893          LAB_R[SP]               ;PICK UP STACK POINTER
                                ;23894
                                ;23895          ;-------------------------------;
                                ;23896          D_DAL.SC,                       ;GET NUMBER OF REGISTERS TO PUSH
U OCD5, 0D18,0000,05C0,F800,0000,0CD6   ;23897          Q_LA-K[.1]              ;GET SP-1 FOR CALCULATING SP+N-1
                                ;23898
                                ;23899          ;-------------------------------;
                                ;23900          D_D.AND.K[.F].LEFT2,            ;CALCULATE NUMB OF BYTES TO PUSH
U OCD6, 0879,0036,6180,F800,0000,0010   ;23901          RETURN10
```

```
                                    ;23902  ;MPUSH -- SUBROUTINE TO PUSH REGISTERS ACCORDING TO MASK
                                    ;23903  ;CALL WITH REGISTER MASK IN D<13:0>, ZEROS IN D<31:14>
                                    ;23904  ;       1F IN SC, SP IN LA.  Q AND SC GET CLOBBERED
                                    ;23905  ;RETURNS WITH VA DECREMENTED AN EXTRA TIME, PC IN D
                                    ;23906
                                    ;23907          ;-------------------------------;
                                    ;23908  MPUSH:  VA_LA-K[.4],                    ;GET NEXT PUSH ADDRESS
                                    ;23909                                          ;MUST NOT UPDATE SP IN CASE STORING IT
                                    ;23910          SC_SC-SHF.VAL,                  ;WHERE IS MSB OF D?
U 0CD8, 0018,0D00,1180,F800,028C,AEBD ;23911        D.NE.0?                         ;AND IS THERE ONE?
                                    ;23912
                                    ;23913  =101    ;0------------------------------;D.EQL.0
                                    ;23914          D_PC,                           ;RETURN PC IN D
U 0EBD, 0814,003A,0180,F800,0000,0001 ;23915        RETURN1                         ;DONE.
                                    ;23916
                                    ;23917          ;1------------------------------;D.NEQ.0
                                    ;23918          D_R(SC),                        ;GET INDICATED REGISTER
U 0EBF, 0800,003C,01E0,F868,0000,0888 ;23919        Q_D                             ;SAVE MASK
                                    ;23920
                                    ;23921  =0      ;-------------------------------;CALL SITE
                                    ;23922          CACHE_D[LONG],                  ;STORE SELECTED REGISTER
                                    ;23923          LAB_R[SP],                      ;READ OUT SP AGAIN
U 0888, 0000,003D,0180,3270,0000,0CD9 ;23924        CAL[,J/PSHSP                    ;UPDATE IT IN RAB
                                    ;23925
                                    ;23926          ;-------------------------------; ENTER HERE FROM PUSHR
                                    ;23927  MPUSH2: D_Q.AND.MASK,                   ;CLEAR MSB OF MASK
                                    ;23928          LAB_R[SP],                      ;GET UPDATED SP INTO LATCH
U 0889, 0801,2034,8D80,FA70,0084,6CD8 ;23929        SC_K[.1F],J/MPUSH               ;RESET SC TO 31.
                                    ;23930
                                    ;23931  ;PSHSP -- SUBROUTINE TO DECREMENT SP BY 4, LOADING RESULT INTO VA
                                    ;23932
                                    ;23933          ;-------------------------------;
                                    ;23934  PSHSP:  R[SP]&VA_LA-K[.4].RLOG,         ;PUT PUSHED SP INTO SP & VA
U 0CD9, 0018,0006,1180,FAF0,0200,0001 ;23935        RETURN[1]
```

```
                                ;23936  ; MPOP - SUBROUTINE TO POP REGISTERS FROM STACK ACCORDING TO A MASK.
                                ;23937
                                ;23938  ;       AT THE CALL STATE, D CONTAINS THE REGISTER MASK IN BITS 15-0 WITH
                                ;23939  ;       D<31:16>=0. D  MUST NOT BE 0 (!!). VA SHOULD POINT AT THE
                                ;23940  ;       ADDRESS OF THE FIRST REGISTER IMAGE.
                                ;23941  ;       THE CALL STATE SHOULD BE:
                                ;23942
                                ;23943  ;                       SC_0, Q_RC[TO], D_D.RIGHT2, MUL?, CALL[MPOP]
                                ;23944
                                ;23945  ;       RETURN IS TO CALL.OR.1 WITH D=SC=0. RC[TO] WILL BE INCREMENTED BY THE
                                ;23946  ;       NUMBER OF BYTES POPPED.
                                ;23947  ;       LEAVES ALU CC'S AND LA, LB ALONE
                                ;23948
                                ;23949  =100
                                ;23950          ;00-----------------------------;  LOW TWO BITS OF D ARE 00
                                ;23951  MPOP:   SC_SC+K[.2], Q_RC[TO],          ;  UPDATE REGISTER NUMBER, GET COUNT
U OEC4, 0210,0C38,09C0,F900,0084,8EC4   ;23952          D_D.RIGHT2, MUL?, J/MPOP        ;  TEST NEXT TWO BITS OF MASK
                                ;23953
                                ;23954          ;01-----------------------------;  LOW 2 BITS OF MASK ARE 01
                                ;23955          D[LONG]_CACHE, RC[TO]_Q+K[.4],  ;READ LONGWORD, BUMP COUNTER
U OEC5, 0019,2D14,11E0,4180,0000,0ECD   ;23956          Q_D, D.NE.0?, J/MPOP.T          ;SAVE MASK, SEE IF EXHAUSTED
                                ;23957
                                ;23958          ;10-----------------------------;  LOW 2 BITS OF MASK ARE 10
                                ;23959          D[LONG]_CACHE, RC[TO]_Q+K[.4],  ;READ LONGWORD, BUMP COUNTER
U OEC6, 0019,2D14,11E0,4180,0080,CECD   ;23960          SC_SC+1, Q_D, D.NE.0?, J/MPOP.1 ;BUMP REG #, SAVE MASK, SEE IF DONE
                                ;23961
                                ;23962          ;11-----------------------------;  LOW 2 BITS OF MASK ARE 11
U OEC7, 0019,2014,01E0,4180,0000,0CDA   ;23963          D[LONG]_CACHE, Q_D, RC[TO]_Q+K[.8] ; READ FIRST OF 2 LONGWORDS, BUMP CTR
                                ;23964
                                ;23965          ;-------------------------------;
U OCDA, 0C01,003C,0180,F8EB,0000,0CDB   ;23966          R(SC)_D, VA_VA+4, D_Q           ;SAVE REGISTER, BUMP ADDR, RESTORE MASK
                                ;23967
                                ;23968          ;-------------------------------;
U OCDB, 0000,0D3C,0180,4000,0080,CECD   ;23969          D[LONG]_CACHE, SC_SC+1, D.NE.0? ;READ 2ND LONGWD, BUMP REG #, SEE IF DONE
                                ;23970
                                ;23971  =101
                                ;23972          ;0------------------------------;  HERE WHEN MASK EXHAUSTED
U OECD, 0001,003E,1980,F8E8,0084,6001   ;23973  MPOP.1: R(SC)_D, SC_K[ZERO], RETURN[1]  ;SAVE LAST REGISTER, CLEAR SC, RETURN
                                ;23974
                                ;23975          ;1------------------------------;  MASK NOT EXHAUSTED
                                ;23976          R(SC)_D, D_Q, VA_VA+4,          ;SAVE REGISTER, RESTORE MASK, BUMP VA
U OECF, 0C01,003C,0580,F8EB,0084,4EC4   ;23977          SC_SC.ANDNOT.K[.T], J/MPOP      ;CLEAR LOW BITS IN SC FOR BUMPING & LOOP
                                ;23978
                                ;23979  .LIST           ;Re-enable full listing
```

23979; This page intentionally left blank.

```
                                        ;23930  .TOC    "REI.MIC"
                                        ;23981  .TOC    "Revision 1.4"
                                        ;23982  ;        P. R. Guilbault
                                        ;23983
;23984  .NOBIN
;23985  .TOC    "       REVISION History"
;23986
;23987  ; 01    Comment patch no. 094 REI CM PC problem.
;23988  ; 00    START OF HISTORY
;23989
                                        ;23990  .BIN
                                        ;23991  .NOLIST          ;Disable listing of PCS code for quickie assemblies
```

```
                                    ;23992  .TOC    "       REI instruction       : REI"
                                    ;23993
                                    ;23994  ;REI -- Return from exception or interrupt"
                                    ;23995  ;ENTER DIRECTLY FROM IRD, WITH NOTHING SET UP.
                                    ;23996
                                    ;23997  08D:    ;-------------------------------;
U 008D, 0000,003C,0180,FA70,0200,0CDC ;23998  REI:    VA_R[SP]                        ;READY TO POP SAVED PC FROM STACK
                                    ;23999
                                    ;24000          ;-------------------------------;
U 00DC, 0000,003C,0180,4000,0000,0CDD ;24001          D[LONG]_CACHE                   ;HERE'S THE PC
                                    ;24002
                                    ;24003          ;-------------------------------;
                                    ;24004          R[T1]_D,                        ;SAVE THE NEW PC
                                    ;24005          SC_K[.2],                       ;SETUP SC FOR LEFT SHIFT 2
U 0CDD, 0001,003C,0980,F98B,0084,6CDE ;24006          VA_VA+4                         ;ADDRESS THE PSL
                                    ;24007
                                    ;24008          ;-------------------------------;
                                    ;24009          R[SP]_LA+K[.8].RLOG,            ;UPDATE SP FOR TWO POPS
U 0CDE, 0018,0018,0180,42F0,0000,0CE0 ;24010          D[LONG]_CACHE                   ;WHILE GETTING PSL INTO D
                                    ;24011
                                    ;24012          ;-------------------------------;
                                    ;24013          R[T2]_D,Q_D,                    ;SAVE IT IN CASE WE NEED IT
                                    ;24014          D_DAL.SC,                       ;SHIFT PROPOSED PSL LEFT 2
U 0CE0, 0D01,0D3C,01EC,F990,0000,0ED6 ;24015          D31?                            ;COMPATABILITY MODE?
                                    ;24016
                                    ;24017  =110    ;0-------------------------------;D31=0, VAX MODE
U 0ED6, 0E00,0C3C,0180,F800,0000,0CE5 ;24018          D_D.SWAP,J/REI.01               ;PROPOSED PREVIOUS MODE TO D1-0
                                    ;24019
                                    ;24020          ;1-------------------------------;D31=1, NEW PSL IS COMPATABILITY MODE
                                    ;24021          ALU_Q.ANDNOT.K[.1F],            ;CLEAR TBIT AND CONDITION CODES
U 0ED7, 0879,2024,8D80,F800,0000,0CE1 ;24022          D_ALU.LEFT2,SI/ZERO             ;SHIFT AWAY CM AND TP
                                    ;24023
                                    ;24024  ;THE NEW (COMPATABILITY MODE) PSL IS IN D, SHIFTED LEFT TWO PLACES,
                                    ;24025  ; AND WITH TBIT AND THE CONDITION CODES CLEARED.  THE ONLY BITS, OF THOSE
                                    ;24026  ; REMAINING, WHICH MAY BE ONES ARE THE CURRENT AND PREVIOUS MODE FIELDS,
                                    ;24027  ; WHICH MUST BOTH BE 11 (USER).
                                    ;24028
                                    ;24029          ;-------------------------------;
                                    ;24030          Q_ID[PSL],                      ;GET PSL FOR CURRENT MODE
U 0CE1, 0B00,003C,3DF0,2C00,0000,0CE2 ;24031          D_D.SWAP                        ;BYTES SWAPPED FOR COMPARE
                                    ;24032                                          ;THIS MOVES CURRENT MODE TO D3-2.
                                    ;24033                                          ; PREVIOUS MODE TO D1-0.
                                    ;24034
                                    ;24035          ;-------------------------------;
                                    ;24036          ALU_D.XOR.K[.F],                ;COMPARE AGAINST REQUIRED VALUE
                                    ;24037          LONG,CLK.UBCC,                  ;ALU Z WILL BE SET IFF PSL OK
U 0CE2, 0C19,0020,6180,F800,0010,0CE3 ;24038          D_Q                             ;MOVE CURRENT PSL TO D
                                    ;24039
                                    ;24040          ;-------------------------------;
U 0CE3, 0B00,003C,0180,F800,0000,0CE4 ;24041          D_D.SWAP                        ;MOVE CURRENT MODE OF PSL TO D2-0
                                    ;24042
                                    ;24043          ;-------------------------------;
                                    ;24044          SC_D.OR.K[.28],                 ;SETUP ID BUS ADDR OF THIS SP
U 0CE4, 0019,0030,2D80,F800,0082,0CF3 ;24045          J/REI.08                        ;REJOIN COMMON CODE
```

```
                                ;24046  ;HERE TO DO REI, WITH VAX-MODE PSL POPPED.  NEW PSL IS BYTE SWAPPED IN D
                                ;24047  ;       OLD PSL IS IN Q
                                ;24048
                                ;24049          ;-------------------------------;
                                ;24050  REI.01: SC_D.AND.K[.3],                 ;T2<PREVIOUS MODE> TO SC
U 0CE5, 0C19,0034,0D80,F800,0082,0CE8   ;24051          D_Q                     ;ORIGINAL T2 BACK IN D
                                ;24052
                                ;24053          ;-------------------------------;
                                ;24054          D_D.SWAP,                       ;T2<CURRENT MODE> TO D1-0
                                ;24055          FE_SC+1,                        ; PREVIOUS MODE+1 TO FE
U 0CE8, 0B00,003C,3DF0,2C00,0100,CCE9   ;24056          Q_ID[PSL]               ;GET CURRENT PSL
                                ;24057
                                ;24058          ;-------------------------------;
                                ;24059          SC_D.AND.K[.3],                 ;T2<CURRENT MODE> TO SC
                                ;24060          D_Q,                            ;PSL TO D
                                ;24061          Q_D,                            ;T2 (BYTE SWAPPED) TO Q
U 0CE9, 0C19,1934,0DE0,F800,0082,0B9B   ;24062          D2?                     ;TEST T2<IS>
                                ;24063
                                ;24064  =1011   ;0-------------------------------;T2<IS>=0
                                ;24065          D_D.SWAP,                       ;GET PSL<CURRENT MODE> TO D1-0
                                ;24066          EALU_SC-FE,CLK.UBCC,            ;COMPARE T2<PREV> TO T2<CURR>
U 0CEA, 0B00,003C,0180,F800,0010,A09D   ;24067          J/REI.02                ;THEN COMPARE T2<CURR> WITH PSL<CURR>
                                ;24068
                                ;24069          ;1-------------------------------;
                                ;24070          D_D.SWAP,                       ;GET PSL<CURRENT MODE> TO D1-0
                                ;24071          EALU_SC-FE,CLK.UBCC,            ;COMPARE T2<PREV> TO T2<CURR>
U 0B9F, 0B00,1C3C,0180,F800,0010,A09D   ;24072          PSL.MODE?               ;AND VERIFY PSL<IS>=1
                                ;24073
                                ;24074  =11101  ;0-------------------------------;PSL<IS>=1
                                ;24075  REI.02: FE_SC+1,                        ;T2<CURRENT MODE> +1 TO FE
                                ;24076          SC_D.AND.K[.3],                 ;GET CURRENT MODE FROM PSL
                                ;24077          D_Q,                            ;GET T2 SWAPPED INTO D
U 009D, 0C19,1234,0D80,F800,0182,CBA7   ;24078          EALU.N?,J/REI.03        ;IS T2<PREV> LSS T2<CURR>?
                                ;24079
                                ;24080          ;1-------------------------------;PSL<IS>=0
U 009F, 0000,003C,0180,F800,0000,0106   ;24081          J/RSVOPR                ;T2<IS>=1, PSL<IS>=0
                                ;24082
                                ;24083  =0111   ;0-------------------------------;EALU N=0
U 0BA7, 0000,003C,0180,F800,0000,0106   ;24084  REI.03: J/RSVOPR                 ;T2<PREV> LSS T2<CURRENT MODE>
                                ;24085
                                ;24086          ;1-------------------------------;EALU N=1
                                ;24087          EALU_SC-FE,CLK.UBCC,            ;COMPARE T2<CURR> TO PSL<CURR>
U 0BAF, 0000,003C,3DF0,2C00,0010,ACEA   ;24088          Q_ID[PSL]               ;READY TO TEST IPL
```

```
                                        ;24089  ;HERE TO FINISH VAX-MODE CHECKS ON NEW PSL VALIDITY
                                        ;24090
                                        ;24091          ;--------------------------------;
                                        ;24092          Q_D.AND.K[.1F00],               ;GET IPL OUT OF T2
                                        ;24093          D_Q,                            ;MOVE PSL TO D
U 0CEA, 0C19,1234,91C0,F800,0000,0BB7   ;24094          EALU.N?                         ;IS T2<CURR> .GEQ. PSL<CURR>?
                                        ;24095
                                        ;24096  =0111   ;0--------------------------------;EALU N=0
U 0BB7, 0000,003C,0180,F800,0000,0106   ;24097          J/RSVOPR                        ;NO, BASH EM
                                        ;24098
                                        ;24099          ;1--------------------------------;EALU N=1
U 0BBF, 0B00,003C,0180,F800,0000,0CEB   ;24100          D_D.SWAP                        ;SETUP PSL FOR GETTING IPL
                                        ;24101
                                        ;24102          ;--------------------------------;
                                        ;24103          D_D.AND.K[.1F00],               ;GET IPL OUT OF PSL
U 0CEB, 0819,1934,9180,F800,0000,0BCB   ;24104          D2?                             ;LOOK AT PSL<IS>
                                        ;24105
                                        ;24106  =1011   ;0--------------------------------;PSL<IS>=0
                                        ;24107  REI.04: ALU_D-Q,CLK.UBCC,               ;COMPARE IPL'S
                                        ;24108          SC_SC+K[.28],                   ;GET APPROPRIATE SP ADDR IN SC
U 0BCB, 001D,0000,2D80,F800,0094,8CF0   ;24109          J/REI.05
                                        ;24110
                                        ;24111          ;1--------------------------------;PSL<IS>=1
U 0BCF, 0000,003C,1180,F800,0084,6BCB   ;24112          SC_K[.4],J/REI.04               ;SETUP ADDRESS OF ISP
                                        ;24113
                                        ;24114          ;--------------------------------;
                                        ;24115  REI.05: D_RC[T2],                       ;GET NEW PSL FOR TESTING MBZ
U 0CF0, 0810,0338,0180,F910,0000,0818   ;24116          C31?                            ;C IS SET IF PSL<IPL> .GEQ. T2<IPL>
                                        ;24117
                                        ;24118  =0*     ;0--------------------------------;ALU C=0
U 0818, 0000,003C,0180,F800,0000,0106   ;24119          J/RSVOPR                        ;T2<IPL> GTR PSL <IPL>
                                        ;24120
                                        ;24121          ;1--------------------------------;ALU C=1
U 081A, 0813,8020,0180,F910,0000,0CF1   ;24122          D_D.OXT[BYTE].XOR.RC[T2]        ;CLEAR T2<7:0> (CC, ETC)
                                        ;24123
                                        ;24124          ;--------------------------------;
U 0CF1, 0B00,003C,0180,F800,0000,0CF2   ;24125          D_D.SWAP                        ;RE-ARRANGE FOR EASIER TEST OF MBZ
                                        ;24126
                                        ;24127          ;--------------------------------;
                                        ;24128          ALU_D.ANDNOT.K[.DFCF],          ;TEST MBZ BITS
U 0CF2, 0019,0024,AD80,F800,0010,0CF3   ;24129          LONG,CLK.UBCC
```

```
                                    ;24130  ;HERE FOR REI, TO CHANGE THE STACK POINTER, PC, AND PSL
                                    ;24131  ; ALU Z IS SET IF THE MBZ BITS IN THE NEW PSL ARE ALL ZERO
                                    ;24132
                                    ;24133          ;-------------------------------;
                                    ;24134  REI.08: D_R[SP],                        ;READY TO SAVE OLD SP
U 0CF3, 0800,003C,3DF0,2E70,0000,0CF4 ;24135        Q_ID[PSL]                       ;GET OLD TP FOR COPY TO NEW
                                    ;24136
                                    ;24137          ;-------------------------------;
                                    ;24138          ID(SC)_D,                       ;SAVE CURRENT SP
                                    ;24139          D_RC[T2],                       ;GET NEW PSL
                                    ;24140          Q_Q.LEFT,                       ;MOVE OLD TP TO Q31
U 0CF4, 0810,0138,01A8,3510,0000,0890 ;24141        Z?                              ;CHECK ON MBZ-NESS
                                    ;24142
                                    ;24143  =0      ;0-------------------------------;ALU Z=0
U 0890, 0000,003C,0180,F800,0000,0106 ;24144        J/RSVOPR                        ;MBZ BITS WEREN'T ZERO
                                    ;24145
                                    ;24146          ;1-------------------------------;ALU Z=1
                                    ;24147  REI.10: ID[PSL]_D,                      ;STUFF NEW PSL
                                    ;24148          D_D.SWAP,                       ;GET IS AND MODE TO D2-0
U 0891, 0B00,0D3C,3D80,3C00,0000,0EDB ;24149        Q31?                            ;LOOK AT OLD TP
                                    ;24150
                                    ;24151  =011    ;0-------------------------------;PSL<TP>=0
                                    ;24152          PC&VA_RC[T1],FLUSH.IB,          ;START INSTR BUFFER AT NEW PC
U 0EDB, 2010,1938,0180,F909,4200,0BD8 ;24153        D2-0?,J/REI.12                  ;LOOK AT NEW IS AND MODE
                                    ;24154
                                    ;24155  ; ***********************************************
                                    ;24156  ; * Patch no. 094, PCS 0EDB trapped ot WCS 11A4 *
                                    ;24157  ; ***********************************************
                                    ;24158
                                    ;24159          ;1-------------------------------;PSL<TP>=1
U 0EDF, 0819,0030,3180,F800,0000,0CF5 ;24160        D_D.OR.K[.40]                   ;SET TP IN NEW PSL
                                    ;24161
                                    ;24162          ;-------------------------------;
                                    ;24163          D_D.SWAP,                       ;RESTORE TO PSL FORMAT TEMPORARILY
U 0CF5, 0B00,003C,01F8,F800,0000,0891 ;24164        Q_0,J/REI.10                    ;PREVENT COMING BACK HERE
```

```
                                        ;24165  ;HERE FOR REI, AFTER SAVING THE OLD SP, TO LOAD THE NEW ONE,
                                        ;24166  ; AND CAUSE AN AST IF APPROPRIATE.
                                        ;24167
                                        ;24168  =1000   ;-------------------------------;
                                        ;24169  REI.12: Q_ID[KSP],                      ;GET NEW SP
                                        ;24170          ALU_RC[T2],LONG,CLK.UBCC,       ;ALU N TELLS IF WE'RE GOING TO COMP MODE
U 0BD8, 0010,0A38,A1F0,2D10,0010,0EE3   ;24171          MODE.LSS.ASTLVL?,J/REI.13
                                        ;24172
                                        ;24173          ;-------------------------------;
                                        ;24174          Q_ID[ESP],
                                        ;24175          ALU_RC[T2],LONG,CLK.UBCC,       ;ALU N TELLS IF WE'RE GOING TO COMP MODE
U 0BD9, 0010,0A38,A5F0,2D10,0010,0EE3   ;24176          MODE.LSS.ASTLVL?,J/REI.13
                                        ;24177
                                        ;24178          ;-------------------------------;
                                        ;24179          Q_ID[SSP],
                                        ;24180          ALU_RC[T2],LONG,CLK.UBCC,       ;ALU N TELLS IF WE'RE GOING TO COMP MODE
U 0BDA, 0010,0A38,A9F0,2D10,0010,0EE3   ;24181          MODE.LSS.ASTLVL?,J/REI.13
                                        ;24182
                                        ;24183          ;-------------------------------;
                                        ;24184          Q_ID[USP],
                                        ;24185          ALU_RC[T2],LONG,CLK.UBCC,       ;ALU N TELLS IF WE'RE GOING TO COMP MODE
U 0BDB, 0010,0A38,ADF0,2D10,0010,0EE3   ;24186          MODE.LSS.ASTLVL?,J/REI.13
                                        ;24187
                                        ;24188          ;-------------------------------;
U 0BDC, 0000,003C,0180,6004,0000,0062   ;24189          LOAD.IB,PC_PC+1,J/IRD           ;NEW PSL<IS>=1. NO MORE NECESSARY
                                        ;24190  =
                                        ;24191  =011    ;-------------------------------;;NEW MODE .GEQ. ASTLVL
                                        ;24192  REI.13: R[SP]_Q,LONG,                   ;LOAD NEW SP
                                        ;24193          Q_ID[SIR],                      ;READY TO SETUP AST
U 0EE3, 0001,203C,39F0,2EF0,0000,0CF6   ;24194          J/REI.14                        ;GO MAKE AST INTERRUPT REQUEST
                                        ;24195
                                        ;24196          ;-------------------------------;;NEW MODE .LSS. ASTLVL
                                        ;24197          R[SP]_Q,LONG,                   ;LOAD NEW SP
                                        ;24198          LOAD.IB,                        ;NO AST NECESSARY, GO DO AN INSTRUCTION
U 0EE7, 0001,3B3C,0180,62F0,0000,0BD7   ;24199          ALU.N?,J/REI.15                 ;IS IT COMPATABILITY MODE?
                                        ;24200
                                        ;24201          ;-------------------------------;
U 0CF6, 0819,2030,1180,6000,0000,0CF8   ;24202  REI.14: D_Q.OR.K[.4],LOAD.IB             ;INTERRUPT REQUEST AT LEVEL 2
                                        ;24203
                                        ;24204          ;-------------------------------;
                                        ;24205          ID[SIR]_D,                      ;SET IT UP
U 0CF8, 0000,1B3C,3980,3C00,0000,0BD7   ;24206          ALU.N?
                                        ;24207
                                        ;24208  =0111   ;-------------------------------;;ALU N=0 (VAX MODE)
U 0BD7, 0000,003C,0180,F804,4000,0062   ;24209  REI.15: INTRPT.STROBE,PC_PC+1,J/IRD
                                        ;24210
                                        ;24211          ;-------------------------------;;ALU N=1 (COMPATABILITY MODE)
U 0BDF, 0000,003C,0180,F805,4000,0762   ;24212          INTRPT.STROBE,PC_PC+2,J/IRD11
                                        ;24213
                                        ;24214  .LIST           ;Re-enable full listing
```

```
                                        ;24215  .TOC    "QUEUE.MIC"
                                        ;24216  .TOC    "Revision 3.0"
                                        ;24217  ;        R. J. Avarbock, P. R. Guilbault
                                        ;24218
;24219  .NOBIN
;24220  .TOC    "       Revision History"
;24221
;24222  ; 03    Remove absolute jumps.
;24223  ;       Add general WCS region
;24224  ;       Change macro names that deal with conditions codes.
;24225  ; 02    Fix I.L. Q's interface to MM.  MM reg clobbering. (RJA 13-Feb-81)
;24226  ;       Fix interlocked queue's mess up PC on memory mng fault
;24227  ;       Fix INSQTI not checking for tail entry (H+4) being quad-aligned.
;24228  ;       Cleaned up file. (RJA - started 6/30/80)
;24229  ;       Fix REMQHI & REMQTI also not checking some quad-aligned data.
;24230  ;       Fix INSQHI.  It used data from user's queue when it should't.
;24231  ; 01    Delete Q1.MIC and put here. Use .REGION to get into WCS.
;24232  ;        Q1     00      Delete micro word for PLA trap. It is now in PATCH. No change in operation.
;24233  ;                       Start of history
;24234  ;       Add control to enable WCS only listing
;24235  ;       Make fix that was patch to WCS120 permanent
;24236  ; 00    Start of history
;24237
                                        ;24238  .BIN
                                        ;24239  .NOLIST         ;Disable listing of PCS code for quickie assemblies
```

```
                                    ;24240  .TOC    ''      Queue instructions    : INSQUE''
                                    ;24241
                                    ;24242  ;INSert QUEue entry
                                    ;24243  ;HERE TO INSERT AN ENTRY INTO A QUEUE.
                                    ;24244  ; Q CONTAINS THE ADDRESS OF THE ENTRY,
                                    ;24245  ; D CONTAINS THE ADDRESS OF THE PREDECESSOR
                                    ;24246
                                    ;24247  3CF:    ;-------------------------------;
                                    ;24248  INSQUE: RC[TO]_D,                      ;SAVE PREDECESSOR ADDRESS
                                    ;24249          D_Q,                           ;GET ENTRY ADDR INTO D&Q
U 03CF, 0C01,003C,1980,F980,0104,6894 ;24250        FE_K[ZERO]                     ;ARRANGE TO GET PRED ADDR BACK FROM PRB
                                    ;24251  =0
                                    ;24252  INSQ.20:
                                    ;24253          ;0-------------------------------;
                                    ;24254          RC[T1]_D,                      ;SAVE ENTRY ADDRESS
                                    ;24255          VA_D,                          ;PREPARE TO PROBE ENTRY
                                    ;24256          D_D.RIGHT,                     ;READY TO CHECK FOR NEAR PAGE BOUND
U 0894, 0601,003D,1180,F988,0284,6D0D ;24257        SC_K[.4],CALL,J/PRB.W          ;GO PROBE FOR WRITE
                                    ;24258
                                    ;24259          ;1-------------------------------;RETURN HERE IF ENTRY WRITABLE
                                    ;24260          D[LONG]_CACHE.WCHK,            ;GET ADDR OF SUCCESSOR FROM PREDECESSOR
U 0895, 0010,0038,01C0,5108,0000,0CF9 ;24261        Q_RC[T1]                       ;GET ENTRY ADDR INTO Q
                                    ;24262
                                    ;24263          ;-------------------------------;
                                    ;24264          VA_D+K[.4],                    ;GET ADDR OF SUCCESSOR BACK LINK
U 0CF9, 0C19,0014,11E0,F800,0200,0CFA ;24265        D_Q,Q_D                        ;SUCCESSOR ADDR TO Q, ENTRY ADDR TO D
                                    ;24266
                                    ;24267          ;-------------------------------;
U 0CFA, 0000,003C,0180,3000,0000,0CFB ;24268        CACHE_D[LONG]                  ;STORE ENTRY ADDR TO SUCCESSOR BACK LINK
                                    ;24269
                                    ;24270  ;HERE INSQUE HAS VERIFIED THAT ALL REFERENCES WILL COMPLETE, AND
                                    ;24271  ; CAN BEGIN STORING QUEUE LINKAGE INTO THE ENTRY.
                                    ;24272
                                    ;24273          ;-------------------------------;
U 0CFB, 0C01,003C,0180,F800,0200,0CFC ;24274        VA_D,D_Q                       ;ADDR ENTRY
                                    ;24275
                                    ;24276          ;-------------------------------;
                                    ;24277          CACHE_D[LONG],                 ;STORE ADDR OF SUCCESSOR INTO ENTRY
U 0CFC, 0010,0038,01C0,3100,0000,0CFD ;24278        Q_RC[TO]                       ;GET PREDECESSOR ADDRESS
                                    ;24279
                                    ;24280          ;-------------------------------;
                                    ;24281          VA_VA+4,                       ;ADDRESS BACK LINK OF ENTRY
                                    ;24282          D_Q,                           ;READY PREDECESSOR ADDR FOR STORING
U 0CFD, 0C1D,C008,0180,F803,0070,0CFE ;24283        A[U_D-Q-1,SET.CC(INST)         ;COMPARE LINKS FOR CC SETTING
                                    ;24284
                                    ;24285  ; ************************************************
                                    ;24286  ; * Patch no. 064, PCS 0CFD trapped to WCS 118A *
                                    ;24287  ; ************************************************
                                    ;24288
                                    ;24289  QU.PA.64:
                                    ;24290          ;-------------------------------;
                                    ;24291          CACHE_D[LONG],                 ;STORE PREDECESSOR ADDR INTO ENTRY
                                    ;24292          Q_RC[T1],                      ;GET ENTRY ADDR BACK
U 0CFE, 0010,0038,01C0,3108,0000,0D05 ;24293        J7REMQ.6                       ;STORE INTO PREDECESSOR
```

```
                                    ;24294  .TOC    ''      Queue instructions    : REMQUE''
                                    ;24295
                                    ;24296  ;REMove entry from QUEue
                                    ;24297  ;HERE WITH ADDRESS OF ENTRY TO BE REMOVED FROM QUEUE IN D
                                    ;24298
                                    ;24299  284:    ;-------------------------------;
                                    ;24300  REMQUE: RC[T1]_D,                      ;SAVE ENTRY ADDRESS
                                    ;24301          D[LONG]_CACHE,                 ;GET SUCCESSOR ADDRESS FROM ENTRY
U 0284, 0001,003C,1980,4188,0104,6D00 ;24302        FE_K[ZERO]                     ;ARRANGE TO READ BACK RC[TO] IN PRB
                                    ;24303
                                    ;24304  REMQ.30:
                                    ;24305          ;-------------------------------;
                                    ;24306          VA_VA+4,                       ;ADDRESS BACK LINK OF ENTRY
                                    ;24307          RC[TO]_D,                      ;SAVE ADDR OF SUCCESSOR
U 0D00, 0001,003C,C180,3D83,0000,0898 ;24308        ID[TO]_D                       ; SAVE TWICE
                                    ;24309
                                    ;24310  =0      ;-------------------------------;
                                    ;24311          D[LONG]_CACHE,                 ;GET PREDECESSOR ADDRESS
                                    ;24312          SC_K[.2],                      ;SET SC FOR PAGE BOUND CHECK
U 0898, 0000,003D,0980,4000,0084,689C ;24313        CALL,J/REMQ.1                  ;GO CHECK BOTH ADDRESSES
                                    ;24314
                                    ;24315          ;-------------------------------;RETURN HERE IF PRED & SUCC WRITABLE
                                    ;24316          ALU_D-Q-1,                     ;COMPARE PRED & SUCC ADDRESSES
                                    ;24317          SET.CC(INST),                  ;SET CC ACCORDINGLY
                                    ;24318          LC_RC[T1],                     ;GET BACK ENTRY ADDRESS
U 0899, 001D,C008,0180,F908,0070,01D4 ;24319        J/REMQ.2                       ;GO CHECK FOR V SETTING
                                    ;24320
                                    ;24321  =0      ;-------------------------------;
                                    ;24322  REMQ.1: RC[T4]_D,Q_D,                  ;SAVE PREDECESSOR ADDRESS
                                    ;24323          ID[T4]_D,                      ; AS CONVENIENT
                                    ;24324          VA_D,                          ;PREPARE TO PROBE IT
                                    ;24325          D_D.RIGHT,                     ;READY FOR PAGE BOUND TEST
U 089C, 0601,003D,D1E0,3DA0,0200,0D0D ;24326        CALL,J/PRB.W                   ;GO PROBE LONGWORD
                                    ;24327
                                    ;24328          ;-------------------------------;RETURN HERE AFTER PROB OF PRED
                                    ;24329          VA_D+K[.4],                    ;READY TO PROBE SUCCESSOR+4
                                    ;24330          FE&SC_K[.4],                   ; TO SUCCESSOR+7
                                    ;24331          Q_D,D_D.RIGHT,                 ;PREPARE FOR PAGE BOUND CHECK
U 089D, 0619,0014,11E0,F800,0384,6D0D ;24332        J/PRB.W                        ;RETURN ABOVE IF SUCCESSFUL
```

```
                                    ;24333  ;HERE AFTER PROBING BOTH PREDECESSOR AND SUCCESSOR ENTRIES TO VERIFY
                                    ;24334  ; THAT THEY ARE WRITABLE.  THE PSL N, Z, AND C CONDITION CODES ARE SET
                                    ;24335  ; APPROPRIATELY, BUT V IS CLEAR.  WE MUST STILL CALCULATE THE ADDRESS
                                    ;24336  ; AT WHICH TO STORE THE ENTRY ADDRESS, AND SET V IF THE ONLY ITEM IN
                                    ;24337  ; THE QUEUE IS THE QUEUE HEADER ITSELF.
                                    ;24338
                                    ;24339  =10****0;-------------------------------;CALL SITE FOR A-SRC SPEC EVAL
                                    ;24340  REMQ.2: ALU_D.XOR.LC,CLK.UBCC,LONG,     ;IS ENTRY ADDR .EQL. BACK LINK?
U 01D4, 0011,0021,0180,F800,0010,047E ;24341        CAL[,J/ASPC                     ;GO EVALUATE DESTINATION SPECIFIER
                                    ;24342
                                    ;24343  =11****0;-------------------------------;MEMORY OPERAND, ADDR IS IN VA
                                    ;24344          D_RC[T1],                       ;GET THE ADDRESS OF THE ENTRY
                                    ;24345          Q_ID[T4],                       ;AND PRED ADDR WHICH WE'LL WANT SOON
U 01F4, 0810,0038,D1F0,2D08,0000,0D02 ;24346        J/REMQ.3
                                    ;24347
                                    ;24348          ;-------------------------------;REGISTER OPERAND
                                    ;24349          D_RC[T1],                       ;GET ENTRY ADDR TO STORE
U 01F5, 0810,0038,D1F0,2D08,0000,0D01 ;24350        Q_ID[T4]                        ; ALSO PRED ADDR
                                    ;24351
                                    ;24352          ;-------------------------------;
                                    ;24353          R(PRN)_D,                       ;STORE ENTRY ADDR IN DEST OPERAND
                                    ;24354          D_Q,                            ;GET PREDECESSOR ADDRESS TO D
                                    ;24355          Q_ID[T0],                       ;AND SUCCESSOR ADDRESS TO Q
U 0D01, 0C01,013C,C1F0,2CD8,0000,08A0 ;24356        Z?,J/REMQ.4                     ;SHOULD WE SET V?
                                    ;24357
                                    ;24358          ;-------------------------------;
U 0D02, 0000,003C,0180,3000,0000,0D03 ;24359  REMQ.3: CACHE_D[LONG]                  ;STORE ENTRY ADDR IN DESTINATION
                                    ;24360
                                    ;24361          ;-------------------------------;
                                    ;24362          D_Q,                            ;GET PREDECESSOR ADDR INTO D
                                    ;24363          Q_ID[T0],                       ;AND SUCCESSOR ADDRESS TO Q
U 0D03, 0C00,013C,C1F0,2C00,0000,08A0 ;24364        Z?                              ;DOES ENTRY POINT TO ITSELF?
                                    ;24365
                                    ;24366  =0      ;-------------------------------;ALU Z=0 (ENTRY DOES NOT ADDRESS SELF)
                                    ;24367  REMQ.4: VA_Q+K[.4],                     ;ADDRESS OF SUCCESSOR BACK LINK
U 08A0, 0019,2014,1180,F800,0200,0D04 ;24368        J/REMQ.5
                                    ;24369
                                    ;24370          ;-------------------------------;ALU Z=1 (ENTRY POINTS TO ITSELF)
                                    ;24371          SET.V,                          ; SET V TO INDICATE NO ENTRY IN QUEUE
U 08A1, 0019,2014,1180,F800,0220,0D04 ;24372        VA_Q+K[.4]                      ;SETUP ADDR OF SUCCESSOR BACK LINK
                                    ;24373
                                    ;24374          ;-------------------------------;
U 0D04, 0000,003C,0180,3000,0000,0D05 ;24375  REMQ.5: CACHE_D[LONG]                  ;STORE PREDECESSOR ADDR INTO SUCCESSOR+4
                                    ;24376
                                    ;24377          ;-------------------------------;
                                    ;24378  REMQ.6: VA_D,                           ;ADDRESS PREDECESSOR FOREWARD LINK
                                    ;24379          D_Q,                            ;GET SUCCESSOR ADDRESS
U 0D05, 0C01,003C,0180,F800,0200,03FD ;24380        J/STOR.L                        ;GO STORE IT THERE, END INSTRUCTION
```

```
                                        ;24381  .TOC    "       Queue instructions    : INSQHI, INSQTI"
                                        ;24382
                                        ;24383  .FFLIST         ;Re-enable full listing
                                        ;24384  .REGION/<WCSR1L>,<WCSR1H>/<WCSR2L>,<WCSR2H>
                                        ;24385
                                        ;24386  ;INPUTS:        D = ADDRESS OF ENTRY FOR INSERT INSTR.
                                        ;24387  ;               D = ADDRESS OF HEADER FOR REMOVE INSTR
                                        ;24388  ;
                                        ;24389  ;TEMPORARIES FOR INSERT INSTR:
                                        ;24390  ;               RC2 = HEADER-ENTRY
                                        ;24391  ;               RC1 = ENTRY
                                        ;24392  ;               RC3 = ADDRESS OF ENTRY AT HEAD.
                                        ;24393  ;               R15 = HEADER
                                        ;24394  ;
                                        ;24395  ; THE NOTATION FOLLOWS THAT OF THE EXAMPLES IN THE ARCHITECTURE HANDBOOK,
                                        ;24396  ; I.E. THE LETTERS H,A,B,C ARE USED TO DENOTE ENTRIES IN THE QUEUES.
                                        ;24397
                                        ;24398  ;       The following microword is in PATCH.  It is included here
                                        ;24399  ;       as a comment so you can see what actually starts the interlocked
                                        ;24400  ;       queue instructions without having to flip thru reams of paper...
                                        ;24401
                                        ;24402  ;1181:  ;-------------------------------;
                                        ;24403  ;       ALU_D.AND.K[.7],N&Z_ALU.V&C_0,  ;  CLOCK LOW 3 ADDRESS BITS TEST FOR
                                        ;24404  ;       IR1?,J/INSERT.Q                 ;  FOR INSERT OR REMOVE REJOIN Q1.MIC
                                        ;24405
                                        ;24406  =00***10
                                        ;24407  INSERT.Q:
                                        ;24408          ;00***10-----------------------;  BRANCH ON IR<1>
                                        ;24409          SC_K[.1],                       ;  USEFUL LATER FOR SETTING CC'S
                                        ;24410          RC[T1]_D,                       ;  SAVE ENTRY IN RC[T1]
U 109A, 0001,C03D,0580,F988,0084,647E   ;24411          CALL,J7ASPC                     ;  EVALUATE HEADER ADDRESS, H
                                        ;24412
                                        ;24413          ;00***11-----------------------;
U 109B, 0000,1A3C,0180,F800,0000,1300   ;24414          PSL.Z?,J/REMOVE.Q               ;  TEST HEADER ALIGNMENT
                                        ;24415
                                        ;24416  =11****0
                                        ;24417          ;11****0-----------------------;
                                        ;24418          ALU_D.AND.K[.7],                ;  ISOLATE LOW 3 BITS OF HEADER ADDR
                                        ;24419          N_AMX.Z_TST,                    ;  CLOCK Z-BIT ON LOW 3 BITS
U 10FA, 0019,1B34,5D80,F800,0030,108D   ;24420          IR0?,J/INSQI                    ;  HEADS OR TAIL ?
                                        ;24421
                                        ;24422  =11****1
                                        ;24423          ;11****1-----------------------;
U 10FB, 0000,003C,0180,F800,0000,0106   ;24424          J/RSVOPR                        ;  REGISTER HEADER
                                        ;24425  =       ;*** End of constraint...
```

```
                                    ;24426  =1101
                                    ;24427  INSQI:  ;1101----------------------------;  BRANCH ON LOW BIT OF THE OPCODE
                                    ;24428          STATE_K[ZERO],                  ;  CLEAR TAIL-BIT OF STATE
                                    ;24429          VA_Q,RC[T1]_Q,                  ;  VA GETS ENTRY ADDRESS
U 108D, 0001,3A3C,1980,F988,1604,7068;24430          PS[.Z?,J/IQ.0                   ;  HEADER ALIGNED?
                                    ;24431
                                    ;24432          ;1111----------------------------;
                                    ;24433          STATE_K[.4],                    ;  SET TAIL-BIT OF STATE
                                    ;24434          VA_Q,RC[T1]_Q,                  ;  VA GETS ENTRY ADDRESS
U 108F, 0001,3A3C,1180,F988,1604,7068;24435          PS[.Z?,J/IQ.0                   ;  HEADER ALIGNED?
                                    ;24436  =10**
                                    ;24437  IQ.0:   ;10**----------------------------;
U 1068, 0000,003C,0180,F800,0000,0106;24438          J/RSVOPR                        ;  HEADER = ENTRY
                                    ;24439
                                    ;24440          ;11**----------------------------;
                                    ;24441          D[LONG]_CACHE.WCHK,             ;  READ ENTRY WITH WRITE/CHECK
                                    ;24442          ALU_D-Q,RC[T2]_ALU,             ;  STORE HEADER-ENTRY IN RC2
U 106C, 001D,0000,01E0,5190,0010,1440;24443          CLK.UBCC,Q_D                    ;
```

```
                                    ;24444          ;--------------------------------;
                                    ;24445          VA_Q,R[R15]_Q,                   ;  LOAD HEADER IN VA AND R15
U 1440, 0001,213C,0180,FAF8,0200,1258 ;24446        Z?                               ;  TEST HEADER - ENTRY
                                    ;24447
                                    ;24448  =0      ;0--------------------------------;
                                    ;24449          ALU_Q.AND.K[.7],                 ;  ISOLATE LOW 3 BITS OF HEADER
                                    ;24450          N_AMX.Z_TST,                     ;  CLOCK PSL Z-BIT ON THEM
                                    ;24451          D[LONG]_CACHE.LK,                ;  READ HEADER CONTENTS INTERLOCKED
U 1258, 0019,2034,5D80,7000,0030,1442 ;24452        J/IQ.10
                                    ;24453
                                    ;24454          ;1--------------------------------;
U 1259, 0000,003C,0180,F800,0000,0106 ;24455        J/RSVOPR
                                    ;24456
                                    ;24457  IQ.10:  ;--------------------------------;
                                    ;24458          SC_K[.1],                        ;  FOR CONDITION CODE SETTING
                                    ;24459          ALU_D.OR.K[.1],D_ALU,            ;  SET 2.INTERLOCK BIT
                                    ;24460          LAB_R[R15],                      ;  LATCH HEADER ADDRESS
U 1442, 0819,1930,05E0,FA78,0084,72CE ;24461        Q_D,DO?                          ;  BRANCH ON 2. INTERLOCK BIT
                                    ;24462
                                    ;24463  =1110   ;1110-------------------------------;
                                    ;24464          CACHE_D[LONG].LK,                ;  RELEASE 1. INTERLOCK, SET 2.
                                    ;24465          ALU_D.AND.K[.6],N_AMX.Z_TST,     ;  CLOCK PSL-Z BIT ON ALIGNMENT
U 12CE, 0019,0034,D580,3800,0030,13A0 ;24466        J/IQ.3
                                    ;24467
                                    ;24468          ;1111-----------------------------;  RELEASE 1. INTERLOCK, DON'T SET 2
                                    ;24469          ALU_0+K[.1],SC_K[.1],            ;  SC GETS CONDITION CODES
                                    ;24470          N&Z_ALU.V&C_0,                   ;  CLEAR ALL CONDITION CODES
                                    ;24471          CACHE_D[LONG].LK,                ;  RELEASE 2. INTERLOCK
U 12CF, 001B,0014,0580,3800,00D4,745B ;24472        J/IQ.FIN                         ;  SET CONDITION CODES
                                    ;24473  =000
                                    ;24474  IQ.3:   ;000--------------------------------;
                                    ;24475          ALU_Q+LB,D_Q,                    ;  ADD (HEADER) TO HEADER [(A-H) + H]
                                    ;24476          RC[T3]_ALU,VA_VA+4,              ;  STORE IN RC3, VA POINTS TO HEADER+4
                                    ;24477          PSL.CC?,                         ;  TEST ALIGNMENT
U 13A0, 0C0D,3A15,0180,F99B,0000,1078 ;24478        CALL,J/IQ.FPD                    ;  (A-H)+H is Addr of 'HEAD entry'
                                    ;24479
                                    ;24480          ;001--------------------------------;
U 13A1, 0000,003C,0580,FA78,0284,7468 ;24481        SC_K[.1],VA_R[R15],J/Q.FAIL.1    ;  SET C-BIT, VA GETS HEADER-ADDRESS
                                    ;24482
                                    ;24483          ;010--------------------------------;
U 13A2, 0000,003C,0580,FA78,0284,7468 ;24484        SC_K[.1],VA_R[R15],J/Q.FAIL.1    ;  SET C-BIT, VA GETS HEADER-ADDRESS
                                    ;24485
                                    ;24486          ;011--------------------------------;
                                    ;24487          VA_RC[T3],                       ;  VA GETS HDR+(HDR) ['HEAD' entry]
U 13A3, 0C10,0038,01E0,F918,0200,1446 ;24488        D_Q,Q_D,J/IQ.4                   ;  D GETS RESTART ADDRESS
                                    ;24489
                                    ;24490  =111    ;111--------------------------------;
                                    ;24491          ALU_D,N&Z_ALU.V&C_0,             ;  CLOCK ALU ON DIFFERENCE
                                    ;24492          D[LONG]_CACHE,                   ;  READ (HEADER+4)
                                    ;24493          D.NE.0?,                         ;  IS QUEUE EMPTY ?
U 13A7, 0001,0D3C,0180,4000,0050,1361 ;24494        J/IQT.4                          ;  JUMP TO TAIL-FLOW
```

```
                                    ;24495  IQ.4:   ;--------------------------------;
                                    ;24496          ALU_Q,N_AMX.Z_TST,              ;  CLOCK ALU N&Z BITS ON DIFFERENCE
                                    ;24497          VA_VA+4,                        ;  POINT TO BACKWARD POINTER
                                    ;24498          LC_RC[T2],                      ;  LATCH HEADER.MINUS.ENTRY
U 1446, 0001,203C,B580,3D13,0030,1448 ;24499        ID[FPDA]_D                      ;  STORE FAULT ADDRESS IN ID[FPDA]
                                    ;24500
                                    ;24501          ;--------------------------------;
U 1448, 0003,003C,0180,F800,0030,1449 ;24502        ALU_0(A),N_AMX.Z_TST            ;  CLEAR ALU N-BIT
                                    ;24503
                                    ;24504          ;--------------------------------;
U 1449, 0810,0000,0180,F800,0000,144A ;24505        ALU_LA-LC,D_ALU                 ;  H-(H-E) = E [Addr of ENTRY]
                                    ;24506
                                    ;24507          ;--------------------------------;  ** This is (H-(H-E))-(H+(A-H))=E-A
                                    ;24508          ALU_D-LC,D_ALU,                 ;  B-A  (B REFERRED TO AS E ABOVE)
U 144A, 0811,0000,0180,F918,0000,144E ;24509        LC_RC[T3]                       ;  LC now has H-(H-A)  or just A.
                                    ;24510
                                    ;24511          ;--------------------------------;
                                    ;24512          ALU_0-D,Q_ALU,                  ;  GENERATE A-B
U 144E, 001F,2000,01C0,3000,0000,1450 ;24513        CACHE_D[LONG]                   ;  Q GETS B-A FROM ABOVE.
                                    ;24514
                                    ;24515          ;--------------------------------;
                                    ;24516          D_Q,Q_D,                        ;  D GETS B-A
U 1450, 0C10,0038,01E0,F908,0200,1451 ;24517        VA_RC[T1]                       ;  VA GETS ENTRY
                                    ;24518
                                    ;24519          ;--------------------------------;
                                    ;24520          Q_RC[T2],                       ;  Q GETS H-A
U 1451, 0010,0038,01C0,3110,0000,1452 ;24521        CACHE_D[LONG]                   ;  STORE B-A
                                    ;24522
                                    ;24523          ;--------------------------------;
                                    ;24524          VA_VA+4,                        ;  POINT TO FORWARD LINK IN ENTRY
                                    ;24525          D_Q,                            ;  D GETS H-A
U 1452, 0C1F,0000,01C0,F803,0000,1454 ;24526        ALU_0-Q,Q_ALU                   ;  Q GETS A-H
                                    ;24527
                                    ;24528
                                    ;24529  IQ.5:   ;--------------------------------;  HERE FROM INSQTI WITH D=C-H, VA=H+4
                                    ;24530          ALU_0+K[.1],CLK.UBCC,           ;  OR   FROM REMQTI WITH D=B-H, VA=H+4
U 1454, 001B,0014,0580,3000,0010,1458 ;24531        CACHE_D[LONG]                   ;  CLEAR ALU  CC's.   WRITE H-A
                                    ;24532
                                    ;24533  IQ.51:  ;--------------------------------;  ENTER HERE FROM REMQ WITH Q = B-H
U 1458, 0000,003C,0180,FA78,0200,1459 ;24534        VA_R[R15]                       ;  VA GETS H
                                    ;24535
                                    ;24536          ;--------------------------------;
                                    ;24537          D[LONG]_CACHE.LK,               ;  SET HARDWARE INTERLOCK
U 1459, 0000,173C,0180,7000,0000,1079 ;24538        STATE3-0?                       ;  HEAD OR TAIL ?
```

```
                                    ;24539  =0**     ;0**-------------------------------; BRANCH ON HEAD/TAIL BIT OF STATE
U 1079, 0C00,003C,0180,F800,0000,145A ;24540        D_Q,J/IQ.7                        ; HEAD: GET NEW CONTENTS OF HEAD
                                    ;24541
                                    ;24542          ;1**------------------------------;
U 107D, 0819,0024,0580,F800,0000,145A ;24543        D_D.ANDNOT.K[.1],J/IQ.7           ; TAIL: CLEAR 2. INTERLOCK IN HEAD
                                    ;24544
                                    ;24545  IQ.7:   ;-------------------------------;
                                    ;24546          ALU_0(A),N_AMX.Z_TST,             ;  CLEAR ALU N-BIT
                                    ;24547          CACHE_D[LONG].LK,                 ;  RELEASE HARDWARE INTERLOCK
U 145A, 0003,003C,0180,3800,2030,0B2C ;24548        CLR.FPD,J/FINI15
                                    ;24549
                                    ;24550
                                    ;24551  ;ROUTINE TO SET CONDTION CODES FOR Q-INSTRUCTIONS
                                    ;24552  ;ENTER WITH NEW CODE IN SC, PSL IN Q
                                    ;24553
                                    ;24554  IQ.FIN: ;-------------------------------;
U 145B, 0000,003C,3DF0,2C00,0000,145D ;24555        Q_ID[PSL]                         ;  READ PSL
                                    ;24556
                                    ;24557  IQ.FIN.1:
                                    ;24558          ;-------------------------------;
U 145D, 0819,2030,1D80,F800,2000,1460 ;24559        ALU_Q.OR.K[SC],D_ALU,CLR.FPD      ;  CLEAR FIRST PART DONE
                                    ;24560
                                    ;24561          ;------------------- --------;
U 1460, 0000,003C,3D80,3C00,0000,0B2C ;24562        ID[PSL]_D,J/FINI15                ;
                                    ;24563
                                    ;24564
                                    ;24565  ;ROUTINE TO SET FPD OR TAKE RESERVED OPERAND FAULT
                                    ;24566  =10**
                                    ;24567  IQ.FPD: ;10**-------------------------------;  BRANCH ON PSL Z-BIT
U 1078, 0000,003C,0180,FA78,0200,1461 ;24568        VA_R[R15],J/Q.FAIL2               ;  ADDRESSES ARE NOT ALIGNED
                                    ;24569
                                    ;24570          ;11**-------------------------------;
                                    ;24571          Q_ID[USTACK],SET.FPD,             ;  Get fault addr.  Set FPD.
U 107C, 0000,173E,81F0,2C00,2400,0003 ;24572        STATE3-0?,RETURN3                 ;  Return back to tail or head.
                                    ;24573
                                    ;24574  Q.FAIL2:;-------------------------------;
U 1461, 0000,003C,0180,7000,0000,1462 ;24575        D[LONG]_CACHE.LK                  ;  ESTABLISH 1. INTERLOCK
                                    ;24576
                                    ;24577          ;-------------------------------;
U 1462, 0819,0024,0580,F800,0000,146D ;24578        D_D.ANDNOT.K[.1]                  ;  CLEAR 2. INTERLOCK
                                    ;24579
                                    ;24580          ;-------------------------------;  RELEASE 1. INTERLOCK
U 1463, 0000,003C,0180,3800,0000,0106 ;24581        CACHE_D[LONG].LK,J/RSVOPR         ;  RETURN2 TO MICRO-MEM-MGMT
```

```
                                ;24582  ;       Control gets here when Memory management wants to take a fault for
                                ;24583  ;       whatever reason.  We get here as the result of our
                                ;24584  ;       FPD fault packing address.  This routine is unlike other
                                ;24585  ;       packing routines because we access memory.  This causes
                                ;24586  ;       a potential problem, because our access potentially can
                                ;24587  ;       have a TB miss which gives control back to memory management.
                                ;24588  ;       We are supposed to return to MM after packing, leaving it's
                                ;24589  ;       STATE (mm regs) untouched from the original fault.  If we generate
                                ;24590  ;       a TB miss, we destroy some of those registers.  This routine
                                ;24591  ;       must save the ones that it needs upon return in case we do
                                ;24592  ;       generate a TB miss micro-trap.  We will do that first, do all
                                ;24593  ;       of our memory access's restore those registers (we don't know if
                                ;24594  ;       we Trapped away, so always restore them), and then return to MM
                                ;24595  ;       so he can finish up the original fault procedure.
                                ;24596
                                ;24597  ;       VA.REF goes to RC[T1]
                                ;24598  ;       Used to hold addr passed back with exception.  (VMS uses it.)
                                ;24599
                                ;24600  ;       PTE.VA goes to RC[T2]
                                ;24601  ;       Used to hold offending VA until it is decided that the
                                ;24602  ;       instruction should pack up for a fault.  It then holds the
                                ;24603  ;       exception parameters (ACCVIO bit <3>) while the FPD pack routine
                                ;24604  ;       is running.  When control is returned from FPD pack, then this
                                ;24605  ;       bit is checked for PTE not valid or ACCESS VIOLATION!!!
                                ;24606
                                ;24607  Q.FAIL.1:
                                ;24608          ;-------------------------------;
U 1468, 0000,003C,0180,F970,0000,1469   ;24609          LC_RC[VA.REF]                   ;  Save VA.REF.
                                ;24610
                                ;24611          ;-------------------------------;
U 1469, 0010,0038,0180,F988,0000,146A   ;24612          RC[T1]_LC                       ;  Save.
                                ;24613
                                ;24614          ;-------------------------------;
U 146A, 0000,003C,0180,F950,0000,146B   ;24615          LC_RC[PTE.VA]                   ;  Get this one too.
                                ;24616
                                ;24617          ;-------------------------------;
U 146B, 0010,0038,0180,7190,0000,146D   ;24618          RC[T2]_LC,D[LONG]_CACHE.LK      ;  Save it and establish 1. interlock.
                                ;24619
                                ;24620  ;       *****  Above microinstruction would be the one to cause
                                ;24621  ;       *****  The TB miss mentioned above if it would occur at all...
                                ;24622
                                ;24623          ;-------------------------------;
                                ;24624          LC_RC[T2],                      ;  Begin the restoration of MM reg's.
U 146D, 0819,0024,0580,F910,0000,1472   ;24625          D_D.ANDNOT.K[.1]                ;  CLEAR 2. INTERLOCK
                                ;24626
                                ;24627          ;-------------------------------;  RELEASE 1. INTERLOCK
                                ;24628          CACHE_D[LONG].LK,               ;  RETURN TO MICRO-MEM-MGMT
U 1472, 0010,0038,0180,39D0,0000,1473   ;24629          RC[PTE.VA]_LC                   ;  Restore this one first.
                                ;24630
                                ;24631          ;-------------------------------;
U 1473, 0000,003C,0180,F908,0000,1478   ;24632          LC_RC[T1]                       ;  Get the other one ready.
                                ;24633
                                ;24634          ;-------------------------------;
                                ;24635          RC[VA.REF]_LC,                  ;  Restore this one too.
U 1478, 0010,0038,0180,F9F0,2000,0D8F   ;24636          CLR.FPD,J/M.FLT.03              ;
```

```
                                        ;24637  =*01
                                        ;24638  IQT.4:  ;*01----------------------------;  Branch on D NE 0  (Q31=0)
                                        ;24639          STATE_K[ZERO],                  ;  INSERT AT HEAD INSTEAD
U 1361, 0C10,0038,19F8,F918,1604,7446   ;24640          VA_RC[T3],D_Q,Q_0,J/IQ.4        ;  JOIN HEAD INSERT FLOW
                                        ;24641
                                        ;24642          ;*11----------------------------;
                                        ;24643          ALU_D+LB,Q_ALU,VA_ALU,          ;  GET ADDRESS OF PREV.TAIL.ENTRY
U 1363, 0C0D,0014,01C0,F800,2600,1479   ;24644          D_Q,SET.FPD                     ;  D GETS FPDA-ADDRESS
                                        ;24645
                                        ;24646          ;-------------------------------;
U 1479, 0019,2034,5D80,F800,0010,147A   ;24647          ALU_Q.AND.K[.7],CLK.UBCC        ;  MASK LO 3 BITS TO CHK QUAD ALIGN.
                                        ;24648
                                        ;24649          ;-------------------------------;
                                        ;24650          ID[FPDA]_D,D_RC[T1],            ;  STORE FPDA, D GETS C (=NEW ENTRY)
U 147A, 0810,0138,B580,3D08,0000,125C   ;24651          Z?                              ;  Punt if H+4 is not QUAD Aligned.
                                        ;24652
                                        ;24653  =0      ;0----------------------------;  Branch on ALU<Z>
                                        ;24654          VA_R[R15],                      ;  Get address of header.
U 125C, 0000,003C,0180,FA78,2200,1461   ;24655          CLR.FPD,J/Q.FAIL2               ;  Used to clear interlock 2.
                                        ;24656
                                        ;24657          ;1----------------------------;
                                        ;24658          ALU_D-Q,RC[T4]_ALU,D_ALU,       ;  C-B (B=ADDRESS OF PREV.TAIL)
U 125D, 081D,0000,01E0,F9A0,0000,147B   ;24659          Q_D                             ;  Q GETS C
                                        ;24660
                                        ;24661          ;-------------------------------;
                                        ;24662          CACHE_D[LONG],                  ;  WRITE C-B IN B
U 147B, 001F,2000,01C0,3000,0000,1480   ;24663          ALU_0-D,Q_ALU                   ;  Q GETS B-C
                                        ;24664
                                        ;24665          ;-------------------------------;
U 1480, 0010,0038,0180,F908,0200,1481   ;24666          VA_RC[T1]                       ;  VA GETS C
                                        ;24667
                                        ;24668          ;-------------------------------;
U 1481, 0810,0000,0180,F800,0000,1483   ;24669          ALU_LA-LC,D_ALU                 ;  H-C
                                        ;24670
                                        ;24671          ;-------------------------------;
                                        ;24672          CACHE_D[LONG],                  ;  WRITE H-C IN C
U 1483, 0003,003C,0180,3C00,0030,1486   ;24673          ALU_0(A),N_AMX.Z_TST            ;  CLEAR ALU N-BIT
                                        ;24674
                                        ;24675          ;-------------------------------;
                                        ;24676          VA_VA+4,                        ;  VA GETS ENTRY+4
  1486, 0C10,0038,01C0,F908,0000,1488   ;24677          D_Q,Q_RC[T1]                    ;  D GETS B-C, Q GETS C
                                        ;24678
                                        ;24679          ;-------------------------------;
                                        ;24680          CACHE_D[LONG],                  ;  WRITE B-C IN C+4
U 1488, 000D,2000,01C0,3000,0000,1489   ;24681          ALU_Q-LB,Q_ALU                  ;  Q GETS C-H
                                        ;24682
                                        ;24683          ;-------------------------------;
                                        ;24684          ALU_LA+K[.4],VA_ALU,            ;  VA GETS H+4
U 1489, 0C18,0014,1180,F800,0200,1454   ;24685          D_Q,J/IQ.5                      ;  D GETS C-H
```

```
                                        ;24686  .TOC    ""      Queue instructions    : REMQHI, REMQTI""
                                        ;24687
                                        ;24688  ;INPUTS:
                                        ;24689  ;               D = ADDRESS OF HEADER
                                        ;24690  ;TEMPORARIES FOR REMOVE QUEUE INSTR:
                                        ;24691  ;               RC2 = HEADER-ENTRY
                                        ;24692  ;               RC1 = ADDRESS
                                        ;24693  ;               R15 = HEADER
                                        ;24694
                                        ;24695  =00**0*0
                                        ;24696  REMOVE.Q:
                                        ;24697          ;-------------------------------;
U 1300, 0000,003C,0180,F800,0000,0106   ;24698          J/RSVOPR                        ;  HEADER NOT ALIGNED
                                        ;24699
                                        ;24700  =00**1*0;-------------------------------;
                                        ;24701          STATE_K[.14],                   ;  SET REGISTER BIT AND TAIL BIT
                                        ;24702          ALU_K[.14],CLK.UBCC,            ;  CLEAR ALU Z-BIT
U 1304, 0018,0039,2180,F800,1414,647E   ;24703          CAL[,J/ASPC                     ;  EVALUATE ADDRESS
                                        ;24704
                                        ;24705  =11**1*0;-------------------------------;
                                        ;24706  REMQHI: STATE_K[.4],                    ;  CLEAR REGISTER BIT
U 1364, 0001,003C,1180,F988,1604,7490   ;24707          VA_D,RC[T1]_D,J/REMQHI.1        ;  STORE ADDRESS IN RC1
                                        ;24708
                                        ;24709  =11**1*1;-------------------------------;
                                        ;24710          RC[T1]_RLOG.RIGHT,              ;  SAVE REGISTER NUMBER
U 1365, 0040,0038,0180,F988,1C00,1491   ;24711          J/RQ.00                         ;
                                        ;24712  =;END
                                        ;24713
                                        ;24714  REMQHI.1:
                                        ;24715          ;-------------------------------;
                                        ;24716          D[LONG]_CACHE.WCHK,             ;  READ ADDRESS WITH WRITE-CHECK
U 1490, 001D,0000,0180,5000,0010,1491   ;24717          ALU_D-Q,CLK.UBCC                ;  CLOCK HEADER-ADDRESS
                                        ;24718  =;END
                                        ;24719
                                        ;24720  RQ.00:  ;-------------------------------;
                                        ;24721          FE_K[.4],                       ;  FE GETS TAIL BIT
                                        ;24722          VA_Q,R[R15]_Q,Z?,               ;  VA GETS HEADER, TEST DIFFERENCE
U 1491, 0001,213C,1180,FAF8,0304,726C   ;24723          J/RQ.0                          ;
```

```
                                    ;24724  =0      ;0------------------------------;  Branch on ALU<Z>
                                    ;24725  RQ.0:   D[LONG]_CACHE.LK,                ;  READ CACHE INTERLOCKED
U 126C, 0000,1B3C,0180,7000,0000,1109 ;24726        IRO?,J/RQ.1                      ;  TEST OPCODE FOR TAIL OR HEAD
                                    ;24727
                                    ;24728          ;1------------------------------;
U 126D, 0000,003C,0180,F800,0000,0106 ;24729        J/RSVOPR                         ;  HEADER = ADDRESS, RESERVED OPERAND
                                    ;24730  =;END
                                    ;24731
                                    ;24732  =1*01   ;1*01-----------------------------;  Branch on IR<0>  (ALU<Z>=0)
                                    ;24733  RQ.1:   STATE_STATE.ANDNOT.FE,           ;  CLEAR TAIL-BIT OF STATE
                                    ;24734          ALU_D.OR.K[.1],D_ALU,            ;  SET 2. INTERLOCK
                                    ;24735          LAB_R[R15],                      ;  LATCH HEADER
U 1109, 0819,1930,05E0,FA78,1400,52EE ;24736        Q_D,DO?,J/RQ.2                   ;  SAVE (HEADER) IN Q
                                    ;24737
                                    ;24738          ;1*11-----------------------------;
                                    ;24739          ALU_D.OR.K[.1],D_ALU,            ;  SET 2. INTERLOCK
                                    ;24740          LAB_R[R15],                      ;  LATCH HEADER
U 110B, 0819,1930,05E0,FA78,0000,12EE ;24741        Q_D,DO?                          ;  SAVE (HEADER) IN Q
                                    ;24742  =;END
                                    ;24743
                                    ;24744  =1110   ;1110-----------------------------;  Branch on D<0>
                                    ;24745  RQ.2:   CACHE_D[LONG].LK,                ;  RELEASE HARDWARE INTERLOCK, SET 2.
                                    ;24746          ALU_D.AND.K[.6],                 ;
                                    ;24747          N_AMK.Z_TST,                     ;  CLOCK Z-BIT ON BITS <2:1>
U 12EE, 0019,0034,D580,3800,0030,13B0 ;24748        J/RIQ.3                          ;
                                    ;24749
                                    ;24750          ;1111-----------------------------;
                                    ;24751          SC_K[.3],                        ;
                                    ;247             ALU_Q+K[.3],N&Z_ALU.V&C_0,       ;  CLEAR PSL CC
                                    ;247             CACHE_D[LONG].LK,                ;  RELEASE HARDWARE INTERLOCK
U 12EF, 001B,0014,0D80,3800,00D4,745B ;24754        J/IQ.FIN                         ;
                                    ;24755  =;END
                                    ;24756
                                    ;24757  =000    ;000----------------------------;
                                    ;24758  RIQ.3:  ALU_Q+LB,D_Q,                    ;  ADD HEADER TO (HEADER)
                                    ;24759          RC[T3]_ALU,VA_VA+4,              ;  STORE ENTRY IN RC3
                                    ;24760          PSL.CC?,                         ;  TEST PSL Z-BIT
U 13B0, 0C0D,3A15,0180,F99B,0000,1078 ;24761        CALL,J/IQ.FPD                    ;
```

```
                                    ;24762  RQ.FAILURE:
                                    ;24763          ;001-------------------------------;
                                    ;24764          VA_R[R15],                        ;  LOAD HEADER ADDRESS
U 13B1, 0000,003C,0980,FA78,0284,7468 ;24765        SC_K[.2],J/Q.FAIL.1               ;  SET V-BIT OF PSL
                                    ;24766
                                    ;24767          ;010-------------------------------;
                                    ;24768          VA_R[R15],                        ;  LOAD HEADER ADDRESS
U 13B2, 0000,003C,0980,FA78,0284,7468 ;24769        SC_K[.2],J/Q.FAIL.1               ;  SET V-BIT OF PSL
                                    ;24770
                                    ;24771          ;011-------------------------------;
                                    ;24772          VA_RC[T3],                        ;  VA GETS OLD HEAD ENTRY (=A)
                                    ;24773          D_Q,                              ;  D GETS FPDA
U 13B3, 0C10,0D38,01E0,F918,0200,1369 ;24774        Q_D,D.NE.0?,J/RQ.4                ;  Q GETS ENTRY
                                    ;24775
                                    ;24776  =111    ;111-------------------------------;
                                    ;24777          RC[T4]_D,                         ;  STORE (HEADER) IN RC4
                                    ;24778          D[LONG]_CACHE,                    ;  READ BACKWARD POINTER
U 13B7, 0001,003C,0180,41A0,0000,1499 ;24779        J/RQT.4                           ;  JUMP TO TAIL-FLOW
                                    ;24780
                                    ;24781  =01     ;01-------------------------------;
                                    ;24782  RQ.4:   ALU_RC[T1],SC_ALU(EXP),           ;  LOAD REGISTER-POINTER IN SC
                                    ;24783          ID[FPDA]_D,                       ;  LOAD FAULT ADDRESS
U 1369, 0F10,0038,B580,3D08,00A3,1495 ;24784        D_0,SET.V,J/RQ.41                 ;  LOAD 0 EVERYWHERE
                                    ;24785
                                    ;24786          ;11--------------------------------;  Here from TAIL flow if last entry.
                                    ;24787  RQ.400: ALU_RC[T1],SC_ALU(EXP),           ;  LOAD SC WITH REGISTER-POINTER
U 136B, 0010,0038,B580,3D08,0083,1492 ;24788        ID[FPDA]_D                        ;  LOAD FAULT ADDRESS
                                    ;24789
                                    ;24790  =;END
                                    ;24791
                                    ;24792  RQ.40:  ;-------------------------------;
                                    ;24793          ALU_0+K[.1],CLK.UBCC,             ;  CLEAR ALU CONDITION CODES
                                    ;24794          LC_RC[T3],                        ;  LATCH A (=OLD HEAD ENTRY)
U 1492, 001B,0014,0580,4118,0010,1493 ;24795        D[LONG]_CACHE                     ;  READ ENTRY
                                    ;24796
                                    ;24797          ;-------------------------------;
                                    ;24798          ALU_D.AND.K[.7],                  ;  ENTRY (H+(H)) must be quad-aligned.
U 1493, 0019,0034,5D80,F800,0010,1494 ;24799        CLK.UBCC                          ;  CLOCK ALU<Z>
                                    ;24800
                                    ;24801          ;-------------------------------;
                                    ;24802          ALU_D+LC,VA_ALU,                  ;
                                    ;24803          Q_ALU,RC[T5]_ALU,                 ;  RC5 GETS B (=NEW HEAD ENTRY)
U 1494, 0011,0114,01C0,F9A8,0200,127C ;24804        Z?                                ;  If ALU<Z> not set, we lose.
                                    ;24805
                                    ;24806  =0      ;0--------------------------------;  Branch on ALU<Z>
                                    ;24807          VA_R[R15],CLR.FPD,                ;  BAD NEWS - GO RELEASE INTERLOCK
U 127C, 0000,003C,0180,FA78,2200,1461 ;24808        J/Q.FAIL2                         ;  AND DIE RES-OP STYLE.
                                    ;24809
                                    ;24810          ;1--------------------------------;  Entry quad aligned.
                                    ;24811          ALU_LA-Q,D_ALU,                   ;  D GETS H-B
U 127D, 081C,0000,0180,F803,0000,1495 ;24812        VA_VA+4                           ;  POINT TO B+4
```

```
                                    ;24813
                                    ;24814  RQ.41:  Q_RC[T3],                    ;  Q GETS A (ENTRY BEING REMOVED)
                                    ;24815          CACHE_D[LONG],              ;  STORE H-B IN B+4
U 1495, 0010,1638,01C0,3118,0000,128C ;24816        STATE7-4?                   ;
                                    ;24817
                                    ;24818  =0      ;0--------------------------;  Branch on STATE<4>
                                    ;24819          VA_RC[T1],                  ;  VA GETS ADDRESS
                                    ;24820          D_Q,Q_D,                    ;  D GETS A, Q GETS H-B
U 128C, 0C10,0038,01E0,F908,0200,1498 ;24821        J7RQ.5                      ;  GO WRITE A IN ADDRESS
                                    ;24822
                                    ;24823          ;1--------------------------;
U 128D, G001,203C,0180,F8E8,0000,1496 ;24824        ALU_Q,R(SC)_ALU             ;  STORE A IN ADDRESS
                                    ;24825
                                    ;24826  =;END
                                    ;24827          ;--------------------------;
                                    ;24828          N_AMX.Z_TST,                ;  CLOCK Z-BIT ON H-B
U 1496, 0C1F,2000,01C0,F800,0030,1458 ;24829        ALU_0-D,Q_ALU,D_Q,J/IQ.51   ;  Q GETS B-H
                                    ;24830
                                    ;24831  RQ.5:   ;--------------------------;
                                    ;24832          ALU_0-Q,Q_ALU,              ;  Q GETS B-H
                                    ;24833          N_AMX.Z_TST,                ;  CLOCK Z-BIT ON DIFFERENCE
                                    ;24834          CACHE_D[LONG],              ;  WRITE A IN ADDRESS
U 1498, 001F,0000,01C0,3000,0030,1458 ;24835        J/IQ.51                     ;
                                    ;24836
                                    ;24837          ;--------------------------;  Enter here with (H+4) in D
U 1499, 0019,0034,5D80,F800,0010,149C ;24838  RQT.4:  ALU_D.AND.K[.7],CLK.UBCC    ;  See if (H+4) is Quad aligned.
                                    ;24839
                                    ;24840          ;--------------------------;
                                    ;24841          ALU_D.XOR.RC[T4],CLK.UBCC,  ;  TEST FOR LAST ENTRY
U 149C, 0011,0120,0180,F920,0010,1294 ;24842        Z?                          ;  First branch on quad alignment.
                                    ;24843
                                    ;24844  =0      ;0--------------------------;  Branch on ALU<Z>
U 1294, 0000,003C,0180,FA78,2200,1461 ;24845        VA_R[R15],CLR.FPD,J/Q.FAIL2 ;  Not quad aligned.
                                    ;24846
                                    ;24847          ;1--------------------------;  Quad aligned.
                                    ;24848          ALU_D+LB,Q_ALU,VA_ALU,      ;  LOAD C IN VA AND Q
U 1295, 0C0D,0D14,01C0,F800,0200,1379 ;24849        D_Q,D.NE.0?                 ;  D GETS FPDA
                                    ;24850
                                    ;24851  =01     ;01-------------------------;  Br on (D.NE.0)  I.E  empty Queue
                                    ;24852          STATE_STATE.ANDNOT.K[.4],   ;  CLEAR TAIL BIT
U 1379, 0010,0038,1180,F918,1604,5369 ;24853        VA_RC[T3],J/RQ.4            ;  USE HEAD-FLOW FOR SINGLE ENTRY QUEUE
                                    ;24854
                                    ;24855          ;11-------------------------;
                                    ;24856          ID[FPDA]_D,                 ;  LOAD FAULT ADDRESS
U 137B, 0000,013C,B580,3C03,0000,12B8 ;24857        VA_VA+4,Z?                  ;  POINT TO C+4,  TEST FOR LAST ENTRY
                                    ;24858
                                    ;24859  =0      ;0--------------------------;  MORE THAN ONE ENTRY LEFT (BR ALUz)
                                    ;24860          ALU_0+K[.1],CLK.UBCC,       ;  CLEAR ALU CC
U 12B8, 001B,0014,0580,4000,0010,149D ;24861        D[LONG]_CACHE,J/RQT.40      ;  READ B-C
                                    ;24862
                                    ;24863          ;1--------------------------;  LAST ENTRY IN QUEUE
                                    ;24864          STATE_STATE.ANDNOT.K[.4],   ;  CLEAR TAIL BIT
U 12B9, 0010,0038,1180,F918,1604,5368 ;24865        VA_RC[T3],J/RQ.400          ;  USE HEAD-FLOW IN THIS CASE
                                    ;24866  =;END
```

```
                                ;24867  RQT.40: ;-------------------------------;  Here with D= (H+(H+4)+4)
U 149D, 0019,0034,5D80,F800,0010,149E   ;24868          ALU_D.AND.K[.7],CLK.UBCC       ;  Test (H+(H+4)+4) quad alignment.
                                ;24869                                         ;  This test is on TAIL entry back ptr.
                                ;24870
                                ;24871          ;-------------------------------;
                                ;24872          ALU_D+Q,VA_ALU,RC[T5]_ALU,     ;  STORE B.  D GETS B AS WELL
U 149E, 081D,0114,0180,F9A8,0200,12BC   ;24873          D_ALU,Z?                       ;  Branch on quad alignment.
                                ;24874
                                ;24875  =0      ;0-------------------------------;  Branching on ALU<Z>
U 12BC, 0000,003C,0180,FA78,2200,1461   ;24876          VA_R[R15],CLR.FPD,J/Q.FAIL2    ;  Unaligned data - res op.
                                ;24877
                                ;24878          ;1-------------------------------;  Quad aligned.
U 12BD, 081C,2000,0180,F800,0000,14A0   ;24879          ALU_LA-D,D_ALU                 ;  D GETS H-B
                                ;24880
                                ;24881          ;-------------------------------;
                                ;24882          CACHE_D[LONG],                 ;  WRITE H-B IN B
                                ;24883          ALU_RC[T1],SC_ALU(EXP),        ;
U 14A0, 0010,1638,0180,3108,0083,12C0   ;24884          STATE7-4?                      ;
                                ;24885
                                ;24886  =0      ;0-------------------------------;  BRANCH ON STATE<4>
                                ;24887          VA_RC[T1],                     ;  VA GETS ADDRESS
                                ;24888          D_J,Q_D,                       ;  D GETS C, Q GETS H-B
U 12C0, 0C10,0038,01E0,F908,0200,14A2   ;24889          J7RQT.5                        ;
                                ;24890
                                ;24891          ;1-------------------------------;
U 12C1, 0001,203C,0180,F8E8,0000,14A1   ;24892          ALU_Q,R(SC)_ALU                ;
                                ;24893
                                ;24894          ;-------------------------------;
                                ;24895          N_AMX.Z_TST,                   ;  CLOCK Z-BIT
U 14A1, 001F,2000,01C0,F800,0030,14A3   ;24896          ALU_0-D,Q_ALU,J/RQT.6          ;  Q GETS B-H
                                ;24897
                                ;24898  RQT.5:  ;-------------------------------;
                                ;24899          N_AMX.Z_TST,                   ;
                                ;24900          CACHE_D[LONG],                 ;  WRITE C IN ADDRESS
U 14A2, 001F,0000,01C0,3000,0030,14A3   ;24901          ALU_0-Q,Q_ALU                  ;  Q GETS B-H
                                ;24902
                                ;24903  RQT.6:  ;-------------------------------;
                                ;24904          ALU_LA+K[.4],VA_ALU,           ;  GENERATE H+4
U 14A3, 0C18,0014,1180,F800,0200,1454   ;24905          D_Q,J/IQ.5                     ;  NOW GET B-H TO D AND GOTO
                                ;24906                                         ;  WRITE IT INTO HEAD+4
                                ;24907
                                ;24908  .REGION/0000,0FFF                      ;Lower 4k for PCS
```

```
                                        ;24909  .TOC    "PROBE.MIC"
                                        ;24910  .TOC    "Revision 1.1"
                                        ;24911  ;        P. R. Guilbault
                                        ;24912
;24913  .NOBIN
;24914  .TOC    ""      Revision History"
;24915
;24916  ; 01    Add patch no. 091 to fix PROBEx C bit problem.
;24917  ;       Start of history
;24918
                                        ;24919  .BIN
                                        ;24920  .NOLIST          ;Disable listing of PCS code for quickie assemblies
```

```
                                        ;24921  .TOC     "       Probe instruction     : PROBE"
                                        ;24922
                                        ;24923  ;PROBE - Test address accessibility
                                        ;24924  ;HERE FOR PROBEW AND PROBER INSTRUCTIONS, WITH THE MODE OPERAND IN Q,
                                        ;24925  ; LENGTH IN D.
                                        ;24926
                                        ;24927  3CE:    ;-------------------------------;
                                        ;24928  PROBE:  RC[T0]_D.OXT[WORD],              ;SAVE LENGTH
U 03CE, 0003,403C,F580,F980,0104,6145   ;24929          FE_K[.A]
                                        ;24930
                                        ;24931  =10****1;-------------------------------;
                                        ;24932          RC[T1]_Q.AND.K[.3],SC_ALU,      ;SAVE MODE
U 0145, 0019,2035,0D80,F988,0082,047E   ;24933          CALL,J7ASPC                     ;GO GET BASE OPERAND
                                        ;24934
                                        ;24935          ;-------------------------------;
                                        ;24936          RC[T2]_D,                       ;SAVE BASE ADDRESS
                                        ;24937          Q_ID[PSL],                      ;GET CURRENT PSL
U 0165, 0001,003C,3DF0,2D90,0181,0D06   ;24938          FE_SC,SC_FE                     ;MODE OPERAND TO FE, 10 TO SC
                                        ;24939
                                        ;24940          ;-------------------------------;
                                        ;24941          D_DAL.SC,                       ;PREVIOUS MODE TO D1-0
U 0D06, 0D19,2024,6180,F998,0000,0D08   ;24942          RC[T3]_Q.ANDNOT.K[.F]           ;SAVE PSL EXCEPT FOR CONDITION CODES
                                        ;24943
                                        ;24944  ; ************************************************
                                        ;24945  ; * Patch no. 091, PCS 0D06 Trapped to WCS 119B *
                                        ;24946  ; ************************************************
                                        ;24947
                                        ;24948  PROB.X: ;-------------------------------;
                                        ;24949          D_D.AND.K[.3],SC_ALU,           ;PREVIOUS MODE BITS TO D, SC
U 0D08, 0819,0034,0D80,F900,0082,0D09   ;24950          LC_RC[T0]                       ;LOAD LENGTH TO LATCHES
                                        ;24951
                                        ;24952          ;-------------------------------;
                                        ;24953          Q_0-LC,                         ;NEGATIVE OF LENGTH TO Q
                                        ;24954          C[R.SD&SS,                      ;SS=0 IMPLIES THIS IS FIRST PROBE
U 0D09, 0013,0000,01C7,F800,0010,AA40   ;24955          EALU_SC-FE,CLK.UBCC             ;COMPARE PREV MODE TO MODE OPERAND
                                        ;24956
                                        ;24957  =00     ;-------------------------------;
                                        ;24958  PROB.1: RC[T0]_NOT.Q,                   ;SAVE LENGTH -1 IN RC TEMP
                                        ;24959          D_D.SWAP,                       ;MOVE GREATER MODE TO 27-24
                                        ;24960          SC_K[.1B],                      ;SETUP BIT # OF FPD FLAG
U 0A40, 0B01,3229,ED80,F980,0084,6666   ;24961          EALU.N?,CALL,J/PROB.2           ;GO INIT FPD RETURN ADDRESS
                                        ;24962
                                        ;24963          ;-------------------------------;RETURN HERE IF ACCESS WILL FAULT
                                        ;24964          D_RC[PTE.VA],                   ;GET FAULT PARAMETER FLAGS
                                        ;24965          SC_ALU,                         ; COPY INTO SC, TOO
                                        ;24966          FE_K[.FFF5],                    ;SETUP MASK IN FE FOR FLAGS
U 0A41, 0810,0038,E180,F950,0186,6D0A   ;24967          J/PROB.7
                                        ;24968
                                        ;24969          ;-------------------------------;RETURN HERE IF ACCESS WILL FAULT
                                        ;24970          D_RC[PTE.VA],                   ;GET FAULT PARAMETER FLAGS
                                        ;24971          SC_ALU,                         ; COPY INTO SC, TOO
                                        ;24972          FE_K[.FFF5],                    ;SETUP MASK IN FE FOR FLAGS
U 0A42, 0810,0038,E180,F950,0186,6D0A   ;24973          J/PROB.7
                                        ;24974
                                        ;24975  =;END OF FPD RETURNS
```

```
                                ;24976  ;HERE WHEN PROBE REFERENCE FAULTS
                                ;24977
                                ;24978          ;-------------------------------;
                                ;24979  PROB.7: D_D.AND.K[.8].RIGHT,              ;D=4 IF ACCESS VIOLATION, ELSE 0
                                ;24980          SC_SC.ANDNOT.FE,                ;ISOLATE ACCESS & PTE FETCH FLAGS
U 0D0A, 0859,0034,0180,F800,0090,4D0B   ;24981          CLR.UBCC                        ;CLEAR EALU N BIT
                                ;24982
                                ;24983          ;-------------------------------;
                                ;24984          D_D.OR.RC[T3],                  ;GET BACK PSL, SETTING Z IF ACCESS VIOL
U 0D0B, 0811,0030,0980,F918,0084,AA48   ;24985          SC_SC-K[.2]                     ;SC=0 IF NOT-VALID FAULT ON PTE FETCH
                                ;24986
                                ;24987  =00     ;-------------------------------;CALL SITE FOR PROBE OR BACKUP.RGS
                                ;24988          RC[T3]_D,                       ;UPDATE SAVED PSL WITH Z IF ACCESS VIOL
U 0A48, 0001,123D,0180,F998,0000,0EEC   ;24989          EALU?,CALL,J/PROB.8             ;FAULT IF SC.EQL.0, PROBE SECOND IF SS=0
                                ;24990
                                ;24991          ;-------------------------------;RETURN HERE FROM PROBE
                                ;24992          ID[PSL]_D,                      ;RESTORE ORIGINAL PSL
U 0A49, 0000,003C,3D80,3C00,0000,008E   ;24993          J/NOP
                                ;24994
                                ;24995          ;-------------------------------;RETURN HERE AFTER REGISTERS BACKED-UP
U 0A4A, 0000,003C,0180,F800,0000,0EB8   ;24996          J/FPD.RTN
                                ;24997  =;END OF CALL CONSTRAINT
                                ;24998
                                ;24999  =100    ;-------------------------------;SC .EQL. 0, TAKE A FAULT
                                ;25000  PROB.8: ID[PSL]_D,                      ;RESTORE PSL
U 0EEC, 0000,153C,3D80,3C00,0000,0DA7   ;25001          RLOG.EMPTY?,J/BAKUP.RGS
                                ;25002
                                ;25003          ;-------------------------------;
                                ;25004          ID[PSL]_D,                      ;RESTORE PSL
U 0EED, 0000,153C,3D80,3C00,0000,0DA7   ;25005          RLOG.EMPTY?,J/BAKUP.RGS
                                ;25006
                                ;25007          ;-------------------------------;SC.NEQ.0, SS=0 -- MUST PROBE OTHER PART
                                ;25008          D_RC[T0],                       ;GET LENGTH -1 TO D
                                ;25009          Q_ID[Q.SV],                     ;GET BASE ADDRESS TO Q
U 0EEE, 0810,0038,BDF0,2D00,0000,08A9   ;25010          J/PROB.4A                       ;PROBE SECOND PART
                                ;25011
                                ;25012          ;-------------------------------;SC.NEQ.0, SS=1 -- REPORT ON PROBE
                                ;25013          ID[PSL]_D,                      ;RESTORE PSL
U 0EEF, 0000,003C,3D80,3C00,0000,008E   ;25014          J/NOP
```

```
                                ;25015  ;HERE ON PROBE TO SETUP TEMPORARY PSL AND TEST ADDRESSES
                                ;25016
                                ;25017  =011*   ;-------------------------------;EALU N=0 (D CONTAINS GREATER MODE)
                                ;25018  PROB.2: D_D.CRNOT.MASK,                 ;SET FPD IN TEMPORARY PSL
                                ;25019          Q_ID[USTACK],                   ;GET RETURN ADDRESS FOR FAULT
U 0666, 0801,001C,81F0,2C00,0000,08A4   ;25020          J/PROB.3                ;GO SET THEM UP
                                ;25021
                                ;25022          ;-------------------------------;EALU N=1 (MODE OPERAND .GTR. PREV MODE)
                                ;25023          D_RC[T1],                       ;GET MODE OP FOR USE AS PROBE MODE
                                ;25024          EALU_K[ZERO],CLK.UBCC,          ;CLEAR EALU N SO WE DON'T LOOP HERE
U 066E, 0810,0038,1980,F908,0014,6A40   ;25025          J/PROB.1
                                ;25026
                                ;25027  =0      ;-------------------------------;CALL SITE FOR PROBE SUBROUTINE
                                ;25028  PROB.3: ID[PSL]_D,                      ;SETUP TEMPORARY PSL
                                ;25029          D_Q,                            ;GET FPD FAULT HANDLER ADDRESS
                                ;25030          Q_RC[T2],                       ;GET BASE ADDRESS FOR PROBE SUBROUTINE
                                ;25031          FE_0(A),                        ;ARRANGE TO GET RC[T0] IN Q ON RETURN
U 08A4, 0C13,0039,3DC0,3D10,0108,68A8   ;25032          CALL,J/PROB.4           ;GO DO THE TEST
                                ;25033
                                ;25034          ;-------------------------------;ACCESS IS OK
                                ;25035  PROB.6: ID[PSL]_D,                      ;RESTORE ORIGINAL PSL, CC CLEARED
U 08A5, 0000,003C,3D80,3C00,0000,008E   ;25036          J/NOP                   ;THEN GO TO NEXT INSTRUCTION
                                ;25037
                                ;25038  =0      ;-------------------------------;CALL SITE FOR BASE PROBE
                                ;25039  PROB.4: ID[FPDA]_D,                     ;ESTABLISH FPD FAULT RETURN ADDRESS
                                ;25040          SD_NOT.SD,                      ;SET SD TO BECOME 1ST HALF DONE FLAG
U 08A8, 0000,1B3D,B583,3C00,0000,0BED   ;25041          IR0?,CALL,J/PROB.5      ;TEST FOR READ/WRITE ACCESSIBILITY
                                ;25042
                                ;25043          ;-------------------------------;BASE IS ACCESSIBLE, HOW ABOUT BASE+LEN
                                ;25044  PROB.4A:VA_D+Q,                         ;SETUP BASE + LENGTH
                                ;25045          SS_SD,                          ;SET SS=1 TO FLAG 1ST PROBE DONE
                                ;25046          FE_K[.3],                       ;ARRANGE FOR RC[T3] (PSL) ON RETURN
U 08A9, 001D,1B14,0D82,F800,0304,6BED   ;25047          IR0?,J/PROB.5           ;TEST FAR EXTENT
                                ;25048
                                ;25049  =1101   ;-------------------------------;IR0=0 (PROBER)
                                ;25050  PROB.5: TEST.TB.RCHK,                   ;PROBE READ ACCESSABILITY
                                ;25051          ALU_0+K[ZERO]+1,CLK.UBCC,       ;CLEAR ALU C31
U 0BED, 001B,0010,1980,0000,0010,0D0C   ;25052          J/PRB.R1
                                ;25053
                                ;25054          ;-------------------------------;IR0=1 (PROBEW)
                                ;25055          TEST.TB.WCHK,                   ;PROBE WRITE ACCESSABILITY
                                ;25056          ALU_0+K[ZERO]+1,CLK.UBCC,       ;CLEAR ALU C31
U 0BEF, 001B,0010,1980,1000,0010,0D0E   ;25057          J/PRB.W1
```

```
                                    ;25058  ;HERE IS READ PROBE SUBROUTINE
                                    ;25059  ;ENTRY CONDITIONS:
                                    ;25060  ; VA=   ADDRESS TO BE PROBED
                                    ;25061  ; FE=   REGISTER NUMBER OF RC REGISTER TO BE IN D & VA ON RETURN
                                    ;25062
                                    ;25063          ;-------------------------------;
                                    ;25064  PRB.R1: FE_SC,SC_FE,                    ;GET RC ADDRESS READY
U 0D0C, 0000,1D3C,0180,F800,0181,00D9 ;25065        TB.TEST?                        ;IS ADDRESS ACCESSIBLE?
                                    ;25066
                                    ;25067  =011*01 ;-------------------------------;TB HIT AND ACCESS ALLOWED
                                    ;25068  PRB.R2: D_RC(SC),VA_ALU,                ;GET OUT SELECTED TEMP
                                    ;25069          SC_FE,FE_SC,                    ;RESTORE ORIGINAL SC AND FE
U 00D9, 0810,003A,0180,F830,0381,0001 ;25070        RETURN1                         ;GO TO CALLER WITH OK RETURN
                                    ;25071
                                    ;25072          ;-------------------------------;TB MISS OR ACCESS VIOLATION
                                    ;25073          RC[PC.SV]_PC,PC_VA,             ;SAVE PC WHILE FILLING TB
                                    ;25074          ID[D.SV]_D,D_Q,                 ; SAVE D & Q, TOO
                                    ;25075          SET.NEST.ERR,                   ;NOTE THAT PC NOT IN PC
U 00DB, 0C14,0039,B980,3DE1,2C00,0D10 ;25076        CALL,J/TBF.R                    ;GO FILL TB
                                    ;25077
                                    ;25078  =111*11 ;-------------------------------;RETURN HERE WHEN TB FILLED
                                    ;25079          D_Q,Q_ID[Q.SV],                 ;RESTORE REGISTERS USED
U 00FB, 0C00,003C,BDF0,2C00,0000,00D9 ;25080        J7PRB.R2
```

```
                                ;25081  ;HERE IS WRITE PROBE SUBROUTINE
                                ;25082  ;ENTRY CONDITIONS:
                                ;25083  ; Q =   ADDRESS TO BE PROBED
                                ;25084  ; VA=   ADDRESS TO BE PROBED
                                ;25085  ; D =   ADDRESS/2
                                ;25086  ; SC=   2, TO PROBE A LONGWORD, OR 4, TO PROBE A QUADWORD
                                ;25087  ; FE=   RC ADDRESS OF REGISTER TO BE LOADED INTO D & VA ON RETURN
                                ;25088
                                ;25089  ;RETURN1 IF ADDRESS WRITABLE, WITH:
                                ;25090  ; D =   RC REGISTER SELECTED BY FE
                                ;25091  ; VA=   RC REGISTER SELECTED BY FE
                                ;25092  ; SC=   ENTRY SC OR'D WITH 3
                                ;25093  ;OTHER REGISTERS UNCHANGED
                                ;25094  ;EXIT TO MEM MGMT FAULT IF ADDRESS NOT WRITABLE
                                ;25095
                                ;25096          ;-------------------------------;
                                ;25097  PRB.W:   ALU_D.SXT[BYTE]+K[SC],CLK.UBCC, ;CHECK FOR NEAR PAGE BOUNDARY
U 0D0D, 001A,8014,1D80,1000,0010,0D0E   ;25098          TEST.TB.WCHK                    ;IS THIS ADDRESS WRITABLE
                                ;25099
                                ;25100          ;-------------------------------;
                                ;25101  PRB.W1:  SC_SC.OR.K[.3],                 ;CONVERT 2 TO 3, OR 4 TO 7
U 0D0E, 0000,1D3C,0D80,F800,0084,2119   ;25102          TB.TEST?                        ;IS ADDRESS WRITABLE?
                                ;25103
                                ;25104  =011*01 ;-------------------------------;TB HIT AND ACCESS ALLOWED
                                ;25105  PRB.W2:  VA_Q+K[SC],                    ;GET ADDRESS OF OTHER END OF DATUM
                                ;25106          SC_FE,FE_SC,                    ;SAVE SC, LOAD IT WITH RC ADDRESS
U 0119, 0019,2314,1D80,F800,0381,085C   ;25107          C3T?,J/PRB.W3                   ;DOES DATUM CROSS PAGE BOUND?
                                ;25108
                                ;25109          ;-------------------------------;TB MISS OR ACCESS VIOLATION
                                ;25110          RC[PC.SV]_PC,PC_VA,             ;SAVE PC WHILE FILLING TB
                                ;25111          ID[D.SV]_D,D_Q,                 ; SAVE D&Q, TOO
                                ;25112          SET.NEST.ERR,                   ;NOTE THAT PC IS SAVED
U 011B, 0C14,0039,B980,3DE1,2C00,0D12   ;25113          CALL,J/TBF.W                    ;GO FILL TB
                                ;25114
                                ;25115  =111*11 ;-------------------------------;RETURN HERE AFTER FILLING TB
                                ;25116          D_Q,Q_ID[Q.SV],                 ;RESTORE REGISTERS USED BY TBFILL RTN
U 013B, 0C00,003C,BDF0,2C00,0000,0119   ;25117          J/PRB.W2                        ;PROCEED, ACCESS IS VALID
                                ;25118
                                ;25119  =0*     ;-------------------------------;ALU C31=0 (NOT PAGE BOUNDARY)
                                ;25120  PRB.W3:  D_RC(SC),VA_ALU,               ;GET SELECTED TEMP FOR CALLER'S USE
                                ;25121          SC_FE,FE_SC,                    ;RESTORE SC&FE
U 085C, 0810,003A,0180,F830,C381,0001   ;25122          RETURN1                         ;BACK TO CALLER
                                ;25123
                                ;25124          ;-------------------------------;ALU C31=1 (DATA CROSSES PAGE BOUNDARY)
                                ;25125          ALU_0+K[ZERO]+1,CLK.UBCC,       ;CLEAR C31 SO WE DON'T COME BACK HERE
                                ;25126          SC_FE,FE_SC,                    ;RESTORE SC&FE
                                ;25127          TEST.TB.WCHK,                   ;TEST OTHER SIDE OF PAGE BOUNDARY
U 085E, 001B,0010,1980,1000,0191,0D0E   ;25128          J/PRB.W1                        ;LOOP BACK TO COMPLETE THE TEST
```

```
                                    ;25129  ;SUBROUTINE TO REFILL TB FOR READ PROBE
                                    ;25130
                                    ;25131          ;-------------------------------;
U 0D10, 0000,003C,BD80,3C00,0000,0A50   ;25132  TBF.R:  ID[Q.SV]_D                       ;SAVE Q REGISTER
                                    ;25133
                                    ;25134  ; ************************************************
                                    ;25135  ; * Patch no. 005, PCS 0D10 trapped to WCS 1144 *
                                    ;25136  ; ************************************************
                                    ;25137
                                    ;25138  =00     ;00-------------------------------; AND SC
                                    ;25139          RC[SC.SV]_K[SC],
U 0A50, 0018,1C39,1D80,F9E8,0000,023C   ;25140          CALL,PSL.MODE?,J/GET.PTE
                                    ;25141
                                    ;25142  TBF.R2: ;01-------------------------------;SUCCESSFUL RETURN
                                    ;25143          SC_RC[SC.SV],                   ;RESTORE SAVED SC
                                    ;25144          TEST.TB.RCHK,                   ;CHECK FOR ACCESS VIOLATION
U 0A51, 0010,0038,0180,0168,0082,0D11   ;25145          J/TBF.R1
                                    ;25146
                                    ;25147          ;10-------------------------------;FAULT RETURN
                                    ;25148          Q_D.AND.K[.10].RIGHT,           ;SET ACCESS VIOLATION IF APPROPRIATE
U 0A52, 0059,0034,65C0,F800,0000,0EA8   ;25149          J7M.FLT.B                       ;GO TAKE THE FAULT
                                    ;25150
                                    ;25151  =;END OF SUBROUTINE CONSTRAINT
                                    ;25152
                                    ;25153          ;-------------------------------;
U 0D11, 0000,1D3C,0180,F800,0000,0039   ;25154  TBF.R1: TB.TEST?                        ;IS THERE AN ACCESS VIOLATION?
                                    ;25155
                                    ;25156  =!1*01  ;-------------------------------;ACCESS IS ALLOWED
                                    ;25157          PC&VA_RC[PC.SV],                ;RESTORE PC
                                    ;25158          Q_ID[D.SV],                     ; GET SAVED D REG
U 0039, 0010,003A,B9F0,2D61,2A00,0020   ;25159          C[R.NEST.ERR,RETURN20           ;GO BACK TO PROBE POINT
                                    ;25160
                                    ;25161          ;-------------------------------;ACCESS VIOLATION!
                                    ;25162          RC[PTE.VA]_K[.8],               ;SET CODE ACCORDINGLY
                                    ;25163          D_K[.8],
U 003B, 0818,0038,0180,F9D0,0000,0EA9   ;25164          J7M.FLT.PRB                     ;
```

```
                                        ;25165  ;HERE IS SUBROUTINE TO FILL TRANSLATION BUFFER FOR A WRITE PROBE
                                        ;25166
                                        ;25167          ;-------------------------------;
U 0D12, 0000,003C,BD80,3C00,0000,0A68   ;25168  TBF.W:  ID[Q.SV]_D                       ;SAVE Q REGISTER
                                        ;25169
                                        ;25170  ; ************************************************
                                        ;25171  ; * Patch no. 006, PCS 0D12 trapped to WCS 1145 *
                                        ;25172  ; ************************************************
                                        ;25173
                                        ;25174  =00     ;-------------------------------;
                                        ;25175          RC[SC.SV]_K[SC],                ; AND SC FOR REFILL ROUTINE
U 0A68, 0018,1C39,1D80,F9E8,0000,023C   ;25176          CALL,PSL.MODE?,J/GET.PTE
                                        ;25177
                                        ;25178          ;-------------------------------;SUCCESS ON GETPTE. UPDATE M.BIT
                                        ;25179          TEST.TB.WCHK,                   ;DOES THIS PTE ALLOW WRITE?
U 0A69, 0000,003D,0180,1000,0000,0D6E   ;25180          CALL,J/TEST.PTE.W
                                        ;25181
                                        ;25182          ;-------------------------------;
                                        ;25183          D_D.OR.K[.4],                   ;FLAG MEM MGMT FAULT HAD WRITE INTENT
U 0A6A, 0819,0030,1180,F800,0000,0EA3   ;25184          J7M.FLT.A                       ;GO TAKE THE FAULT
                                        ;25185
                                        ;25186          ;-------------------------------;SUCCESS RETURN FROM SETTING M.BIT
                                        ;25187  TBF.X:  PC&VA_RC[PC.SV],                ;RESTORE PC
                                        ;25188          Q_ID[D.SV],                     ; GET SAVED D REG
U 0A6B, 0010,003A,B9F0,2D61,2A00,0020   ;25189          CLR.NEST.ERR,RETURN20           ;GO BACK TO PROBE POINT
                                        ;25190
                                        ;25191  .LIST           ;Re-enable full listing
```

```
                                        ;25192  .TOC    'MOVPR.MIC''
                                        ;25193  .TOC    'Revision 1.5''
                                        ;25194  ;        P. R. Guilbault
                                        ;25195
;25196  .NOBIN
;25197  .TOC    ''      Revision History''
;25198
;25199  ; 01    Remove absolute jumps.
;25200  ;       Comment patch no. 096 to add PRN ^X3F for TB miss check.
;25201  ; 00    Start of history
;25202
                                        ;25203  .BIN
                                        ;25204  .NOLIST          ;Disable listing of PCS code for quickie assemblies
```

```
                                        ;25205  .TOC    ""       SVP/LDPCTX, MF/MTPR   : SVPCTX - Save Process Context""
                                        ;25206
                                        ;25207  ;HERE FROM A-FORK TO SAVE REGISTERS IN PCB
                                        ;25208
                                        ;25209  088:    ;-------------------------------;
                                        ;25210  SVPCTX: VA_R[SP],                       ;READY TO GET PC & PSL FROM STACK
U 0088, 0000,1C3D,0180,FA70,0200,003E   ;25211          PSL.MODE?,CALL,J/KERNEL.CHK     ;AFTER VALIDATING THIS USER
                                        ;25212
                                        ;25213  0A8:    ;-------------------------------;THIS ONE'S OK
                                        ;25214          D[LONG]_CACHE,                  ;GET SAVED PC FROM STACK
                                        ;25215          R[SP]_LA+K[.8].RLOG,            ;UPDATE SP ACCORDING TO 2 POP'S
U 00A8, 0018,0018,01C0,42F0,0000,0D14   ;25216          Q_ALU                           ;HOLD UPDATED SP VALUE
                                        ;25217
                                        ;25218          ;-------------------------------;
                                        ;25219          RC[TO]_D,                       ;SAVE STACKED PC
U 0D14, 0001,003C,0180,F983,0000,0D15   ;25220          VA_VA+4                         ;ADDRESS THE STACKED PSL
                                        ;25221
                                        ;25222          ;-------------------------------;
                                        ;25223          D[LONG]_CACHE,                  ;GET STACKED PSL
U 0D15, 0000,003C,E580,4000,0084,6D16   ;25224          SC_K[.1A]                       ;GET BIT # (26) OF PSL<IS> BIT
```

```
                                ;25225  ;HERE HAVING OBTAINED PC AND PSL FROM STACK.  DO WE NEED TO SWITCH
                                ;25226  ; TO THE INTERRUPT STACK?
                                ;25227
                                ;25228          ;-------------------------------;
                                ;25229          RC[T1]_D,                       ;SAVE STACKED PSL
                                ;25230          D_Q,                            ;UPDATED STACK POINTER READY TO STORE
U 0D16, 0C01,1C3C,0180,F988,0000,00BD   ;25231          PSL.MODE?                       ;ARE WE ON THE INTERRUPT STACK?
                                ;25232
                                ;25233  =11101  ;-------------------------------;PSL<IS>=1
                                ;25234  SVPC.1: Q_ID[PCBB],                     ;PREPARE TO STORE INTO PCB
U 00BD, 0000,003C,E9F0,2C00,0000,0D20   ;25235          J/SVPC.4
                                ;25236
                                ;25237          ;-------------------------------;PSL<IS>=0
U 00BF, 0000,003C,3DF0,2C00,0000,0D18   ;25238          Q_ID[PSL]                       ;MUST GET ONTO INTERRUPT STACK
                                ;25239
                                ;25240  ;HERE IF THE INSTRUCTION WAS EXECUTED FROM THE KERNEL STACK.  WE MUST
                                ;25241  ;GET ONTO THE INTERRUPT STACK BEFORE PROCEEDING
                                ;25242
                                ;25243          ;-------------------------------;
                                ;25244  SVPC.2: ID[KSP]_D,                      ;SAVE SP AS KSP
U 0D18, 0801,201C,A180,3C00,0000,0D19   ;25245          D_Q.ORNOT.MASK                  ;SET PSL<IS>
                                ;25246
                                ;25247          ;-------------------------------;
                                ;25248          D_D.SWAP,                       ;RE-ARRANGE PSL FOR TESTING IPL FIELD
U 0D19, 0B00,003C,B1F0,2C00,0000,0D1A   ;25249          Q_ID[ISP]                       ;GET ISP READY AS NEW SP
                                ;25250
                                ;25251          ;-------------------------------;
                                ;25252          ALU_D.AND.K[.1F00],             ;CHECK IPL FIELD FOR ZERO
                                ;25253          LONG,CLK.UBCC,
U 0D1A, 0B19,0034,9180,F800,0010,0D1B   ;25254          D_D.SWAP                        ;RESTORE FORMAT OF PSL
                                ;25255
                                ;25256          ;-------------------------------;
                                ;25257          R[SP]_Q,                        ;SETUP ISP AS SP
                                ;25258          SC_K[.10],                      ;GET 16 FOR SETTING IPL TO 1, IF NEEDED
U 0D1B, 0001,213C,6580,FAF0,0084,68AC   ;25259          Z?                              ;IS THE IPL FIELD ZERO?
                                ;25260
                                ;25261  =0      ;-------------------------------;ALU Z=0.  IPL FIELD IS NON-ZERO
                                ;25262  SVPC.3: ID[PSL]_D,                      ;STORE UPDATED PSL, THEN GO SAVE REGS
U 08AC, 0000,003C,3D80,3C00,0000,00BD   ;25263          J/SVPC.1
                                ;25264
                                ;25265          ;-------------------------------;ALU Z=1.  IPL FIELD IS ZERO
                                ;25266          D_D.ORNOT.MASK,                 ;SET BIT 16 TO GIVE IPL = 1
U 08AD, 0801,001C,0180,F800,0000,08AC   ;25267          J/SVPC.3
```

```
                                    ;25268  ;HERE TO BEGIN STORING REGISTERS INTO THE PROCESS CONTROL BLOCK
                                    ;25269
                                    ;25270          ;-------------------------------;
                                    ;25271  SVPC.4: SC_K[.28],                      ;SET ID BUS ADDRESS OF KSP
U 0D20, 0001,203C,2D80,F800,0284,6D21 ;25272        VA_Q                            ;GET PHYSICAL ADDRESS OF PCB
                                    ;25273
                                    ;25274          ;-------------------------------;
                                    ;25275          Q_ID(SC),                       ;GET KSP READY TO STORE
U 0D21, 0000,003C,01F0,2400,0080,CD22 ;25276        SC_SC+1                         ;ADDRESS NEXT SP
                                    ;25277
                                    ;25278          ;-------------------------------;
                                    ;25279  SVPC.5: D_Q,                            ;READY SP FOR STORING TO MEMORY
                                    ;25280          FE_SC.ANDNOT.K[.FFFC],          ;CHECK FOR 4 POINTERS STORED
U 0D22, 0C00,003C,F180,F800,0114,4D24 ;25281        CLR.UBCC
                                    ;25282
                                    ;25283          ;-------------------------------;
                                    ;25284          CACHE.P_D[LONG],                ;STORE A POINTER
U 0D24, 0000,123C,0180,A800,0000,0BFB ;25285        EALU.Z?                         ;SHOULD WE STORE MORE?
                                    ;25286
                                    ;25287  =1011   ;-------------------------------;EALU Z=0 (MORE TO GO)
                                    ;25288          Q_ID(SC),                       ;GET NEXT POINTER
                                    ;25289          SC_SC+1,                        ;BUMP ADDRESS
                                    ;25290          VA_VA+4,                        ;ADDRESS TO STORE INTO
U 0BFB, 0000,003C,01F0,2403,0080,CD22 ;25291        J/SVPC.5                        ;LOOP TO STORE ANOTHER
                                    ;25292
                                    ;25293          ;-------------------------------;EALU Z=1 (ALL SP'S STORED)
                                    ;25294          SC_FE,                          ;CLEAR SC (REGISTER NUMBER)
                                    ;25295          FE_K[.D],                       ;SET TERMINATING VALUE
                                    ;25296          CLR.UBCC,                       ;FORCE EALU Z=0
                                    ;25297          D_R[R0],                        ;GET FIRST REGISTER TO STORE
U 0BFF, 0800,003C,8980,FA03,0195,6D25 ;25298        VA_VA+4                         ; BUMP ADDRESS FOR IT
```

```
                                    ;25299  ;HERE TO STORE GENERAL REGISTERS INTO THE PCB
                                    ;25300
                                    ;25301          ;-------------------------------;
                                    ;25302  SVPC.6: CACHE.P_D[LONG]                 ;STORE A REGISTER
                                    ;25303          SC_SC+1,                        ;BUMP REG NUMBER
U 0D25, 0000,123C,0180,A800,0080,CC0B ;25304        EA[U.Z?                         ;LAST ONE?
                                    ;25305
                                    ;25306  =1011   ;-------------------------------;EALU Z=0, MORE REG'S TO GO
                                    ;25307          D_R(SC),                        ;GET A REGISTER
                                    ;25308          VA_VA+4,                        ;BUMP STORAGE ADDRESS
                                    ;25309          EA[U_SC-FE,CLK.UBCC,            ;TEST FOR LAST ONE
U 0C0B, 0800,003C,0180,F86B,0010,AD25 ;25310        J/SVPC.6                        ;LOOP BACK
                                    ;25311
                                    ;25312          ;-------------------------------;EALU Z=1.  R0-R13 ALL STORED
                                    ;25313          D_RC[T0],                       ;GET STACKED PC
U 0C0F, 0810,0038,0180,F903,0000,0D26 ;25314        VA_VA+4
                                    ;25315
                                    ;25316          ;-------------------------------;
U 0D26, 0000,003C,0180,A800,0000,0D28 ;25317        CACHE.P_D[LONG]                 ;STORE STACKED PC INTO PCB
                                    ;25318
                                    ;25319          ;-------------------------------;
                                    ;25320          D_RC[T1],                       ;GET STACKED PSL
U 0D28, 0810,0038,0180,F90B,0000,0D29 ;25321        VA_VA+4                         ;BUMP ADDRESS
                                    ;25322
                                    ;25323          ;-------------------------------;
                                    ;25324          CACHE.P_D[LONG],                ;STORE STACKED PSL INTO PCB
                                    ;25325          CLR.IB.OPC,                     ;GO ON TO NEXT INSTR
U 0D29, C00C,003C,0180,A804,4000,0062 ;25326        PC_PC+1,J/IRD
```

```
                                    ;25327  .TOC    ""      SVP/LDPCTX, MF/MTPR   : LDPCTX - Load Process Context""
                                    ;25328
                                    ;25329  ;HERE FROM A-FORK TO LOAD PROCESSOR REGISTERS FROM PCB
                                    ;25330
                                    ;25331  089:    ;-------------------------------;
U 0089, 0000,003C,E9F0,2C00,0000,04C4 ;25332  LDPCTX: Q_ID[PCBB]                      ;GET PHYSICAL ADDRESS OF PCB
                                    ;25333
                                    ;25334  =0***** ;-------------------------------;
                                    ;25335          VA_Q,                           ;ITS LEGAL.  LOAD PCB BASE ADDRESS
                                    ;25336          SC_K[.28],STATE_K[.28],         ;SETUP ID BUS ADDR OF FIRST SP
U 04C4, 0001,3C3D,2D80,F800,1684,603E ;25337          PS[.MODE?,CALL,J/KERNEL.CHK     ;MAKE SURE THIS INSTRUCTION IS LEGAL NOW
                                    ;25338
                                    ;25339          ;-------------------------------;
                                    ;25340  LDPC.1: D[LONG]_CACHE.P,                ;GET NEXT SP VALUE FROM PCB
                                    ;25341          FE_SC+1,                        ;COUNT ID ADDRESS TO NEXT
U 04E4, 0000,173C,0180,C800,0100,CC1B ;25342          STATE2?                         ;WRITTEN ALL 4?
                                    ;25343
                                    ;25344  =1011   ;-------------------------------;EALU Z=0.  LOAD ANOTHER SP
                                    ;25345          ID(SC)_D,                       ;WRITE NEXT STACK POINTER
                                    ;25346          VA_VA+4,                        ;ADVANCE ADDRESS TO NEXT
                                    ;25347          SC_FE,STATE_FE,                 ;GET NEXT STACK POINTER ADDRESS
U 0C1B, 0000,003C,0180,3403,1481,64E4 ;25348          J/LDPC.1                        ;GO READ ANOTHER
                                    ;25349
                                    ;25350          ;-------------------------------;EALU Z=1.  BEGIN LOADING REGISTERS
                                    ;25351          FE_SC,                          ;USE LEFTOVER X'2C TO TERMINATE LOOP
                                    ;25352          SC_K[.20].ALU,                  ;SETUP INITIAL REGISTER ADDR
U 0C1F, 0018,0038,7580,F800,0182,0C2B ;25353          J/LDPC.9                        ;BEGIN LOOP
                                    ;25354
                                    ;25355          ;-------------------------------;
                                    ;25356  LDPC.2: D[LONG]_CACHE.P,                ;GET NEXT REGISTER
                                    ;25357          SC_SC+1,                        ;COUNT TO ITS ADDRESS
U 0D2A, 0000,123C,0180,C800,0080,CC2B ;25358          EALU.Z?                         ;HAVE WE GOT TO 13 YET?
                                    ;25359
                                    ;25360  =1011   ;-------------------------------;EALU Z=0.  MORE REGS TO GO
                                    ;25361  LDPC.9: R(SC)_D,                        ;LOAD NEXT
                                    ;25362          VA_VA+4,                        ;ADVANCE ADDRESS
                                    ;25363          EALU_SC-FE,CLK.UBCC,            ;CHECK FOR LAST TRANSFER
U CC2B, 0001,003C,0180,F8EB,0010,AD2A ;25364          J/LDPC.2                        ;LOOP BACK
                                    ;25365
                                    ;25366          ;-------------------------------;EALU Z=1.  STORE REG 13
                                    ;25367          R(SC)_D,                        ;LOAD FINAL REGISTER
                                    ;25368          VA_VA+4,                        ;ADVANCE ADDRESS TO SAVED PC
U 0C2F, 0001,003C,7D80,F8EB,0084,6D2B ;25369          SC_K[.18]                       ;GET 24 FOR BUILDING MASK
```

```
                                ;25370  ;HERE LDPCTX HAS LOADED THE STACK POINTERS AND GENERAL REGISTERS
                                ;25371  ; FROM THE PROCESS CONTROL BLOCK INTO INTERNAL REGISTERS.  NOW THE PC AND
                                ;25372  ; PSL, THEN THE MAPPING REGISTERS MUST BE OBTAINED FROM THE PCB.
                                ;25373
                                ;25374          ;-------------------------------;
                                ;25375          D[LONG]_CACHE.P,                ;GET SAVED PC FROM PCB
U 0D2B, 0003,0010,0180,C988,0000,0D30   ;25376          RC[T1]_0+MASK+1                 ;GET 1S IN BITS 31-24
                                ;25377
                                ;25378          ;-------------------------------;
                                ;25379          VA_VA+4,                        ;ADVANCE ADDRESS TO SAVED PSL
U 0D30, 0001,003C,0180,F993,0000,0D31   ;25380          RC[T2]_D                        ;SAVE PCB[PC] FOR PUSHING ONTO STACK
                                ;25381
                                ;25382          ;-------------------------------;
U 0D31, 0000,003C,0180,C800,0000,0D32   ;25383          D[LONG]_CACHE.P                 ;GET PCB[PSL]
                                ;25384
                                ;25385          ;-------------------------------;
                                ;25386          VA_VA+4,                        ;ADVANCE ADDRESS TO P0BR
U 0D32, 0000,003C,C180,3C03,0000,0425   ;25387          ID[T0]_D                        ;SAVE PCB[PSL] FOR PUSHING ONTO STACK
                                ;25388
                                ;25389  =00***  ;-------------------------------;
                                ;25390          D[LONG]_CACHE.P,                ;GET P0BR VALUE
                                ;25391          SC_K[.8],                       ;SET PROC REG # OF P0BR
U 0425, 0000,003D,0180,C800,0084,6548   ;25392          CA[L,J/WPR.2                    ;GO LOAD P0BR
                                ;25393
                                ;25394  ::10***  ;-------------------------------;RETURN HERE IF P0BR VALUE ILLEGAL
U 0435, 0000,003C,0180,F800,0000,0106   ;25395          J/RSVOPR                        ;BOMB
                                ;25396
                                ;25397          ;-------------------------------;RETURN HERE WITH P0BR LOADED
U 043D, 0000,003C,0180,F803,0000,0444   ;25398          VA_VA+4                         ;ADVANCE ADDRESS TO P0LR & AST
```

```
                                    ;25399  ;HERE AFTER OBTAINING PC, PSL, AND P0BR.
                                    ;25400
                                    ;25401  =00***  ;--------------------------------;
                                    ;25402          D[LONG]_CACHE.P,                 ;GET POLR AND AST
                                    ;25403          Q_RC[T1],                        ;GET MASK FOR SEPARATING THEM
U 0444, 0010,0039,01C0,C908,0000,0445 ;25404        CALL,J/LDPC.3
                                    ;25405
                                    ;25406  =10***  ;--------------------------------;
U 0454, 0000,003C,0180,F800,0000,0106 ;25407        J/RSVOPR
                                    ;25408
                                    ;25409          ;--------------------------------;RETURN HERE AFTER POLR SET UP
                                    ;25410          D[LONG]_CACHE.P,                 ;GET P1BR
                                    ;25411          SC_K[.A],                        ;SET UP PROC REG # OF P1BR
U 045C, 0000,003C,F580,C800,0084,64A5 ;25412        J/LDPC.4
                                    ;25413
                                    ;25414  =00***  ;--------------------------------;
                                    ;25415  LDPC.3: RC[T1]_D.ANDNOT.Q,               ;SAVE POLR IN T1
                                    ;25416          D_D.SWAP,                        ;GET AST INTO LOW BYTE
                                    ;25417          VA_VA+4,                         ;ADVANCE ADDRESS TO P1BR
U 0445, 081D,0025,0180,F988,0000,0553 ;25418        CALL,J/WPR.6                     ;GO LOAD AST LEVEL
                                    ;25419
                                    ;25420  =10***  ;--------------------------------;RETURN HERE IF AST LEVEL OUT OF RANGE
U 0455, 0000,003C,0180,F800,0000,0106 ;25421        J/RSVOPR
                                    ;25422
                                    ;25423          ;--------------------------------;RETURN HERE HAVING SET UP ASTLVL
                                    ;25424          D_RC[T1],                        ;GET POLR BACK
                                    ;25425          SC_K[.9],FE_EALU,                ;SET UP ITS PROC REG NUMBER
U 045D, 0810,0038,D980,F908,0184,6549 ;25426        J/WPR.3C
                                    ;25427
                                    ;25428  =00***  ;--------------------------------;
                                    ;25429  LDPC.4: Q_D,                             ;SAVE P1BR IN Q
                                    ;25430          D_K[.8000],                      ;BEGIN BUILDING 2**23
                                    ;25431          VA_VA+4,                         ;ADVANCE ADDR TO P1LR & PME
U 04A5, 0818,0039,45E0,F803,0000,0D53 ;25432        CALL,J/WPR.2A                    ;GO LOAD P1BR
                                    ;25433
                                    ;25434  =10***  ;--------------------------------;HERE IF P1BR IS ILLEGAL
U 04B5, 0000,003C,0180,F800,0000,0106 ;25435        J/RSVOPR
                                    ;25436
                                    ;25437          ;--------------------------------;P1BR LOADED OK
                                    ;25438          D[LONG]_CACHE.P,                 ;GET P1LR & PME
                                    ;25439          Q_0,                             ;CLEAR Q FOR SHIFTING IN D31
U 04BD, 0000,003C,8DF8,C800,0084,65A1 ;25440        SC_K[.1F]                        ;SETUP 31 IN SC FOR CLEARING PME
```

```
                                        ;25441  ;HERE AFTER LOADING P0BR, P0LR, AND P1BR.
                                        ;25442
                                        ;25443  =00***  ;-------------------------------;
                                        ;25444          RC[T1]_D.AND.MASK,              ;CLEAR PME BIT IN P1LR
                                        ;25445          Q_Q.LEFT,SI/ASHL,               ;GET PME INTO Q0
                                        ;25446          D_0,                            ;PREVENT ERROR RETURN FROM LOADING PME
U 05A1, 0F01,0035,0128,F988,0000,01AA   ;25447          CALL,J/LDPC.5
                                        ;25448
                                        ;25449  =10***  ;-------------------------------;ERROR RETURN FROM LOADING P1LR
U 05B1, 0000,003C,0180,F800,0000,0106   ;25450          J/RSVOPR
                                        ;25451
                                        ;25452          ;-------------------------------;SUCCESS RETURN FROM LOADING P1LR
                                        ;25453          D_R[SP],                        ;GET SP AVAILABLE FOR STORING
                                        ;25454          SC_K[.1A],                      ;GET BIT # (26) OF PSL<IS>
U 05B9, 0800,1C3C,E580,FA70,0084,60DD   ;25455          PSL[.MODE?,J/LDPC.6             ;IS PSL<IS> SET?
                                        ;25456
                                        ;25457  =01***  ;-------------------------------;
                                        ;25458  LDPC.5: RC[T0]_Q.LEFT3,                 ;SAVE 8* PME
                                        ;25459          Q_ID[CES],                      ;GET CPU REG, WHICH CONTAINS PME
U 01AA, 00A1,203D,31F0,2D80,0000,0D71   ;25460          CALL,J/WPR.9                    ;GO STORE PME
                                        ;25461
                                        ;25462          ;-------------------------------;RETURN HERE FROM SETTING PME
                                        ;25463          D_RC[T1],                       ;GET P1LR
                                        ;25464          SC_SC-K[.1],FE_EALU,            ;GET PROC REG # (0B) OF P1LR
U 01BA, 0810,0038,0580,F908,0184,A549   ;25465          J/WPR.3C                        ;GO STORE P1LR
```

```
                                ;25466  ;HERE ALL THE REGISTERS HAVE BEEN LOADED FROM THE PCB.
                                ;25467  ;IF PSL<IS> IS SET, WE MUST GET OFF THE INTERRUPT STACK
                                ;25468
                                ;25469  =11101  ;--------------------------------;PSL<IS>=1
                                ;25470  LDPC.6: Q_ID[PSL],                      ;PREPARE TO CLEAR IT
U 00DD, 0000,003C,3DF0,2C00,0000,0D38   ;25471          J/LDPC.8
                                ;25472
                                ;25473          ;--------------------------------;PSL<IS>=0
                                ;25474  LDPC.7: Q_ID[TBER0],                     ;PREPARE TO SET TB INVALIDATE BOTH
U 00DF, 1000,003C,49F0,2C00,0000,01CA   ;25475          STOP.IB
                                ;25476
                                ;25477  =01***  ;--------------------------------;
                                ;25478          FE_K[.20],                      ;LOOP COUNT FOR CLEARING PROCESS SPACE
                                ;25479          D_0,                            ;CHANGE NOTHING IN TBER0
U 01CA, 0F00,003D,7580,F800,0104,6D73   ;25480          CALL,J/WPR.10B                  ;GO INVALIDATE TB
                                ;25481
                                ;25482          ;--------------------------------;RETURN FROM TB INVALIDATION
                                ;25483          LAB_R[SP],                      ;GET KERNEL SP TO LATCH
U 01DA, 0000,003C,C1F0,2E70,0000,0D34   ;25484          Q_ID[T0]                        ;GET SAVED PSL FROM PCB
                                ;25485
                                ;25486  ;FINALLY, WE ARE ON THE KERNEL STACK.  PUSH ON PSL AND PC FROM PCB
                                ;25487
                                ;25488          ;--------------------------------;
                                ;25489          VA_LA-K[.4],                    ;READY TO PUSH PSL
U 0D34, 0C18,0000,1180,F800,0200,0D35   ;25490          D_Q                             ;GET PSL INTO D
                                ;25491
                                ;25492          ;--------------------------------;
                                ;25493          CACHE_D[LONG],                  ;PUSH PCB[PSL] ONTO STACK
U 0D35, 0010,0038,01C0,3110,0000,0D36   ;25494          Q_RC[T2]                        ;GET PCB[PC]
                                ;25495
                                ;25496          ;--------------------------------;
                                ;25497          R[SP]&VA_LA-K[.8].RLOG,         ;READY TO PUSH PCB[PC]
                                ;25498          D_Q,                            ;GET DATA READY
U 0D36, 0C18,0004,018C,FAF0,0200,03FD   ;25499          J/STOR.L                        ;STORE, THEN GO TO NEXT INSTR
                                ;25500
                                ;25501  ;HERE TO GET ONTO THE KERNEL STACK FROM THE INTERRUPT STACK
                                ;25502
                                ;25503          ;--------------------------------;
                                ;25504  LDPC.8: ID[ISP]_D,                      ;SAVE SP AS ISP
U 0D38, 0801,2034,B180,3C00,0000,0D39   ;25505          D_Q.AND.MASK                    ;CLEAR IS BIT IN PSL
                                ;25506
                                ;25507          ;--------------------------------;
U 0D39, 0000,003C,A1F0,2C00,0000,0D3A   ;25508          Q_ID[KSP]                       ;GET KSP
                                ;25509
                                ;25510          ;--------------------------------;
                                ;25511          ID[PSL]_D,                      ;NOW OFF INTERRUPT STACK
                                ;25512          R[SP]_Q,                        ;SETUP KSP AS CURRENT SP
U 0D3A, 0001,203C,3D80,3EF0,0000,00DF   ;25513          J/LDPC.7                        ;REJOIN KERNEL FLOW
```

```
                                ;25514  .TOC     ''      SVP/LDPCTX, MF/MTPR   : MFPR - Move From Processor Register''
                                ;25515
                                ;25516  ;GET HERE WITH PROCESSOR REGISTER ADDRESS IN D
                                ;25517
                                ;25518  2C7:    ;-------------------------------;
U 02C7, 0000,1C3D,0180,F800,0000,003E   ;25519  MFPR:   PSL.MODE?,CALL,J/KERNEL.CHK     ;SHOULD WE ALLOW THIS OPERATION?
                                ;25520
                                ;25521  2E7:    ;-------------------------------;RETURN FROM KERNEL CHK MEANS OK
U 02E7, 0000,003D,0180,F800,0000,0500   ;25522          CALL,J/RPR                      ;GO READ REGISTER
                                ;25523
                                ;25524  2F7:    ;-------------------------------;
U 02F7, 0000,003C,0180,F800,0000,0106   ;25525          J/RSVOPR                        ;TAKE RESERVED OPERAND TRAP
                                ;25526
                                ;25527  2FF:    ;-------------------------------;
                                ;25528          D_Q,                            ;SUCCESS.  WRITE RESULT
U 02FF, FC00,003F,01F0,F847,0000,0200   ;25529          B.FORK
                                ;25530
                                ;25531  ;HERE IS RPR SUBROUTINE, WHICH DOES THE WORK OF MFPR
                                ;25532  ;IT IS A SUBROUTINE SO AS TO BE USABLE BY THE CONSOLE INTERFACE
                                ;25533
                                ;25534  =000000 ;-------------------------------;MAKE SPACE FOR MANY REGISTER NUMBERS
                                ;25535  RPR:    SC_D-K[.8],CLK.UBCC,            ;IS THIS A STACK POINTER?
                                ;25536          RC[TO]_ALU.RIGHT,               ;SAVE BITS 2-1 IN RC
                                ;25537          FE_K[.8],                       ;READY FE TO RESTORE PRN IN SC
U 0500, 0059,0001,0180,F980,0196,6D7B   ;25538          CALL,J/SELPR                    ;GO SELECT PROCESSOR REGISTER
                                ;25539
                                ;25540  =000010 ;-------------------------------;REG # .LSSU. 8, STACK POINTER
                                ;25541          Q_ID(SC),                       ;GET SELECTED STACK POINTER
U 0502, 0000,003E,01F0,2400,0000,0018   ;25542          RETURN18
                                ;25543
                                ;25544          ;-------------------------------;REG # .EQL. CURRENT MODE
                                ;25545          Q_R[SP],                        ;GET CURRENT STACK POINTER
U 0503, 0000,003E,01C0,FA70,0000,0018   ;25546          RETURN18
                                ;25547
                                ;25548  =001000 ;-------------------------------;08:  P0BR
                                ;25549          Q_ID[P0BR],                     ;PROCESS 0 BASE REGISTER
U 0508, 0000,003E,91F0,2C00,0000,0018   ;25550          RETURN18                        ;RETURN IT
                                ;25551
                                ;25552          ;-------------------------------;09: P0LR
                                ;25553          Q_ID[P0LR],                     ;PROCESS 0 LENGTH REGISTER
U 0509, 0000,003E,F1F0,2C00,0000,0018   ;25554          RETURN18
                                ;25555
                                ;25556          ;-------------------------------;0A: P1BR
                                ;25557          D_K[.8000],                     ;PROCESS 1 BASE REGISTER
U 050A, 0818,0038,4580,F800,0000,0D3B   ;25558          J7RPR.2                         ;BEGIN BUILDING 2**23
                                ;25559
                                ;25560          ;-------------------------------;0B: P1LR
                                ;25561          D_K[.8000].RIGHT2,              ;BEGIN BUILDING 2**21
U 050B, 0898,0038,4580,F800,0000,0D41   ;25562          J7RPR.3
```

```
                                    ;25563  ;MFPR CONTINUED
                                    ;25564
                                    ;25565          ;-------------------------------;0C: SBR
                                    ;25566          Q_ID[SBR],                      ;SYSTEM BASE REGISTER
U 050C, 0000,003E,99F0,2C00,0000,0018 ;25567        RETURN18
                                    ;25568
                                    ;25569          ;-------------------------------;0D: SLR
                                    ;25570          Q_ID[SLR],                      ;SYSTEM LENGTH REGISTER
U 050D, 0000,003E,F9F0,2C00,0000,0018 ;25571        RETURN18
                                    ;25572
                                    ;25573  ;0E & 0F ARE SELECTED OUT BY SELPR
                                    ;25574
                                    ;25575  =010000 ;-------------------------------;10: PCBB
                                    ;25576          Q_ID[PCBB],                     ;PROCESS CONTROL BLOCK BASE
U 0510, 0000,003E,E9F0,2C00,0000,0018 ;25577        RETURN18
                                    ;25578
                                    ;25579          ;-------------------------------;11: SCBB
                                    ;25580          Q_ID[SCBB],                     ;SYSTEM CONTROL BLOCK BASE REGISTER
U 0511, 0000,003E,EDF0,2C00,0000,0018 ;25581        RETURN18
                                    ;25582
                                    ;25583          ;-------------------------------;12: IPL
                                    ;25584          SC_K[.10],                      ;SETUP 16 IN SC FOR SHIFTING
U 0512, 0000,003C,6580,F800,0084,6D43 ;25585        J/RPR.5
                                    ;25586
                                    ;25587          ;-------------------------------;13: ASTR
                                    ;25588          Q_ID[CES],                      ;ASYNCRONOUS SYSTEM TRAP REGISTER
U 0513, 0000,003C,31F0,2C00,0000,0D48 ;25589        J/RPR.6
                                    ;25590
                                    ;25591          ;-------------------------------;14: SIRR
U 0514, 0000,003E,0180,F800,0000,0010 ;25592        RETURN10                        ;CAN'T READ REQUEST NUMBER
                                    ;25593
                                    ;25594          ;-------------------------------;15: SISR
                                    ;25595          Q_ID[SIR],                      ;SOFTWARE INTERRUPT SUMMARY REGISTER
U 0515, 0000,003E,39F0,2C00,0000,0018 ;25596        RETURN18
                                    ;25597
                                    ;25598  ; 16 & 17 ARE SELECTED OUT BY SELPR
```

```
                                        ;25599  ;CONTINUATION OF MFPR
                                        ;25600
                                        ;25601  =011000 ;-------------------------------;18: ICCS
                                        ;25602          Q_ID[CLK.CS],                   ;INTERVAL CLOCK CONTROL/STATUS
U 0518, 0000,003C,29F0,2C00,0000,0D49   ;25603          J7RPR.8
                                        ;25604
                                        ;25605          ;-------------------------------;19: NICR
                                        ;25606          Q_ID[NXT.PER],                  ;NEXT INTERVAL COUNT REGISTER
U 0519, 0000,003E,25F0,2C00,0000,0018   ;25607          RETURN18
                                        ;25608
                                        ;25609          ;-------------------------------;1A: ICR
                                        ;25610          Q_ID[INTERVAL],                 ;INTERVAL COUNT REGISTER
U 051A, 0000,003E,2DF0,2C00,0000,0018   ;25611          RETURN18
                                        ;25612
                                        ;25613          ;-------------------------------;1B: TODR
                                        ;25614          Q_ID[DAY.TIME],                 ;TIME OF DAY REGISTER
                                        ;25615          A[U_K[.1],CLK.UBCC,             ;GUARANTEE ALU Z=0
U 051B, 0018,0038,05F0,2C00,0010,0D4B   ;25616          J/RPR.9                         ;WAIT UNTIL TIME STOPS CHANGING
                                        ;25617
                                        ;25618          ;-------------------------------;1C
U 051C, 0000,003E,0180,F800,0000,0010   ;25619          RETURN10                        ;UNDEFINED
                                        ;25620
                                        ;25621          ;-------------------------------;1D
U 051D, 0000,003E,0180,F800,0000,0010   ;25622          RETURN10                        ;UNDEFINED
                                        ;25623
                                        ;25624  ; 1E & 1F ARE SELECTED OUT BY SELPR
                                        ;25625
                                        ;25626  =100000 ;-------------------------------;20: RXCS
                                        ;25627          Q_ID[RXCS],                     ;CONSOLE RECEIVE CONTROL/STATUS
U 0520, 0000,003E,11F0,2C00,0000,0018   ;25628          RETURN18
                                        ;25629
                                        ;25630          ;-------------------------------;21: RXDB
                                        ;25631          Q_ID[RXDB],                     ;CONSOLE RECEIVE DATA BUFFER
U 0521, 0000,003E,15F0,2C00,0000,0018   ;25632          RETURN18
                                        ;25633
                                        ;25634          ;-------------------------------;22: TXCS
                                        ;25635          Q_ID[TXCS],                     ;CONSOLE TRANSMIT CONTROL/STATUS
U 0522, 0000,003E,19F0,2C00,0000,0018   ;25636          RETURN18
                                        ;25637
                                        ;25638          ;-------------------------------;23: TXDB
U 0523, 0000,003E,0180,F800,0000,0010   ;25639          RETURN10                        ;CAN'T READ TRANSMIT DATA BUFFER
```

```
                                    ;25640  ;CONTINUATION OF MFPR
                                    ;25641
                                    ;25642          ;-------------------------------;24 (AND OTHER ILLEGAL VALUES)
U 0524, 0000,003E,0180,F800,0000,0010 ;25643        RETURN10                        ;UNDEFINED
                                    ;25644
                                    ;25645          ;-------------------------------;25
U 0525, 0000,003E,0180,F800,0000,0010 ;25646        RETURN10                        ;UNDEFINED
                                    ;25647
                                    ;25648  ; 26 & 27 ARE SELECTED OUT BY SELPR
                                    ;25649
                                    ;25650  =101000 ;-------------------------------;28: ACCS
                                    ;25651          Q_ID[ACC.CS],                   ;ACCELERATOR CONTROL/STATUS
U 0528, 0000,003E,5DF0,2C00,0000,0018 ;25652        RETURN18
                                    ;25653
                                    ;25654          ;-------------------------------;29: ACRR
                                    ;25655          Q_ID[ACC.0],                    ;ACCELERATOR RESERVED REGISTER
U 0529, 0000,003E,51F0,2C00,0000,0018 ;25656        RETURN18
                                    ;25657
                                    ;25658          ;-------------------------------;2A
U 052A, 0000,003E,0180,F800,0000,0010 ;25659        RETURN10                        ;UNDEFINED
                                    ;25660
                                    ;25661          ;-------------------------------;2B
U 052B, 0000,003E,0180,F800,0000,0010 ;25662        RETURN10                        ;UNDEFINED
                                    ;25663
                                    ;25664          ;-------------------------------;2C: WCSA
                                    ;25665          Q_ID[WCS.ADDR],                 ;WRITABLE CONTROL STORE ADDRESS
U 052C, 0000,003E,89F0,2C00,0000,0018 ;25666        RETURN18
                                    ;25667
                                    ;25668          ;-------------------------------;2D: WCSD
                                    ;25669          Q_ID[WCS.DATA],                 ;WRITABLE CONTROL STORE DATA
U 052D, 0000,003E,8DF0,2C00,0000,0018 ;25670        RETURN18
                                    ;25671
                                    ;25672  ; 2E & 2F ARE SELECTED OUT BY SELPR
```

```
                                    ;25673  ;CONTINUATION OF MFPR
                                    ;25674
                                    ;25675  =110000 ;-------------------------------;30: SBIFS
                                    ;25676          Q_ID[FAULT],                    ;SBI FAULT/STATUS
U 0530, 0000,003E,6DF0,2C00,0000,0018 ;25677        RETURN18
                                    ;25678
                                    ;25679          ;-------------------------------;31: SBIS
                                    ;25680          Q_ID[SILO],                     ;SBI SILO
U 0531, 0000,003E,61F0,2C00,0000,0018 ;25681        RETURN18
                                    ;25682
                                    ;25683          ;-------------------------------;32: SBISC
                                    ;25684          Q_ID[COMP],                     ;SBI SILO COMPARITOR
U 0532, 0000,003E,71F0,2C00,0000,0018 ;25685        RETURN18
                                    ;25686
                                    ;25687          ;-------------------------------;33: SBIMT
                                    ;25688          Q_ID[MAINT],                    ;SBI MAINTENANCE
U 0533, 0000,003E,75F0,2C00,0000,0018 ;25689        RETURN18
                                    ;25690
                                    ;25691          ;-------------------------------;34: SBIER
                                    ;25692          Q_ID[SBI.ERR],                  ;GET SBI ERROR REGISTER
U 0534, 0000,003E,65F0,2C00,0000,0018 ;25693        RETURN18
                                    ;25694
                                    ;25695          ;-------------------------------;35: SBITA
                                    ;25696          Q_ID[TIME.ADDR],                ;TIMEOUT ADDRESS
U 0535, 0000,003E,69F0,2C00,0000,0018 ;25697        RETURN18
                                    ;25698
                                    ;25699          ;-------------------------------;36: SBIQC
U 0536, 0000,003E,0180,F800,0000,0010 ;25700        RETURN10                        ;CAN'T READ A QUAD CLEAR!
                                    ;25701
                                    ;25702  ; 37 IS SELECTED OUT BY SELPR
```

```
                                    ;25703  ;CONTINUATION OF MFPR
                                    ;25704
                                    ;25705  =111000 ;-------------------------------;38: MME
                                    ;25706          Q_ID[TBER0],                    ;GET TB STATUS
U 0538, 0000,003C,49F0,2C00,0000,0D52 ;25707        J7RPR.11                        ;GET MEM MGMT ENABLE FROM IT
                                    ;25708
                                    ;25709          ;-------------------------------;39: TBIA
U 0539, 0000,003E,0180,F800,0000,0010 ;25710        RETURN10                        ;CANNOT READ TB INVALIDATE
                                    ;25711
                                    ;25712          ;-------------------------------;3A: TBIS
U 053A, 0000,003E,0180,F800,0000,0010 ;25713        RETURN10                        ; DITTO FOR SINGLE INVALIDATE
                                    ;25714
                                    ;25715          ;-------------------------------;3B
U 053B, 0000,003E,0180,F800,0000,0010 ;25716        RETURN10                        ;UNDEFINED
                                    ;25717
                                    ;25718          ;-------------------------------;3C: MBRK
                                    ;25719          Q_ID[UBREAK],                   ;MICROBREAK
U 053C, 0000,003E,85F0,2C00,0000,0018 ;25720        RETURN18
                                    ;25721
                                    ;25722          ;-------------------------------;3D: PMR
                                    ;25723          Q_ID[CES],                      ;GET CPU ERROR/STATUS WHICH HAS PME
U 053D, 0000,003C,31F0,2C00,0000,0D50 ;25724        J7RPR.10
                                    ;25725
                                    ;25726          ;-------------------------------;3E: SID
                                    ;25727          Q_ID[SYS.ID],                   ;SYSTEM IDENTIFICATION
U 053E, 0000,003E,0DF0,2C00,0000,0018 ;25728        RETURN18
                                    ;25729
                                    ;25730  ; 3F AND ABOVE ARE SELECTED OUT BY SELPR
                                    ;25731  =;END OF RPR CONSTRAINT BLOCK
```

```
                                    ;25732  ;HERE TO READ P1BR
                                    ;25733
                                    ;25734          ;--------------------------------;
                                    ;25735  RPR.2:  D_D.SWAP,                       ;GENERATE CONSTANT 2**23
U 0D3B, 0B00,003C,95F0,2C00,0000,0D40 ;25736        Q_ID[P1BR]                      ;READ INTERNAL VALUE
                                    ;25737
                                    ;25738          ;--------------------------------;
                                    ;25739          Q_Q+D,                          ;RESTORE OFFSET
U 0D40, 001D,2016,01C0,F800,0000,0018 ;25740        RETURN18
                                    ;25741
                                    ;25742  ;HERE TO READ P1LR
                                    ;25743
                                    ;25744          ;--------------------------------;
                                    ;25745  RPR.3:  D_D.SWAP,                       ;CONSTANT 2**21
U 0D41, 0B00,003C,F5F0,2C00,0000,0D42 ;25746        Q_ID[P1LR]                      ;GET INTERNAL VALUE
                                    ;25747
                                    ;25748          ;--------------------------------;
                                    ;25749          Q_Q-D,                          ;REMOVE OFFSET
U 0D42, 001D,2002,01C0,F800,0000,0018 ;25750        RETURN18
                                    ;25751
                                    ;25752  ;HERE TO READ IPL
                                    ;25753
                                    ;25754          ;--------------------------------;
U 0D43, 0000,003C,3DF0,2C00,0000,0D44 ;25755  RPR.5:  Q_ID[PSL]                      ;IPL IS PSL BITS 20-16
                                    ;25756
                                    ;25757          ;--------------------------------;
U 0D44, 0D00,003C,0180,F800,0000,0D46 ;25758        D_DAL.SC                        ;HIGH 16 BITS OF PSL TO LOW OF D
                                    ;25759
                                    ;25760          ;--------------------------------;
                                    ;25761          Q_D.AND.K[.1F],                 ;GET IPL ALONE
U 0D46, 0019,0036,8DC0,F800,0000,0018 ;25762        RETURN18
                                    ;25763
                                    ;25764  ;HERE TO GET ASTLVL
                                    ;25765
                                    ;25766          ;--------------------------------;
                                    ;25767  RPR.6:  Q_Q.AND.K[.7],                  ;STRIP OFF REST OF CES REGISTER
U 0D48, 0019,2036,5DC0,F800,0000,0018 ;25768        RETURN18
```

```
                                ;25769  ;HERE TO READ INTERVAL CLOCK CONTROL/STATUS
                                ;25770
                                ;25771          ;--------------------------------;
                                ;25772  RPR.8:   D_Q.AND.K[.8000],                ;TEST BIT 15
U 0D49, 0819,2034,4580,F800,0010,0D4A   ;25773          LONG,CLK.UBCC
                                ;25774
                                ;25775          ;--------------------------------;
                                ;25776          SC_K[.1F],                      ;SETUP FOR MASK TO SET BIT 31
                                ;25777          Q_Q.ANDNOT.D,                   ;CLEAR BIT 15
U 0D4A, 001D,2124,8DC0,F800,0084,68B0   ;25778          Z?                              ;WAS BIT 15 SET?
                                ;25779
                                ;25780  =C      ;--------------------------------;ALU Z=0, BIT 15 WAS SET
                                ;25781          Q_Q.ORNOT.MASK,                 ;SET BIT 31 INSTEAD
U 08B0, 0001,201E,01C0,F800,0000,0018   ;25782          RETURN18
                                ;25783
                                ;25784          ;--------------------------------;ALU Z=1, BIT 15 WAS CLEAR
U 08B1, 0000,003E,0180,F800,0000,0018   ;25785          RETURN18                        ;RETURN IT SO
                                ;25786
                                ;25787  ;HERE TO READ DAY CLOCK, WHICH MAY BE CHANGING
                                ;25788
                                ;25789          ;--------------------------------;
                                ;25790  RPR.9:   D_Q,                            ;SAVE LAST VALUE READ
                                ;25791          Q_ID[DAY.TIME],                 ;GET ANOTHER
U 0D4B, 0C00,013C,05F0,2C00,0000,08B4   ;25792          Z?                              ;WAS LAST COMPARE EQUAL?
                                ;25793
                                ;25794  =0      ;--------------------------------;ALU Z=0, NOT EQUAL
                                ;25795          ALU_D.XOR.Q,                    ;COMPARE MOST RECENT TWO VALUES READ
                                ;25796          LONG,CLK.UBCC,
U 08B4, 001D,0020,0180,F800,0010,0D4B   ;25797          J/RPR.9                         ;WAIT FOR TWO VALUES THE SAME
                                ;25798
                                ;25799          ;--------------------------------;ALU Z=1, EQUAL VALUES
                                ;25800          Q_D,                            ;GET KNOWN GOOD VALUE
U 08B5, 0000,003E,01E0,F800,0000,0018   ;25801          RETURN18                        ;GIVE IT BACK
                                ;25802
                                ;25803  ;HERE TO READ PERFORMANCE MONITOR ENABLE
                                ;25804
                                ;25805          ;--------------------------------;
U 0D50, 0059,2034,01C0,F800,0000,0D51   ;25806  RPR.10: Q_Q.AND.K[.8].RIGHT         ;GET THE BIT ALONE
                                ;25807
                                ;25808          ;--------------------------------;
                                ;25809          Q_Q.RIGHT2,                     ;MOVE IT TO BIT0
U 0D51, 0000,003E,0190,F800,0000,0018   ;25810          RETURN18
                                ;25811
                                ;25812  ;HERE TO READ MEMORY MANAGEMENT ENABLE
                                ;25813
                                ;25814          ;--------------------------------;
                                ;25815  RPR.11: Q_Q.AND.K[.1],                   ;SELECT MME BIT FROM TB STATUS 0
U 0D52, 0019,2036,05C0,F800,0000,0018   ;25816          RETURN18
```

```
                                ;25817  .TOC    ""      SVP/LDPCTX, MF/MTPR   : MTPR - Move To Processor Register""
                                ;25818  ;GET HERE WITH SOURCE IN Q, PROCESSOR REGISTER ADDRESS IN D
                                ;25819
                                ;25820  347:    ;-------------------------------;
U 0347, 0000,003C,0180,F800,0000,0600   ;25821  MTPR:   J/MTPR.1                 ;GET TO UNCONSTRAINED ADDRESS
                                ;25822
                                ;25823  =000*** ;-------------------------------;CALL SITE FOR MTPR
U 0600, 0000,1C3D,0180,F800,0000,003E   ;25824  MTPR.1: PSL.MODE?,CALL,J/KERNEL.CHK ;FIRST VALIDATE THIS USE
                                ;25825
                                ;25826  =100*** ;-------------------------------;RETURN FROM KERNEL CHK MEANS OK
U 0620, 0000,003D,0180,F800,0000,0540   ;25827          CALL,J/WPR              ;GO LOAD REGISTER
                                ;25828
                                ;25829  =110*** ;-------------------------------;
U 0630, 0000,003C,0180,F800,0000,0106   ;25830          J/RSVOPR                ;TAKE RESERVED OPERAND TRAP
                                ;25831
                                ;25832  =111*** ;-------------------------------;
                                ;25833          CLR.IB.OPC,             ;SUCCESSFUL COMPLETION
U 0638, C000,003C,0180,F804,4000,0062   ;25834          PC_PC+1,J/IRD
                                ;25835
                                ;25836  ;HERE IS WPR SUBROUTINE, WHICH DOES THE WORK OF MTPR
                                ;25837  ;IT IS A SUBROUTINE SO AS TO BE USABLE BY THE CONSOLE INTERFACE
                                ;25838
                                ;25839  =000000 ;-------------------------------;MAKE SPACE FOR MANY REGISTER NUMBERS
                                ;25840  WPR:    SC_D-K[.8],CLK.UBCC,    ;IS THIS A STACK POINTER?
                                ;25841          RC[TO]_ALU.RIGHT,       ;SAVE PRN BITS 2-1
                                ;25842          FE_K[.8],               ;READY FE TO RESTORE PRN IN SC
U 0540, 0059,0001,0180,F980,0196,6D7B   ;25843          CALL,J/SELPR            ;GO SELECT PROCESSOR REGISTER
                                ;25844
                                ;25845  =000010 ;-------------------------------;REG # .LSSU. 8, STACK POINTER
                                ;25846  WPR.1:  ID(SC)_D,               ;STORE STACK POINTER
U 0542, 0000,003E,0180,3400,0000,0018   ;25847          RETURNT8
                                ;25848
                                ;25849          ;-------------------------------;REG # .EQL. CURRENT MODE
                                ;25850          R[SP]_D,                ;STORE CURRENT STACK POINTER
U 0543, 0001,003E,0180,FAF0,0000,0018   ;25851          RETURNT8
                                ;25852
                                ;25853  =001000 ;-------------------------------;08:  P0BR
                                ;25854  WPR.2:  ALU_D.AND.K[.3],        ;PROCESS 0 BASE REGISTER
                                ;25855          CLK.UBCC,               ;TEST MBZ'S
                                ;25856          D_D.SWAP,               ;PREP TO TEST OTHER MBZ'S
                                ;25857          Q_0,                    ;CLEAR OFFSET
U 0548, 0B19,0D34,0DF8,F800,0010,0EF6   ;25858          D31?,J/WPR.2B           ;FIRST CHECK MBO IN BIT 31
                                ;25859
                                ;25860          ;-------------------------------;09: P0LR
                                ;25861  WPR.3C: Q_D,                    ;SAVE NEW LENGTH
                                ;25862          STATE_FE,               ;GET REG # TO STATE
                                ;25863          D_K[.8000].RIGHT2,      ;BEGIN BUILDING 2**21
U 0549, 0898,0038,45E0,F800,1400,6D5B   ;25864          J7WPR.3
                                ;25865
                                ;25866          ;-------------------------------;0A: P1BR
                                ;25867          Q_D,                    ;PROCESS 1 BASE REGISTER
                                ;25868          D_K[.8000],             ;BEGIN BUILDING 2**23
U 054A, 0818,0038,45E0,F800,0000,0D53   ;25869          J7WPR.2A                ;
```

```
                                ;25870          ;--------------------------------;0B: P1LR
                                ;25871          Q_D,                            ;SAVE NEW LENGTH
                                ;25872          STATE_FE,                       ;GET REG # TO STATE
                                ;25873          D_K[.8000].RIGHT2,              ;BEGIN BUILDING 2**21
U 054B, 0898,0038,45E0,F800,1400,6D5B   ;25874          J7WPR.3
                                ;25875
                                ;25876          ;--------------------------------;0C: SBR
                                ;25877          ALU_D.AND.K[.3],                ;SYSTEM BASE REGISTER
                                ;25878          CLK.UBCC,                       ;VERIFY MBZ
                                ;25879          D_D.SWAP,
U 054C, 0B19,0034,0D80,F800,0010,0D59   ;25880          J7WPR.2C
                                ;25881
                                ;25882          ;--------------------------------;0D: SLR
                                ;25883          Q_D,                            ;SAVE NEW LENGTH
                                ;25884          STATE_FE,                       ;GET REG # TO STATE
                                ;25885          D_K[.8000].RIGHT2,              ;BEGIN BUILDING 2**21
U 054D, 0898,0038,45E0,F800,1400,6D5B   ;25886          J7WPR.3
                                ;25887
                                ;25888  ; 0E & 0F ARE SELECTED OUT BY SELPR
                                ;25889
                                ;25890  =010000 ;--------------------------------;10: PCBB
                                ;25891          ALU_D.AND.K[.3],                ;PROCESS CONTROL BLOCK BASE REGISTER
                                ;25892          CLK.UBCC,                       ;VERIFY THAT BITS 1-0 ARE ZERO
                                ;25893          D_D.SWAP,                       ;READY TO CHECK 31-30
U 0550, 0B19,0034,0D80,F800,0010,0D61   ;25894          J7WPR.4
                                ;25895
                                ;25896          ;--------------------------------;11: SCBB
                                ;25897          ALU_D.AND.K[.3],                ;SYSTEM CONTROL BLOCK BASE REGISTER
                                ;25898          CLK.UBCC,                       ;VERIFY THAT BITS 1-0 ARE ZERO
                                ;25899          D_D.SWAP,                       ;READY TO CHECK 31-30
U 0551, 0B19,0034,0D80,F800,0010,0D61   ;25900          J7WPR.4
                                ;25901
                                ;25902          ;--------------------------------;12: IPL
                                ;25903          SC_K[.10],                      ;SETUP 16 IN SC FOR SHIFTING
                                ;25904          Q_0,                            ;AND A 0 FOR SHIFTING IN
U 0552, 0000,003C,65F8,F800,0084,6D62   ;25905          J7WPR.5
                                ;25906
                                ;25907          ;--------------------------------;13: ASTR
                                ;25908  WPR.6:  D_D.AND.K[.7],                  ;ASYNCHRONOUS SYSTEM TRAP
                                ;25909          R[TO]_ALU,                      ;SAVE NEW VALUE IN TO
U 0553, 0819,0034,5D80,F980,0000,0D6A   ;25910          J/WPR.6A
                                ;25911
                                ;25912          ;--------------------------------;14: SIRR
                                ;25913          Q_ID[SIR],                      ;GET INTERRUPT REQUEST REG
                                ;25914          SC_D,                           ;GET REQUEST NUMBER
U 0554, 0001,003C,39F0,2C00,0082,0D70   ;25915          J/WPR.7                         ;ADD THIS REQUEST TO SUMMARY
                                ;25916
                                ;25917          ;--------------------------------;15: SISR
                                ;25918  WPR.7A: ID[SIR]_D,                      ;SOFTWARE INTERRUPT SUMMARY REGISTER
U 0555, 0000,003E,3980,3C00,0000,0018   ;25919          RETURN18
                                ;25920
                                ;25921  ; 16 & 17 ARE SELECTED OUT BY SELPR
```

```
                                    ;25922  ;CONTINUATION OF MTPR
                                    ;25923
                                    ;25924  =011000 ;-------------------------------;18: ICCS
U 0558, 0000,0D3C,0180.F800,0000,0EFE ;25925        D31?,J/WPR.8                    ;INTERVAL CLOCK CONTROL/STATUS
                                    ;25926
                                    ;25927          ;-------------------------------;19: NICR
                                    ;25928          ID[NXT.PER]_D,                  ;NEXT INTERVAL COUNT REGISTER
U 0559, 0000,003E,2580,3C00,0000,0018 ;25929        RETURN18
                                    ;25930
                                    ;25931          ;-------------------------------;1A: ICR
U 055A, 0000,003E,0180,F800,0000,0010 ;25932        RETURN10                        ;CANNOT WRITE INTERVAL REGISTER
                                    ;25933
                                    ;25934          ;-------------------------------;1B: TODR
                                    ;25935          ID[DAY.TIME]_D,                 ;TIME OF DAY REGISTER
U 055B, 0000,003E,0580,3C00,0000,0018 ;25936        RETURN18
                                    ;25937
                                    ;25938          ;-------------------------------;1C
U 055C, 0000,003E,0180,F800,0000,0010 ;25939        RETURN10                        ;UNDEFINED
                                    ;25940
                                    ;25941          ;-------------------------------;1D
U 055D, 0000,003E,0180,F800,0000,0010 ;25942        RETURN10                        ;UNDEFINED
                                    ;25943
                                    ;25944  ; 1E & 1F ARE SELECTED OUT BY SELPR
                                    ;25945
                                    ;25946  =100000 ;-------------------------------;20: RXCS
                                    ;25947          ID[RXCS]_D,                     ;CONSOLE RECEIVE CONTROL/STATUS
U 0560, 0000,003E,1180,3C00,0000,0018 ;25948        RETURN18
                                    ;25949
                                    ;25950          ;-------------------------------;21: RXDB
U 0561, 0000,003E,0180,F800,0000,0010 ;25951        RETURN10                        ;CANNOT WRITE RECEIVE DATA BUFFER
                                    ;25952
                                    ;25953          ;-------------------------------;22: TXCS
                                    ;25954          ID[TXCS]_D,                     ;CONSOLE TRANSMIT CONTROL/STATUS
U 0562, 0000,003E,1980,3C00,0000,0018 ;25955        RETURN18
                                    ;25956
                                    ;25957          ;-------------------------------;23: TXDB
                                    ;25958          ID[TXDB]_D,                     ;CONSOLE TRANSMIT DATA BUFFER
U 0563, 0000,003E,1D80,3C00,0000,0018 ;25959        RETURN18
```

```
                                    ;25960  ;CONTINUATION OF MTPR
                                    ;25961
                                    ;25962          ;-------------------------------;24 (AND OTHER ILLEGAL VALUES)
U 0564, 0000,003E,0180,F800,0000,0010 ;25963        RETURN10                        ;UNDEFINED
                                    ;25964
                                    ;25965  ; ***********************************************
                                    ;25966  ; * Patch no. 096, PCS 0564 trapped to WCS 1159 *
                                    ;25967  ; ***********************************************
                                    ;25968
                                    ;25969          ;-------------------------------;25
U 0565, 0000,003E,0180,F800,0000,0010 ;25970        RETURN10                        ;UNDEFINED
                                    ;25971
                                    ;25972  ; 26 & 27 ARE SELECTED OUT BY SELPR
                                    ;25973
                                    ;25974  =101000 ;-------------------------------;28: ACCS
                                    ;25975          ID[ACC.CS]_D,                   ;ACCELERATOR CONTROL/STATUS
U 0568, 0000,003E,5D80,3C00,0000,0018 ;25976        RETURN18                        ;
                                    ;25977
                                    ;25978          ;-------------------------------;29: ACRR
                                    ;25979          ID[ACC.0]_D,                    ;ACCELERATOR RESERVED REGISTER
U 0569, 0000,003E,5180,3C00,0000,0018 ;25980        RETURN18                        ;
                                    ;25981
                                    ;25982          ;-------------------------------;2A
U 056A, 0000,003E,0180,F800,0000,0010 ;25983        RETURN10                        ;UNDEFINED
                                    ;25984
                                    ;25985          ;-------------------------------;2B
U 056B, 0000,003E,0180,F800,0000,0010 ;25986        RETURN10                        ;UNDEFINED.
                                    ;25987
                                    ;25988          ;-------------------------------;2C: WCSA
                                    ;25989          ID[WCS.ADDR]_D,                 ;WRITABLE CONTROL STORE ADDRESS
U 056C, 0000,003E,8980,3C00,0000,0018 ;25990        RETURN18                        ;
                                    ;25991
                                    ;25992          ;-------------------------------;2D: WCSD
                                    ;25993          ID[WCS.DATA]_D,                 ;WRITABLE CONTROL STORE DATA
U 056D, 0000,003E,8D80,3C00,0000,0018 ;25994        RETURN18                        ;
                                    ;25995
                                    ;25996  ; 2E & 2F ARE SELECTED OUT BY SELPR
```

```
                                        ;25997  ;CONTINUATION OF MTPR
                                        ;25998
                                        ;25999  =110000 ;-------------------------------;30: SBIFS
                                        ;26000          ID[FAULT]_D,                    ;SBI FAULT/STATUS
U 0570, 0000,003E,6D80,3C  0,0018       ;26001          RETURN18
                                        ;26002
                                        ;26003          ;-------------------------------;31: SBIS
                                        ;26004          ID[SILO]_D,                     ;SBI SILO
U 0571, 0000,003E,6180,3C00,0000,0018   ;26005          RETURN18
                                        ;26006
                                        ;260C7          ;-------------------------------;32: SBISC
                                        ;26008          ID[COMP]_D,                     ;SBI SILO COMPARITOR
U 0572, 0000,003E,7180,3C00,0000,0018   ;26009          RETURN18
                                        ;26010
                                        ;26011          ;-------------------------------;33: SBIMT
                                        ;26012          ID[MAINT]_D,                    ;SBI MAINTENANCE
U 0573, 0000,003E,7580,3C00,0000,0018   ;26013          RETURN18
                                        ;26014
                                        ;26015          ;-------------------------------;34: SBIER
                                        ;26016          ID[SBI.ERR]_D,                  ;SBI ERROR REGISTER
U 0574, 0000,003E,6580,3C00,0000,0018   ;26017          RETURN18
                                        ;26018
                                        ;26019          ;-------------------------------;35: SBITA
U 0575, 0000,003E,0180,F800,0000,0010   ;26020          RETURN10                        ;READ ONLY...
                                        ;26021
                                        ;26022          ;-------------------------------;36: SBIQC
                                        ;26023          VA_D,                           ;QUAD CLEAR AT GIVEN ADDRESS
                                        ;26024          0_D,                            ;COPY ADDRESS FOR GROUPING MBZ'S
                                        ;26025          SC_K[.2],                       ;PREPARE TO SHIFT
U 0576, 0001,003C,09E0,F800,0284,6D78   ;26026          J/@PR.12
                                        ;26027
                                        ;26028  ; 37 IS SELECTED OUT BY SELPR
```

```
                                ;26029  ;CONTINUATION OF MTPR
                                ;26030
                                ;26031  =111000 ;-------------------------------;38: MME
                                ;26032          D_D.AND.K[.1],                  ;GET MEM MGMT BIT ALONE
                                ;26033          CLK.UBCC,                       ;NOTE WHETHER IT IS SET
                                ;26034          STOP.IB,                        ;DON'T LET IB INTERFERE
U 0578, 1819,0034,0580,F800,0010,0D72   ;26035          J/WPR.10
                                ;26036
                                ;26037          ;-------------------------------;39: TBIA
                                ;26038          ALU_K[.1],CLK.UBCC,             ;TRANSLATION BUFFER INVALIDATE ALL
                                ;26039          D_0,                            ;NO CHANGE TO TBER0
                                ;26040          STOP.IB,                        ;DON'T LET IB INTERFERE
U 0579, 1F18,0038,0580,F800,0010,0D72   ;26041          J/WPR.10
                                ;26042
                                ;26043          ;-------------------------------;3A: TBIS
                                ;26044          VA_D,                           ;TRANSLATION BUFFER INVALIDATE SINGLE
                                ;26045          D_0,
                                ;26046          INHIBIT.IB,                     ;HOLD OFF IB READS WHILE WRITING TB
U 057A, 0F01,003C,0180,0800,0200,0D76   ;26047          J/WPR.11
                                ;26048
                                ;26049          ;-------------------------------;3B
U 057B, 0000,003E,0180,F800,0000,0010   ;26050          RETURN10                        ;UNDEFINED
                                ;26051
                                ;26052          ;-------------------------------;3C: MBRK
                                ;26053          ID[UBREAK]_D,                   ;MICROBREAK
U 057C, 0000,003E,8580,3C00,0000,0018   ;26054          RETURN18
                                ;26055
                                ;26056          ;-------------------------------;3D: PMR
                                ;26057          RC[TO]_D.LEFT3,                 ;SHIFT BIT FROM 0 TO 3
                                ;26058          D_D.RIGHT,                      ;BIT 0 GONE SHOULD LEAVE 0
                                ;26059          Q_ID[CES],                      ;GET CPU ERROR/STATUS WHICH HAS PME
U 057D, 06A1,003C,31F0,2D80,0000,0D71   ;26060          J/WPR.9
                                ;26061
                                ;26062          ;-------------------------------;3E: SID
U 057E, 0000,003E,0180,F800,0000,0010   ;26063          RETURN10                        ;CAN'T WRITE SYSTEM ID, DUMMY
                                ;26064
                                ;26065  ; 3F AND ABOVE ARE SELECTED OUT BY SELPR
                                ;26066  =;END OF WPR CONSTRAINT BLOCK
```

```
                                ;26067  ;HERE TO LOAD PROCESS 1 BASE REGISTER (P1BR)
                                ;26068
                                ;26069          ;-------------------------------;
U 0D53, 0B00,003C,0180,F800,0000,0D54   ;26070  WPR.2A: D_D.SWAP                 ;MAKE 2**23
                                ;26071
                                ;26072          ;-------------------------------;
                                ;26073          D_D+Q,                          ;OFFSET BASE ADDRESS BY 2**23
U 0D54, 081D,0014,01E0,F800,0000,0D58   ;26074  Q_D                             ;SAVE OFFSET VALUE
                                ;26075
                                ;26076          ;-------------------------------;
                                ;26077          ALU_D.AND.K[.3],CLK.UBCC,       ;VERIFY LONGWORD ALIGNMENT
                                ;26078          D_D.SWAP,                       ;RE-ARRANGE FOR TEST OF SYS SPACE
                                ;26079          Q_Q.LEFT,                       ;MAKE 2**24 READY
U 0D58, 0B19,0D34,0DA8,F800,0010,0EF6   ;26080  D31?                            ;IS BASE IN SYSTEM HALF OF WORLD?
                                ;26081
                                ;26082  ;HERE FOR P0BR OR P1BR
                                ;26083
                                ;26084  =110    ;-------------------------------;D31=0
U 0EF6, 0000,003E,0180,F800,0006,0010   ;26085  WPR.2B: RETURN10                 ;BASE ADDRESS NOT IN SYSTEM SPACE
                                ;26086
                                ;26087          ;-------------------------------;D31=1
                                ;26088          ALU_D.AND.K[.4],CLK.UBCC,       ;VERIFY BIT 30=0
                                ;26089          D_D.SWAP,                       ;RESTORE CORRECT ORDER
U 0EF7, 0B19,0134,1180,F800,0010,08B8   ;26090  Z?,J/WPR.2D                      ;CHECK ZERONESS OF BITS 1-0
                                ;26091
                                ;26092  ;HERE FOR SBR
                                ;26093
                                ;26094          ;-------------------------------;
                                ;26095  WPR.2C: ALU_D.AND.K[.C],CLK.UBCC,       ;CHECK BITS 31-30 ZERO
                                ;26096          D_D.SWAP,                       ;RESTORE NORMAL BIT ORDER
                                ;26097          Q_0,                            ;CLEAR OFFSET
U 0D59, 0B19,0134,85F8,F800,0010,08B8   ;26098  Z?                              ;WERE BITS 1-0 ZERO?
                                ;26099
                                ;26100  =0     ;-------------------------------;ALU Z=0
U 08B8, 0000,003E,0180,F800,0000,0010   ;26101  WPR.2D: RETURN10                 ;BITS 1-0 WEREN'T ZERO.
                                ;26102
                                ;26103          ;-------------------------------;ALU Z=1
                                ;26104          D_D-Q,                          ;OFFSET BASE IF P1BR
U 08B9, 081D,0100,0180,F800,0000,08BC   ;26105  Z?                              ;TEST ZERONESS OF BIT 30 (AND 31 IF SBR)
                                ;26106
                                ;26107  =0     ;-------------------------------;ALU Z=0
U 08BC, 0000,003E,0180,F800,0000,0010   ;26108          RETURN10                ;BITS 31-30 WEREN'T VALID
                                ;26109
                                ;26110  ; ************************************************
                                ;26111  ; * Patch no. 038, PCS 08BC trapped to WCS 1172 *
                                ;26112  ; ************************************************
                                ;26113
                                ;26114  MO.PA.38:
                                ;26115          ;-------------------------------;ALU Z=1
U 08BD, 0058,0038,1DC0,F800,0000,0D5A   ;26116  Q_K[SC].RIGHT                    ;SHIFT THE REG # TO INDEX REG SPACE
                                ;26117
                                ;26118          ;-------------------------------;
                                ;26119          SC_Q.OR.K[.20],                 ;MAKE A REGISTER NUMBER INTO ID ADDRESS
U 0D5A, 0019,2030,7580,F800,0082,0542   ;26120  J/WPR.1                          ;GO WRITE IT
```

```
                                ;26121  ;HERE TO LOAD SEGMENT LENGTH REGISTERS
                                ;26122
                                ;26123          ;---------------------------------;
                                ;26124  WPR.3:   D_D.SWAP,                        ;BUILD 2**21
U 0D5B, 0B00,003C,6580,F800,0084,8D60   ;26125          SC_SC+K[.10]                    ;REG # + 10
                                ;26126
                                ;26127  ; ************************************************
                                ;26128  ; * Patch no. 037, PCS 0D5B trapped to WCS 1171 *
                                ;26129  ; ************************************************
                                ;26130
                                ;26131  MO.PA.37:
                                ;26132          ;---------------------------------;
                                ;26133          ALU_D-Q,CLK.UBCC,               ;VALIDATE LENGTH
                                ;26134          D_Q,                            ;LENGTH REGISTER TO D
                                ;26135          Q_D,                            ;2**21 TO Q
U 0D60, 0C1D,1700,01E0,F800,0010,0C3    ;26136          STATE1?                         ;TEST FOR P1LR
                                ;26137
                                ;26138  =1101   ;---------------------------------;
                                ;26139  WPR.3A: Q_K[SC].RIGHT,                  ; REG #/2 + 8
U 0C3D, 0058,0338,1DC0,F800,0000,0878   ;26140          C31?,J/WPR.3B                   ;TEST LENGTH VALID
                                ;26141
                                ;26142          ;---------------------------------;
U 0C3F, 081D,0014,0180,F800,0000,0C3D   ;26143          D_D+Q,J/WPR.3A                  ;OFFSET LENGTH FOR INTERNAL USE
                                ;26144
                                ;26145  =0*     ;---------------------------------;
U 0878, 0000,003E,0180,F800,0000,0010   ;26146  WPR.3B: RETURN10                        ;NEW VALUE TOO BIG
                                ;26147
                                ;26148          ;---------------------------------;
                                ;26149          SC_Q.OR.K[.30],                 ;SC HAS ID ADDRESS FOR THIS LENGTH REG
U 087A, 0019,2030,7980,F800,0082,0542   ;26150          J/WPR.1                         ;GO WRITE IT
                                ;26151
                                ;26152  ;HERE TO LOAD SCBB OR PCBB
                                ;26153
                                ;26154          ;---------------------------------;
                                ;26155  WPR.4:   ALU_D.AND.K[.C],                ;CHECK THAT BITS 31-30 ARE ZERO
                                ;26156          CLK.UBCC,                       ; WITH ALU Z
                                ;26157          SC_SC+K[.C],                    ;SC= REG # + C
                                ;26158          D_D.SWAP,                       ;RESTORE NORMAL BIT ORDER
U 0D61, 0B19,0134,8580,F800,0094,88C0   ;26159          Z?                              ;ARE BITS 1-0 ZERO?
                                ;26160  =0
                                ;26161  MO.PA.78:
                                ;26162          ;---------------------------------;ALU Z=0
U 08C0, 0000,003E,0180,F800,0000,0010   ;26163          RETURN10                        ;BITS 1-0 WEREN'T ZERO
                                ;26164
                                ;26165          ;---------------------------------;ALU Z=1
                                ;26166          SC_SC+K[.1E],                   ;SC=REG # + 2A
U 08C1, 0000,013C,5180,F800,0084,88CC   ;26167          Z?                              ;VERIFY THAT BITS 31-30 ARE ZERO
                                ;26168
                                ;26169  =0      ;---------------------------------;ALU Z=0
U 08CC, 0000,003E,0180,F800,0000,0010   ;26170          RETURN10                        ;BITS 31-30 AREN'T ZERO
                                ;26171
                                ;26172          ;---------------------------------;ALU Z=1
                                ;26173          ID(SC)_D,                       ;WRITE THE REGISTER
U 08CD, 0000,C 3E,0180,3400,0000,0018   ;26174          RETURN18                        ;TAKE GOOD RETURN
```

```
                                    ;26175  ;HERE FOR MTPR IPL
                                    ;26176
                                    ;26177          ;--------------------------------;
U 0D62, 0819,0034,8D80,F800,0000,0D64   ;26178  WPR.5:  D_D.AND.K[.1F]                    ;STRIP IPL
                                    ;26179
                                    ;26180          ;--------------------------------;
U 0D64, 0D00,003C,0180,F800,0000,0D65   ;26181          D_DAL.SC                        ;ALIGN NEW IPL TO BITS 20-16
                                    ;26182
                                    ;26183          ;--------------------------------;
                                    ;26184          Q_D,                            ;SAVE ALIGNED IPL
U 0D65, 0818,0038,8DE0,F800,0000,0D66   ;26185          D_K[.1F]                        ;GET MASK FOR IPL
                                    ;26186
                                    ;26187          ;--------------------------------;
                                    ;26188          RC[TO]_Q,                       ;SAVE ALIGNED IPL
                                    ;26189          D_DAL.SC,                       ;ALIGN THE MASK TO 20-16
U 0D66, 0D01,203C,3DF0,2D80,0000,0D68   ;26190          Q_ID[PSL]                       ;GET CURRENT PSL
                                    ;26191
                                    ;26192          ;--------------------------------;
                                    ;26193          Q_Q.ANDNOT.D,                   ;MASK OUT OLD IPL
U 0D68, 001D,2024,61C0,F800,0084,6D69   ;26194          SC_K[.F]                        ;PUT PSL ADDRESS IN SC
                                    ;26195
                                    ;26196          ;--------------------------------;
                                    ;26197          D_Q.OR.RC[TO],                  ;COMBINE NEW IPL WITH OLD PSL
U 0D69, 0811,2030,0180,F900,0000,0542   ;26198          J7WPR.1                         ;GO WRITE OUT AGAIN
```

```
                                ;26199  ;HERE TO SET AST LEVEL
                                ;26200
                                ;26201          ;--------------------------------;
                                ;26202  WPR.6A: ALU_D-K[.4]-1,                  ;ASYNCRONOUS SYSTEM TRAP REGISTER
U 0D6A, 0019,0008,1180,F800,0010,0D6B   ;26203          CLK.UBCC                ;VALIDATE VALUE
                                ;26204
                                ;26205          ;--------------------------------;
                                ;26206          Q_ID[CES],                      ;CPU ERROR/STATUS CONTAINS ASTLVL
                                ;26207          RC[TO]_D.0XT[BYTE],             ;SAVE NEW AST LEVEL
U 0D6B, 0003,833C,31F0,2D80,0000,087C   ;26208          C31?                    ;CHECK THAT IT IS VALID
                                ;26209
                                ;26210  =0*     ;--------------------------------;
                                ;26211          D_Q.ANDNOT.K[.7],               ;STRIP OUT OLD AST LVL
U 087C, 0819,2024,5D80,F800,0000,0F05   ;26212          J7WPR.9A                ;GO PLUG IN NEW
                                ;26213
                                ;26214          ;--------------------------------;
U 087E, 0000,003E,0180,F800,0000,0010   ;26215          RETURN10                ;NOT VALID GIVES AN ERROR
                                ;26216
                                ;26217  ;HERE TO POST A SOFTWARE INTERRUPT REQUEST
                                ;26218
                                ;26219          ;--------------------------------;
                                ;26220  WPR.7:  D_Q.ORNOT.MASK,                 ;SET REQUESTED INTERRUPT BIT
U 0D70, 0801,201C,0180,F800,0000,0555   ;26221          J7WPR.7A                ;GO WRITE IT BACK
                                ;26222
                                ;26223  ;HERE TO WRITE INTERVAL CLOCK CONTROL/STATUS
                                ;26224
                                ;26225  =110    ;--------------------------------;D31=0, NOT SETTING ERROR BIT
                                ;26226  WPR.8:  ID[CLK.CS]_D,                   ;INTERVAL CLOCK CONTROL/STATUS
U 0EFE, 0000,003E,2980,3C00,0000,0018   ;26227          RETURN18
                                ;26228
                                ;26229          ;--------------------------------;D31=1, COPY TO ERROR BIT
                                ;26230          D_D.OR.K[.8000],                ;SET REAL ERROR BIT (15)
U 0EFF, 0819,0030,4580,F800,0000,0EFE   ;26231          J7WPR.8
                                ;26232
                                ;26233  ;HERE TO WRITE PERFORMANCE MONITOR ENABLE
                                ;26234
                                ;26235          ;--------------------------------;
                                ;26236  WPR.9:  D_Q.ANDNOT.K[.8],               ;CLEAR OLD PME
U 0D71, 0819,2D24,0180,F800,0000,0F05   ;26237          D.NE.0?                 ;TEST THAT ONLY BIT0 SET
                                ;26238
                                ;26239  =101    ;--------------------------------;D .EQL. 0
                                ;26240  WPR.9A: D_D.OR.RC[TO],                  ;PLUG IN NEW VALUE
                                ;26241          SC_K[.C],                       ;SETUP ID ADDRESS OF CES REGISTER
U 0F05, 0811,0030,8580,F900,0084,6542   ;26242          J/WPR.1                 ;GO WRITE IT
                                ;26243
                                ;26244          ;--------------------------------;D .NEQ. 0
U 0F07, 0000,003E,0180,F800,0000,0010   ;26245          RETURN10                ;TRIED TO WRITE DATA .GTRU. 1
```

```
                                ;26246  ;HERE TO INVALIDATE ENTIRE TB OR CHANGE MEM MGMT ENABLE
                                ;26247
                                ;26248          ;-------------------------------;
                                ;26249  WPR.10: Q_ID[TBER0],                    ;GET TB STATUS
U 0D72, 0000,013C,49F0,2C00,0000,08D0   ;26250          Z?
                                ;26251
                                ;26252  =0      ;-------------------------------;ALU Z=0, MME BEING SET
                                ;26253  WPR.10A:FE_K[.40],                     ;LOOP COUNT FOR INVALIDATE ALL ADDRESSES
U 08D0, 0000,003C,3180,F800,0104,6D73   ;26254          J/WPR.10B
                                ;26255
                                ;26256          ;-------------------------------;ALU Z=1, MME BEING CLEARED
                                ;26257          D_Q.ANDNOT.K[.1],               ;CLEAR MME IN TBER0
U 08D1, 0819,2024,0580,F800,0000,03A8   ;26258          J/WPR.10D               ;GO WRITE IT BACK
                                ;26259
                                ;26260          ;-------------------------------;
                                ;26261  WPR.10B:SC_K[.14],                     ;GET 20 DECIMAL
                                ;26262          D_D.OR.Q,                       ;SET MME IF DESIRED
U 0D73, 081D,0030,2180,F988,0084,61EC   ;26263          RC[T1]_ALU              ;SAVE DESIRED FINAL VALUE OF TBER0
                                ;26264
                                ;26265  =01***  ;-------------------------------;
                                ;26266          D_D.ORNOT.MASK,                 ;SET WRITE-BOTH BIT
                                ;26267          SC_K[.9],                       ;READY TO BUILD CONSTANT 512
U 01EC, 0801,001D,D980,F800,0084,63A8   ;26268          CALL,J/WPR.10D          ;SETUP TBER0 FOR LOOP
                                ;26269
                                ;26270          ;-------------------------------;
                                ;26271          Q_K[.8000].RIGHT,               ;SETUP BIT 14 AS STARTING ADDRESS
                                ;26272          VA_ALU,                         ;INITIAL TB INDEX IS ZERO
                                ;26273          SC_FE,                          ;GET LOOP COUNT INTO SC
                                ;26274          STOP.IB,INHIBIT.IB,             ;PREVENT INSTRUCTION BUFFER CYCLES
                                ;26275          D_0,                            ;PREPARE TO WRITE ZEROS TO TB
U 01FC, 1F58,0038,45C0,0A78,0281,0D74   ;26276          LAB_R[R15]              ;GET 512 INTO LATCH
                                ;26277
                                ;26278          ;-------------------------------;
                                ;26279  WPR.10C:ID[TBUF]_D,                     ;CLEAR NEXT PAIR OF TB ENTRIES
                                ;26280          D_RC[T1],                       ;GET READY TO RESTORE TBER0 IF END
U 0D74, 0810,0C38,4180,3D08,0000,03A8   ;26281          MUL?                    ;CHECK SC FOR COMPLETION
                                ;26282
                                ;26283  =0**    ;-------------------------------;SC.EQL.0, DONE
                                ;26284  WPR.10D:ID[TBER0]_D,                    ;LEAVE CORRECT VALUE IN TBER0
                                ;26285          R[R15]_NOT.MASK,                ;BUILD CONSTANT IN R15
                                ;26286          START.IB,                       ;TURN IB BACK ON
U 03A8, 3003,001E,4980,3EF8,0000,0018   ;26287          RETURN18
                                ;26288
                                ;26289          ;-------------------------------;SC.NEQ.0, MORE TO INVALIDATE
                                ;26290          VA_Q-LB,                        ;UPDATE ADDRESS IN VA
                                ;26291          Q_ALU,                          ;READY UPDATED ADDRESS FOR NEXT
                                ;26292          D_0,                            ;CLEAR D FOR WRITE TO TBUF
                                ;26293          SC_SC-K[.1],                    ;DECREMENT LOOP COUNT
                                ;26294          INHIBIT.IB,                     ;PREVENT IB FROM MESSING TB ADDRESS
U 03AC, 0F0D,2000,05C0,0800,0284,AD74   ;26295          J/WPR.10C
                                ;26296
                                ;26297  ;HERE TO INVALIDATE ONE TB ENTRY
                                ;26298
                                ;26299          ;-------------------------------;
U 0D76, 0000,003E,4180,3C00,0000,0018   ;26300  WPR.11: ID[TBUF]_D,RETURN18       ;CLEAR THE ENTRY SELECTED BY VA
```

```
                                    ;26301  ;HERE TO PERFORM A QUAD CLEAR TO REMOVE DOUBLE ERRORS IN MOS MEMORY
                                    ;26302
                                    ;26303          ;-------------------------------;
U 0D78, 0D00,003C,0180,F800,0000,0D79   ;26304  WPR.12: D_DAL.SC                      ;GET ALL MBZ'S INTO LOW BITS OF D
                                    ;26305
                                    ;26306          ;-------------------------------;
U 0D79, 0019,0034,8D80,F800,0010,0D7A   ;26307          ALU_D.AND.K[.1F],CLK.UBCC       ;TEST MBZ'S
                                    ;26308
                                    ;26309          ;-------------------------------;
                                    ;26310          D_0,                            ;CLEAR WITH ZEROS
U 0D7A, 0F00,013C,0180,F800,0000,08D4   ;26311          Z?
                                    ;26312
                                    ;26313  =0      ;-------------------------------;ALU Z=0, MBZ BITS WERE SET
U 08D4, 0000,003E,0180,F800,0000,0010   ;26314          RETURN10
                                    ;26315
                                    ;26316          ;-------------------------------;ALU Z=1, MBZ WERE CLEAR
                                    ;26317          MCT/EXTWRITE.P,                 ;PERFORM EXTENDED WRITE
U 08D5, 0000,003E,0180,A000,0000,0018   ;26318          LONG,RETURN18
```

```
                                ;26319  ;HERE IS SUBROUTINE TO SELECT A PROCESSOR REGISTER
                                ;26320
                                ;26321          ;-------------------------------;
                                ;26322  SELPR:  ALU_D.ANDNOT.K[.3F],          ;RANGE CHECK PROCESSOR REGISTER NUMBER
                                ;26323          CLK.UBCC,                     ;SET Z IF REG # .LEQU. 3F
                                ;26324          SC_SC+FE,                     ;RESTORE REG # IN SC
                                ;26325          FE_EALU,                      ; AND FE
U 0D7B, 0019,0324,5580,F800,0190,888C   ;26326          C3T?                          ;IS THIS A STACK POINTER?
                                ;26327
                                ;26328  =0*     ;-------------------------------;ALU C=0, REG # .LSSU. 8
                                ;26329          RC[TO]_Q,                     ;SAVE REGISTER DATA
                                ;26330          Q_ID[PSL],                    ;GET PSL FOR COMPARING MODE
U 088C, 0001,203C,3DF0,2D80,0000,0D81   ;26331          J7SELP.3                      ;GO CHECK MODE AGAINST REG #
                                ;26332
                                ;26333          ;-------------------------------;ALU C=1, REG # .GEQ. 8
                                ;26334          D_Q,                          ;SAVE OTHER OPERAND IN D
                                ;26335          ALU_NOT.RC[TO],               ;REG # BITS 2-1 TO ALU 1-0
                                ;26336          Q_ID[USTACK],                 ;GET RETURN ADDRESS FROM USTACK
U 088E, 0C13,011C,81F0,2D00,0000,08D8   ;26337          Z?                            ;TEST FOR REG # IN RANGE
                                ;26338
                                ;26339  =0      ;-------------------------------;ALU Z=0, REG # .GTRU. 3F
                                ;26340          D_Q.OR.K[.24],                ;FORCE ILLEGAL NUMBER
                                ;26341          Q_D,                          ;KEEP OTHER OPERAND OUT OF WAY
U 08D8, 0819,2030,E9E0,F800,0000,0C6F   ;26342          J7SELP.2                      ;RETURN TO CALLER
                                ;26343
                                ;26344          ;-------------------------------;ALU Z=1, REG # OK
                                ;26345          EALU_SC-K[.34],CLK.UBCC,      ;READY CHECK TO ALLOW 36, 3E
                                ;26346          D_Q,Q_D,                      ;RETURN BASE TO D, SAVE DATA IN Q
U 08D9, 0C00,153C,29E0,F800,0014,AC4B   ;26347          ALU1-0?                       ;TEST FOR REG #  OF FORM ***11*
                                ;26348
                                ;26349  =1011   ;-------------------------------;ALU 1-0 .NEQ. 0
                                ;26350          D_D.OR.K[SC],                 ;RETURN TO CALL PLUS REG #
U 0C4B, 0819,0030,1D80,F800,0000,0C6F   ;26351          J7SELP.2                      ;KEEP THE DATA SAFE
                                ;26352
                                ;26353          ;-------------------------------;ALU1-0 .EQL. 0 (REG # WAS ***11*)
                                ;26354          STATE_FE,                     ;REG # TO STATE FOR TEST OF EVENNESS
                                ;26355          D_D.OR.K[.24],                ;FORCE RETURN TO ILLEGAL REG #
U 0C4F, 0819,123C,E980,F800,1400,6C67   ;26356          EALU.N?                       ;IS REG # .LSS. 34?
                                ;26357
                                ;26358  =0110   ;CREATE CONSTRAINT BLOCK FOR TRICKY BRANCHING
                                ;26359  =0111   ;-------------------------------;EALU N=0 (REG # .GEQ. 34)
U 0C67, 0000,173C,0180,F800,0000,0C6E   ;26360          STATE0?                       ;IS REG # ALSO EVEN?
                                ;26361
                                ;26362  =1110   ;-------------------------------;REG # IS EVEN AND GTR 34
U 0C6E, 0819,0030,1D80,F800,0000,0C6F   ;26363          D_D.OR.K[SC]                  ;ALLOW RETURNS TO 36 AND 3E
                                ;26364
                                ;26365  =1111   ;-------------------------------;
                                ;26366  SELP.2: ID[USTACK]_D,                 ;SETUP RETURN ADDRESS
U 0C6F, 0C00,003C,8180,3C00,0000,0D80   ;26367          D_Q                           ;RETURN DATA IN D
                                ;26368
                                ;26369          ;-------------------------------;
U 0D80, 0000,003E,0180,F800,0000,0000   ;26370          RETURN0                       ;GO TO CALL SITE + REG #
```

```
                                        ;26371  ;HERE WHEN SELECTED PROCESSOR REGISTER IS A STACK POINTER.
                                        ;26372  ; CHECK TO SEE WHETHER IT IS THE CURRENT STACK POINTER
                                        ;26373
                                        ;26374          ;--------------------------------;
                                        ;26375  SELP.3: ALU_D-K[.4]-1,CLK.UBCC,          ;MAKE SURE REG # IS LESS THAN 5
U 0D81, 0C19,0008,1180,F800,0010,0D82   ;26376          D_Q                             ;MODE NOW TO D REGISTER
                                        ;26377
                                        ;26378          ;--------------------------------;
                                        ;26379          D_D.SWAP,                       ;CURRENT MODE AND IS BIT TO D<2:0>
                                        ;26380          FE_SC+K[.28],                   ;SETUP ID BUS ADDRESS OF SELECTED SP
U 0D82, 0B00,033C,2D80,F800,0104,88C4   ;26381          C3T?                            ;LEGAL REGISTER NUMBER?
                                        ;26382
                                        ;26383  =0*     ;--------------------------------;
                                        ;26384          D_D.AND.K[.7],                  ;STRIP TO JUST IS AND MODE
U 08C4, 0819,0034,5D80,F800,0000,0D83   ;26385          J/SELP.4
                                        ;26386
                                        ;26387          ;--------------------------------;ALU C31=1 (REGISTER NUMBER ILLEGAL)
U 08C6, 0000,003E,0180,F800,0000,0024   ;26388          RETURN24                        ;TAKE ERROR RETURN
                                        ;26389
                                        ;26390          ;--------------------------------;
                                        ;26391  SELP.4: ALU_D.XOR.K[SC],CLK.UBCC,       ;IS REGISTER NUMBER = CURRENT MODE?
U 0D83, 0019,0020,1D80,F800,0091,0D84   ;26392          SC_FE                           ;SETUP ID BUS ADDRESS OF REGISTER
                                        ;26393
                                        ;26394          ;--------------------------------;
                                        ;26395          D_RC[T0],                       ;GET BACK THE DATA
U 0D84, 0810,013A,0180,F900,0000,0002   ;26396          Z?,RETURN2                      ;RETURN 2, OR 3 IF CURRENT SP SELECTED
                                        ;26397
                                        ;26398  ;SUBROUTINE TO CALL (WITH PSL.MODE?) TO VALIDATE THAT PROCESSOR IS IN
                                        ;26399  ; KERNEL MODE.  RETURN20 VERIFIES KERNEL, NO RETURN IF NOT
                                        ;26400
                                        ;26401  =11110  ;--------------------------------;SOMEBODY MADE A NO-NO
                                        ;26402  KERNEL.CHK:
U 003E, 0018,0038,6580,F980,0000,08FC   ;26403          RC[T0]_K[.10],J/EXCPT           ;TAKE RESERVED OPERATION EXCEPTION
                                        ;26404
                                        ;26405          ;--------------------------------;OK
U 003F, 0000,003E,0180,F800,0000,0020   ;26406          RETURN20
                                        ;26407
                                        ;26408  .LIST           ;Re-enable full listing
```

```
                                        ;26409  .TOC    "11MODE.MIC"
                                        ;26410  .TOC    "Revision 1.0"
                                        ;26411  ;        P. R. Guilbault
                                        ;26412
;26413  .NOBIN
;26414  .CHANGE/NATIVE=0
;26415  .TOC     ""      Revision History"
;26416
;26417  ; 01    Change macro names that deal with conditions codes.
;26418  ; 00    Start of history
;26419
                                        ;26420  .BIN
                                        ;26421  .NOLIST          ;Disable listing of PCS code for quickie assemblies
```

```
;26422  .TOC    ""      Compatibility mode    : Description"
;26423
;26424  ;THE 11-INSTUCTIONS ARE DIVIDED INTO THE FOLLOWING CLASSES:
;26425
;26426  ;       A-CLASS:CLR,CLRB,COM,COMB,DEC,DECB,ADC,ADCB,
;26427  ;               SBC,SBCB,BIC,BICB,BIS,BISB,ADD,SUB,XOR,ASL,ASLB
;26428  ;               ROL,ROLB
;26429  ;       J-CLASS: JMP,JSR
;26430  ;       BIN:    BIC,BICB,BIS,BISB,ADD,SUB,XOR
;26431  ;       C-CLASS: A-CLASS.AND.(.NOT.BIN)
;26432  ;       MFP:    MFPI,MFPD
;26433  ;       MTD:    MTPI,MTPD
;26434
```

```
                                    ;26435  .TOC    "       Compatibility mode    : DP0 -  BASIC"
                                    ;26436
                                    ;26437  ;DP0 - DECISION POINT 0
                                    ;26438  ;HERE OFF IRD WHEN THERE IS A TRAP, INTERRUPT, OR CONSOLE REQUEST UP
                                    ;26439
                                    ;26440  ;720:   ;-------------------------------;
                                    ;26441  ;       J/720
                                    ;26442
                                    ;26443  724:    ;-------------------------------;
U 0724, 0000,003C,0180,F800,0000,0024 ;26444        J/TRACE                         ;TRACE TRAP PENDING
                                    ;26445
                                    ;26446  ;728:   ;-------------------------------;
                                    ;26447  ;       J/728
                                    ;26448
                                    ;26449  ;72C:   ;-------------------------------;
                                    ;26450  ;       J/72C
                                    ;26451
                                    ;26452  730:    ;-------------------------------;
U 0730, 0000,003C,0180,F800,0000,0F8F ;26453        J/INTRPT                        ;INTERNAL INTERRUPT REQUEST
                                    ;26454
                                    ;26455  734:    ;-------------------------------;HALT PENDING
U 0734, 0814,0038,0180,F800,0000,0D85 ;26456        D_PC                            ;UNDO EXTRA PC ADVANCE
                                    ;26457
                                    ;26458          ;-------------------------------;
                                    ;26459          PC&VA_D-K[.2],                  ;SET PC TO NEXT INSTRUCTION TO DO
U 0D85, 0019,0000,0980,F801,0200,0439 ;26460        J/CONS.HALT                     ;GO TO HALT LOOP
                                    ;26461
                                    ;26462  738:    ;-------------------------------;
U 0738, 0000,003C,0180,F800,0000,0F8D ;26463        J/INTIO                         ;SBI INTERRUPT REQUEST
                                    ;26464
                                    ;26465  73C:    ;-------------------------------;
U 073C, 0000,003C,0180,F800,0000,003C ;26466        J/ARITHT                        ;ARITHMETIC TRAP
                                    ;26467
                                    ;26468          ;-------------------------------;
                                    ;26469  GO.IRD: LOAD.IB.11,PC_PC+2,WORD,        ;LOAD I-BUFFER, UPDATE PC
U 0D86, 0000,403C,0180,6005,4000,0762 ;26470        INTRPT.STROBE,J/IRD11           ;STROBE INTERRUPTS, FETCH NEXT INSTR.
                                    ;26471
                                    ;26472  760:    ;-------------------------------;
U 0760, 0000,003D,0180,F800,0000,0E64 ;26473        CALL,J/IB.TBM                   ;STOPPED WITH IB MISS, GOTO FILL IB
                                    ;26474
                                    ;26475  761:    ;-------------------------------;
U 0761, 0000,003D,0180,F800,0000,0B80 ;26476        CALL,J/IB.ERR                   ;STOPPED WITH AN ERROR
                                    ;26477
                                    ;26478  762:    ;-------------------------------;
U 0762, 0804,003B,65F0,F857,1284,6700 ;26479  IRD11: IRD.11                          ;INSTR DECODING, ALSO IB STALL STATE
```

```
                                    ;26480  ;MISCELLANEOUS INSTRUCTIONS EXECUTES:
                                    ;26481  ;ENTER HERE FROM IRD,
                                    ;26482  ;WITH LA GETS DST,
                                    ;26483  ;LB GETS SRC,
                                    ;26484  ;VA AND D GET SRC OR PC (LB OR PC),
                                    ;26485  ;Q GETS IB DATA BYTES 2 AND 3,
                                    ;26486  ;PC IS ADDED BY N WHERE N IS DETERMINED BY IB,
                                    ;26487  ;SC IS SET TO 16.
                                    ;26488
                                    ;26489  754:    ;-------------------------------;
                                    ;26490  CLRCC:  ALU_Q.AND.K[.1E],               ;CLEAR COND CODES INSTRUCTIONS
U 0754, 0859,2034,5180,F800,0000,0D88 ;26491        D_ALU.RIGHT,SI/ZERO             ;GET BITS 3-0 OF INSTR IN D
                                    ;26492
                                    ;26493          ;-------------------------------;
U 0D88, 0000,003C,3DF0,2C00,0000,0D89 ;26494        Q_ID[PSL]                       ;GET PSL TO CLEAR CONDITION CODES
                                    ;26495
                                    ;26496          ;-------------------------------;
U 0D89, 081D,2024,0180,F800,0000,0D8A ;26497        D_Q.ANDNOT.D                    ;CLEAR RESPECTIVE COND CODES
                                    ;26498
                                    ;26499          ;-------------------------------;
                                    ;26500  STPSL:  ID[PSL]_D,                      ;STORE PSL
U 0D8A, 4000,003C,3D80,3C05,4000,0762 ;26501        CLR.IB0-1,PC_PC+2,J/IRD11       ;GO ON TO NEXT INSTR
                                    ;26502
                                    ;26503  725:    ;-------------------------------;
                                    ;26504  SETCC:  ALU_Q.AND.K[.1E],               ;SET COND CODES INSTRUCTIONS
U 0725, 0859,2034,5180,F800,0000,0D8B ;26505        D_ALU.RIGHT,SI/ZERO             ;SET BITS 3-0 OF INSTR IN D
                                    ;26506
                                    ;26507          ;-------------------------------;
U 0D8B, 0000,003C,3DF0,2C00,0000,0D8C ;26508        Q_ID[PSL]                       ;GET PSL TO SET CONDITION CODES
                                    ;26509
                                    ;26510          ;-------------------------------;
U 0D8C, 081D,0030,0180,F800,0000,0D8A ;26511        D_D[OR]Q,J/STPSL                ;SET CC, GOTO STORE PSL
                                    ;26512
                                    ;26513  7A0:    ;-------------------------------;
                                    ;26514  BXXOK:                                  ;BRANCH INSTRUCTIONS: BRANCH
                                    ;26515          PC&VA_Q.SXT[WORD]+PC,           ;ADD DISPLACEMENT TO GET NEW PC
U 07A0, 2016,6014,0180,F801,4200,0D86 ;26516        FLUSH.IB,J/GO.IRD               ;FLUSH IB, GO TO RELOAD IT
                                    ;26517
                                    ;26518  7A5:    ;-------------------------------;
                                    ;26519          PC&VA_Q.SXT[WORD]+PC,           ;BRANCH INSTRUCTIONS: BRANCH
U 07A5, 2016,6014,0180,F801,4200,0D86 ;26520        FLUSH.IB,J/GO.IRD
                                    ;26521
                                    ;26522  7A6:    ;-------------------------------;
                                    ;26523          PC&VA_Q.SXT[WORD]+PC,           ;BRANCH INSTRUCTIONS: BRANCH
U 07A6, 2016,6014,0180,F801,4200,0D86 ;26524        FLUSH.IB,J/GO.IRD
                                    ;26525
                                    ;26526
                                    ;26527  7A1:    ;-------------------------------;
U 07A1, 4000,003C,0180,F805,4000,0762 ;26528  BXX:   CLR.IB0-1,PC_PC+2,J/IRD11       ;BRANCH INSTRUCTIONS: NO BRANCH
                                    ;26529
                                    ;26530  7A2:    ;-------------------------------;
U 07A2, 4000,003C,0180,F805,4000,0762 ;26531        CLR.IB0-1,PC_PC+2,J/IRD11       ;BRANCH INSTRUCTIONS: NO BRANCH
                                    ;26532
                                    ;26533  7A4:    ;-------------------------------;
U 07A4, 4000,003C,0180,F805,4000,0762 ;26534        CLR.IB0-1,PC_PC+2,J/IRD11       ;BRANCH INSTRUCTIONS: NO BRANCH
```

```
                                    ;26535  ;HERE FOR RTI/RTT, WHICH ARE TREATED IDENTICALLY
                                    ;26536
                                    ;26537  750:    ;-------------------------------;
                                    ;26538  RTIRTT: VA_R[R6],                       ;GET STACK POINTER VIA LA
U 0750, 0000,003C,3DF0,2E30,0200,0D8D ;26539        Q_ID[PSL]                       ;GET PSL (NEED BITS 31-16)
                                    ;26540
                                    ;26541          ;-------------------------------;
                                    ;26542          D[WORD]_CACHE,                  ;GET (SP)
U 0D8D, 0019,6024,C1C0,4000,0000,0D8E ;26543        Q_Q.ANDNOT.K[.FFFF]             ;KEEP BITS 31-16 OF PSL
                                    ;26544
                                    ;26545          ;-------------------------------;
                                    ;26546          PC&VA_D.OXT[WORD],              ;GET NEW PC FROM STACK
U 0D8E, 2003,403C,0180,F801,4200,0D90 ;26547        FLUSH.IB
                                    ;26548
                                    ;26549          ;-------------------------------;
U 0D90, 0000,003C,0180,6000,0000,0D91 ;26550        LOAD.IB.11                      ;RELOAD INSTRUCTION BUFFER
                                    ;26551
                                    ;26552          ;-------------------------------;
U 0D91, 0018,0014,0980,F800,0200,0D92 ;26553        VA_LA+K[.2]                     ; VA_R[R6]+2
                                    ;26554
                                    ;26555          ;-------------------------------;
                                    ;26556          D[WORD]_CACHE,                  ;GET PSW FROM STACK
U 0D92, 0018,4018,1180,42B0,0000,0D93 ;26557        R6_LA+K[.4].RLOG                ;UPDATE STACK POINTER
                                    ;26558
                                    ;26559          ;-------------------------------;
U 0D93, 0819,0034,8D80,F800,0000,0D94 ;26560        D_D.AND.K[.1F]                  ;IGNORE PSW EXCEPT FOR T, N, Z, V, C
                                    ;26561
                                    ;26562          ;-------------------------------;
U 0D94, 081F,4030,0180,F800,0000,0D96 ;26563        D_D.OXT[WORD].OR.Q              ;COMBINE PSL BITS 31-16 WITH TNZVC
                                    ;26564
                                    ;26565          ;-------------------------------;
                                    ;26566          ID[PSL]_D,                      ;STORE NEW PSL
U 0D96, 0000,003C,3D80,3C05,4000,0762 ;26567        PC_PC+2,INTRPT.STROBE,J/IRD11   ;UPDATE PC, STROBE INTERRUPTS, FETCH
```

```
                                    ;26568  ;HERE FOR RTS
                                    ;26569
                                    ;26570  7AB:    ;--------------------------------;
U 07AB, 0000,003C,0180,FA30,0200,0D98 ;26571  RTS:    VA_R[R6]                        ;GET STACK POINTER VIA LA
                                    ;26572
                                    ;26573          ;--------------------------------;
                                    ;26574          D[WORD]_CACHE,                  ;GET SAVED REGISTER FROM STACK
                                    ;26575          R6_LA+K[.2].RLOG,               ;POP STACK POINTER
U 0D98, 0018,4918,0980,42B0,0000,0A72 ;26576          PC.MODES?                       ;IS REGISTER=PC?
                                    ;26577
                                    ;26578  =10     ;--------------------------------;DST R .NEQ. PC
                                    ;26579          LA_R(DST)&LB_R(SRC),WORD,       ;NO:    NOT PC
U 0A72, 0000,403C,0180,F850,0000,0D9? ;26580          J/RTS.L1
                                    ;26581
                                    ;26582          ;--------------------------------;DST R .EQL. PC
                                    ;26583          PC&VA_D.OXT[WORD],              ;RESTORE PC SAVED ON STACK
U 0A73, 2003,403C,0180,F801,4200,0D86 ;26584          FLUSH.IB,J/GO.IRD               ;FETCH NEXT INSTRUCTION
                                    ;26585
                                    ;26586          ;--------------------------------;
U 0D99, 0001,403C,0180,F8C8,0000,0D9A ;26587  RTS.L1: R(DST)_D,WORD                    ;RESTORE REGISTER SAVED ON STACK
                                    ;26588
                                    ;26589          ;--------------------------------;
                                    ;26590          PC&VA_ALU,ALU_LA.AND.K[.FFFF],  ;RESTORE PC FROM OLD VALUE OF REGISTER
U 0D9A, 2018,0034,C180,F801,4200,0D86 ;26591          FLUSH.IB,J/GO.IRD               ;GO TO RELAOD IB AND FETCH
                                    ;26592
                                    ;26593
                                    ;26594  7AA:    ;--------------------------------;
U 07AA, 0019,4004,0580,F8C0,0010,0D9B ;26595  ?0811:  R(SRC)_D-K[.1].RLOG,CLK.UBCC     ;R(SRC) GETS SRC-1
                                    ;26596
                                    ;26597          ;--------------------------------;
                                    ;26598          Q_Q.AND.K[.7E],                 ;GET BRANCH OFFSET
U 0D9B, 4019,2134,F9C0,F800,4000,08DC ;26599          C[R.IB0-1,Z?                    ;DID SRC REG GO TO ZERO?
                                    ;26600                                          ;BRANCH ON ALU Z-BIT
                                    ;26601  =0      ;--------------------------------;
U 08DC, 001F,0000,01C0,F800,0000,07A0 ;26602          Q_0-Q,J/BXXOK                   ;Z = 0: BRANCH -- PC_PC-OFFSET
                                    ;26603
                                    ;26604          ;--------------------------------;
U 08DD, 0000,003C,0180,F805,0000,0762 ;26605          PC_PC+2,J/IRD11                 ;Z = 1: NEXT INSTRUCTION
```

```
                                        ;26606  .TOC    ''       Compatibility mode    : DP0 -  [SM0]*DM0''
                                        ;26607
                                        ;26608  ;REGISTER EXECUTES FOR 1 OPERAND INSTRUCTIONS, REGISTER TO REGISTER
                                        ;26609  ;  EXECUTES FOR 2 OPERAND INSTRUCTIONS.
                                        ;26610  ;ENTER HERE FROM IPD,
                                        ;26611  ;  WITH LA GETS DST,
                                        ;26612  ;  LB GETS SRC,
                                        ;26613  ;  VA AND D GET SRC OR PC (LB OR PC),
                                        ;26614  ;  Q GETS IB DATA BYTES 2 AND 3,
                                        ;26615  ;  PC IS ADDED BY N WHERE N IS DETERMINED BY IB,
                                        ;26616  ;  SC IS SET TO 16.
                                        ;26617
                                        ;26618  753:    ;-------------------------------;
                                        ;26619  ACLASS: ALU_LA[INST.DEP]D,R(DST)_ALU,   ;A/CLASS + MOV, DST IS NOT PC
                                        ;26620          SET.CC(INST),                   ;CLOCK PSW CONDITION CODES
                                        ;26621          CLR.IB0-1,                      ;CLEAR FIRST 2BYTES OF I-BUFFER
U 0753, 401C,E00C,0180,F8CD,4070,0762   ;26622          PC_PC+2,J/IRD11                 ;INCREMENT PC, FETCH NEXT INSTRUCTION
                                        ;26623
                                        ;26624  7D3:    ;-------------------------------;A-CLASS, DEST IS PC
U 07D3, 0814,0038,01E0,F800,0000,0DA0   ;26625  ACLPC:  Q_D,D_PC                         ;DST = PC, SO LOAD PC IN D
                                        ;26626
                                        ;26627          ;-------------------------------;
                                        ;26628          PC&VA_ALU,ALU_D[INST.DEP]Q,     ;PERFORM OPERATION ON PC
                                        ;26629          SET.CC(INST),                   ;SET CONDITION CODES
U 0DA0, 201D,C00C,0180,F801,4270,0D86   ;26630          FLUSH.IB,J/GO.IRD               ;NEW PC, SO FLUSH I-BUFFER
                                        ;26631
                                        ;26632
                                        ;26633  7D6:    ;-------------------------------;
                                        ;26634  NEG.B:  D_0,Q_LA,                       ;Q GETS DESTINATION
U 07D6, 0F00,093C,01C0,F800,0000,0A82   ;26635          PC.MODES?                       ;TEST FOR DEST REG=PC
                                        ;26636
                                        ;26637  =10     ;-------------------------------;DST R ISN'T PC
                                        ;26638          ALU_D-Q,R(DST)_ALU,             ;NEGATE DESTINATION, STORE RESULT
                                        ;26639          SET.CC(INST),                   ;SET CONDITION CODES
U 0A82, 401D,C000,0180,F8CD,4070,0762   ;26640          CLR.IB0-1,PC_PC+2,J/IRD11       ;CLEAR 2 BYTES, UPDATE PC
                                        ;26641
                                        ;26642          ;-------------------------------;DST REG IS PC
                                        ;26643          PC&VA_D-PC,SET.CC(INST),        ;NEGATE PC, SET CONDITION CODES
U 0A83, 2015,C000,0180,F801,4270,0D86   ;26644          FLUSH.IB,J/GO.IRD               ;NEW PC, SO FLUSH I-BUFFER
```

```
                                ;26645  7D9:    ;--------------------------------;
U 07D9, 0800,093C,0180,F800,0000,0AB2   ;26646  ASR.B:  D_LA,PC.MODES?                  ;D GETS DEST, SETUP FOR SIGN EXT
                                ;26647                                                  ;TEST DST-MODE
                                ;26648  =10     ;--------------------------------;DST R ISN'T PC
                                ;26649          R(DST) D.SXT[INST.DEP].RIGHT,   ;SHIFT RIGHT AND SIGN-EXTEND
                                ;26650          SI/ASHR,
                                ;26651          SET.PSL.C(AMX),                 ;SET PSL-C-BIT ON ORIGINAL DATA
U 0AB2, 0042,C03C,0080,F8C8,0060,0DA3   ;26652          J/ASR.B1
                                ;26653
                                ;26654          ;--------------------------------;DST R IS PC
U 0AB3, 0814,0038,0180,F800,0000,0DA1   ;26655          D_PC                            ;GET DST
                                ;26656  =;END
                                ;26657
                                ;26658          ;--------------------------------;
                                ;26659          D_D.SXT[INST.DEP].RIGHT,        ;SIGN EXTEND AND SHIFT PC
                                ;26660          SI/ASHR,
U 0DA1, 0842,C03C,0080,F800,0060,0DA2   ;26661          SET.PSL.C(AMX)                  ;SET PSL-C-BIT FROM ORIGINAL DATA
                                ;26662
                                ;26663          ;--------------------------------;
                                ;26664          PC&VA_ALU,ALU_D[AND]K[.FFFF],   ;RELOAD PC WITH SHIFTED PC
U 0DA2, 2019,C034,C180,F801,4270,0D86   ;26665          SET.CC(INST),FLUSH.IB,J/GO.IRD  ;NEW PC, SO FLUSH IB
                                ;26666
                                ;26667          ;--------------------------------;
                                ;26668  ASR.B1: ALU_R(DST),SET.CC(INST),        ;DST .NE. PC, SET CC
U 0DA3, 4000,C03C,0180,F84D,4070,0762   ;26669          CLR.IB0-1,PC_PC+2,J/IRD11       ;CLEAR 2 BYTES OF I-BUFFER, UPDATE PC
                                ;26670
                                ;26671
                                ;26672  7DB:    ;--------------------------------;
U 07DB, 0800,C03C,0180,F800,0070,0DA4   ;26673  MFP:    D_LA,SET.CC(INST)               ;MFPI, MFPD
                                ;26674
                                ;26675          ;--------------------------------;
U 0DA4, 0000,093C,0180,FA30,0000,0ABA   ;26676          LAB_R[R6],PC.MODES?             ;GET STACK-POINTER, TEST DST-MODE
                                ;26677
                                ;26678  =10     ;--------------------------------;DST R ISN'T PC
                                ;26679  MFPX:   R6_LA-K[.2].RLOG,               ;SP_SP-2
U 0ABA, 0018,4004,0980,FAB0,0200,0DC9   ;26680          VAR/LOAD,WORD,J/STORE0          ;VA_SP-2 FOR STORING RESULT
                                ;26681
                                ;26682          ;--------------------------------;DST R IS PC
U 0ABB, 0814,C038,0180,F800,0070,0ABA   ;26683          D_PC,SET.CC(INST),J/MFPX        ;GET PC
```

```
                                    ;26684  75E:    ;--------------------------------;
                                    ;26685  SXT:    D_K[.FFFF],Q_0,                 ;D HAS MINUS, Q HAS PLUS SIGN
U 075E, 0818,1A38,C1F8,F800,0000,0C57   ;26686          PSL.N?                          ;TEST PSL.N TO PROPAGATE IT
                                    ;26687
                                    ;26688  =0111   ;--------------------------------;PSL N=0
                                    ;26689          ALU_Q,R(DST)_ALU,               ;STORE ZERO
                                    ;26690          SET.CC(INST),
U 0C57, 4001,E03C,0180,F8CD,4070,0762   ;26691          CLR.IB0-1,PC_PC+2,J/IRD11
                                    ;26692
                                    ;26693          ;--------------------------------;PSL N=1
                                    ;26694          R(DST)_D,SET.CC(INST),          ;SET CC, DST_-1
U 0C5F, 4001,C03C,0180,F8CD,4070,0762   ;26695          CLR.IB0-1,PC_PC+2,J/IRD11
                                    ;26696
                                    ;26697  7DE:    ;--------------------------------;
U 07DE, 0000,003C,0180,F800,0000,075E   ;26698  SXTPC:  J/SXT                           ;DST IS PC, NOP AS 11/70
                                    ;26699
                                    ;26700  751:    ;--------------------------------;
                                    ;26701  TCLASS: ALU_LA[INST.DEP]D,              ;TST.B,BIT.B,CMP.B,
                                    ;26702          SET.CC(INST),                   ;SET/CLR COND CODES
U 0751, 401C,E00C,0180,F805,4070,0762   ;26703          CLR.IB0-1,PC_PC+2,J/IRD11
                                    ;26704
                                    ;26705  7D1:    ;--------------------------------;
U 07D1, 0814,0038,01E0,F800,0000,0DA5   ;26706  TCLPC:  Q_D,D_PC                        ;DST = PC
                                    ;26707
                                    ;26708          ;--------------------------------;
                                    ;26709          ALU_D[INST.DEP]Q,
                                    ;26710          SET.CC(INST),
U 0DA5, 401D,C00C,0180,F805,4070,0762   ;26711          CLR.IB0-1,PC_PC+2,J/IRD11
```

```
                                    ;26712
                                    ;26713  7D5:    ;--------------------------------;
                                    ;26714  MOVB:   D_D.SXT[BYTE],SET.CC(BYTE),     ;16 BIT MOVE FOR MOVB TO REG
U 07D5, 0802,893C,0180,F800,0070,0AC2 ;26715        PC.MODES?                       ;IS DESTINATION PC?
                                    ;26716
                                    ;26717  =10     ;--------------------------------;DST R ISN'T PC
                                    ;26718          R(DST)_D,WORD,                  ;WRITE 16-BIT REG
U 0AC2, 4001,403C,0180,F8CD,4000,0762 ;26719        CLR.IB0-1,PC_PC+2,J/IRD11
                                    ;26720
                                    ;26721          ;--------------------------------;DST R IS PC
                                    ;26722          PC&VA_D.OXT[WORD],SET.CC(WORD), ;MOVE TO PC
U 0AC3, 2003,403C,0180,F801,4270,0D86 ;26723        FLUSH.IB,J/GO.IRD
                                    ;26724
                                    ;26725  7DF:    ;--------------------------------;
                                    ;26726  SWAB:   D_LA,                           ;D GETS DST OPD,
                                    ;26727          SC_K[.10],Q_0,                  ; SET WORD SWAP CT
U 07DF, 0800,093C,65F8,F800,0084,6ACA ;26728        PC.MODES?                       ;CK IF DST IS PC
                                    ;26729
                                    ;26730  =10     ;--------------------------------;DST R ISN'T PC
                                    ;26731  SWAB.0: D_DAL.SC,                       ;MOVE SOURCE TO D BYTES 3 & 2
                                    ;26732          ALU_D.AND.K[.FF00],             ;CC'S SET FROM RESULTING LOW BYTE
U 0ACA, 0D19,4034,4D80,F800,0050,0DA6 ;26733        WORD,N&Z_ALU.V&C_0,J/SWAB.1
                                    ;26734
                                    ;26735          ;--------------------------------;DST R IS PC
U 0ACB, 0814,0038,0180,F800,0000,0ACA ;26736        D_PC,J/SWAB.0
                                    ;26737
                                    ;26738          ;--------------------------------;
                                    ;26739  SWAB.1: D_D.SWAP,                       ;SWAP BYTES 3&2 INTO 0&1
U 0DA6, 0B00,093C,0180,F800,0000,0AE2 ;26740        PC.MODES?                       ;IS DESTINATION PC?
                                    ;26741
                                    ;26742  =10     ;--------------------------------;DST R ISN'T PC
                                    ;26743          R(DST)_D,WORD,                  ;STORE RESULT
U 0AE2, 4001,403C,0180,F8CD,4000,0762 ;26744        CLR.IB0-1,PC_PC+2,J/IRD11
                                    ;26745
                                    ;26746          ;--------------------------------;
U 0AE3, 2001,003C,0180,F801,4200,0D86 ;26747        PC&VA_D,FLUSH.IB,J/GO.IRD       ;STORE RESULT TO PC AND VA
```

```
                                ;26748  757:    ;--------------------------------;
                                ;26749  ROR:    D_LA.RIGHT,                     ;GET SOURCE SHIFTED RIGHT
                                ;26750          SC_K[.F],                       ;SETUP TO MASK BIT 15
                                ;26751          SET.PSL.C(AMX),                 ;SET COND CODES PART 1
U 0757, 0840,1A3C,6180,F800,00E4,6C7E   ;26752  PSL.C?                          ;SEN ON PSL<C>
                                ;26753
                                ;26754  =1110   ;--------------------------------;
                                ;26755  RORX:   ALU_D.AND.MASK,R(DST)_ALU,      ;MASK OUT LMB OF BYTE/WORD
                                ;26756          SET.CC(INST),
U 0C7E, 4001,C034,0180,F8CD,4070,0762   ;26757  CLR.IB0-1,PC_PC+2,J/IRD11
                                ;26758
                                ;26759          ;--------------------------------;
                                ;26760          ALU_D.ORNOT.MASK,R(DST)_ALU,    ;SET LMB OF BYTE/WORD
                                ;26761          SET.CC(INST),
U 0C7F, 4001,C01C,0180,F8CD,4070,0762   ;26762  CLR.IB0-1,PC_PC+2,J/IRD11
                                ;26763
                                ;26764  758:    ;--------------------------------;
                                ;26765  RORB:   D_LA.RIGHT,SI/ZERO,
                                ;26766          SC_K[.7],
                                ;26767          SET.PSL.C(AMX),
U 0758, 0840,1A3C,5D80,F800,00E4,6C7E   ;26768  PSL.C?,J/RORX
                                ;26769
                                ;26770  7D7:    ;--------------------------------;
U 07D7, 0814,0038,6180,F800,0084,6DA8   ;26771  RORPC:  D_PC,SC_K[.F],J/RORY
                                ;26772
                                ;26773  7D8:    ;--------------------------------;
U 07D8, 0814,0038,5D80,F800,0084,6DA8   ;26774  RORBPC: D_PC,SC_K[.7]
                                ;26775
                                ;26776          ;--------------------------------;
                                ;26777  RORY:   D_D.RIGHT,SI/ZERO,              ;SHIFT RIGHT
                                ;26778          ALU_D,SET.PSL.C(AMX),           ;SET COND CODES PART 1
U 0DA8, 0601,1A3C,0180,F800,0060,0C8E   ;26779  PSL.C?                          ;IS OLD C SET?
                                ;26780
                                ;26781  =1110   ;--------------------------------;
                                ;26782          PC&VA_ALU,ALU_D[AND]K[.FFFF],
                                ;26783          SET.CC(INST),
U 0C8E, 2019,C034,C180,F801,4270,0D86   ;26784  FLUSH.IB,J/GO.IRD
                                ;26785
                                ;26786          ;--------------------------------;
                                ;26787          ALU_D.ORNOT.MASK,PC&VA_ALU,
                                ;26788          SET.CC(INST),                   ;SET COND CODES PART 2
U 0C8F, 2001,C01C,0180,F801,4270,0D86   ;26789  FLUSH.IB,J/GO.IRD
```

```
                                        ;26790  .TOC     "       Compatibility mode    : DP0 -  DM0*(ASH+ASHC+MUL+DIV)"
                                        ;26791
                                        ;26792  ;ASH, ASHC, MUL, DIV
                                        ;26793  ;ENTER HERE FROM IRD,
                                        ;26794  ;  WITH LA GETS DST,
                                        ;26795  ;  LB GETS SRC,
                                        ;26796  ;  VA AND D GET SRC OR PC (LB OR PC),
                                        ;26797  ;  Q GETS IB DATA BYTES 2 AND 3,
                                        ;26798  ;  PC IS ADDED BY N WHERE N IS DETERMINED BY IB,
                                        ;26799  ;  SC IS SET TO 16.
                                        ;26800
                                        ;26801  7BB:    ;-------------------------------;
                                        ;26802  ASH:    SC_LA.AND.K[.3F],
                                        ;26803          Q_D,
U 07BB, 0018,0934,55E0,F800,0082,0B12   ;26804          PC.MODES?,J/ASHU
                                        ;26805
                                        ;26806  7BC:    ;-------------------------------;
                                        ;26807  ASHC:   Q_LA.AND.K[.3F],
U 07BC, 0018,0934,55C0,F800,0000,0B1A   ;26808          PC.MODES?,J/ASHCU
                                        ;26809
                                        ;26810  7B8:    ;-------------------------------;
                                        ;26811  MULELE: D_LA,Q_D,                       ;GO PROCESS AS MEM TO REG
U 07B8, 0800,093C,01E0,F800,0000,0AF2   ;26812          PC.MODES?
                                        ;26813
                                        ;26814  =10     ;-------------------------------;
                                        ;26815  MUL6:   SC_K[.7],                       ;SET LOOP CT FOR 8
                                        ;26816          D_Q.SXT[WORD],Q_D,
U 0AF2, 0802,6D3C,5DE0,F800,0084,6F65   ;26817          D.NE.0?,J/MULU                  ;SIGN EXT M'CAND IN D
                                        ;26818
                                        ;26819          ;-------------------------------;
U 0AF3, 0814,0038,0180,F800,0000,0AF2   ;26820          D_PC,J/MUL6                     ;DST IS PC
                                        ;26821
                                        ;26822  7BE:    ;-------------------------------;
                                        ;26823  DIVELE: D_LA,Q_D,                       ;GO PROCESS AS MEM TO REG
                                        ;26824          FE&SC_K[.10],
U 07BE, 0800,093C,65E0,F800,0184,6AFA   ;26825          PC.MODES?
                                        ;26826
                                        ;26827  =10     ;-------------------------------;
                                        ;26828  DIV6:   RC[TO]_Q.SXT[WORD],             ;SIGN EXT D'END, SET SHF CT
U 0AFA, 0002,603C,01F8,F980,0000,0DD4   ;26829          Q_0,J/DIVX
                                        ;26830
                                        ;26831          ;-------------------------------;
U 0AFB, 0814,0038,0180,F800,0000,0AFA   ;26832          D_PC,J/DIV6                     ;DST IS PC
```

```
                                ;26833  .TOC "  Compatibility mode    : DP0 -  SRC MODE DECODING FOR BIN INSTRUCTIONS"
                                ;26834
                                ;26835  ;DECODE SRC ADDRESS MODE FOR 2 OPERAND INSTRUCTIONS EXCEPT ASH, ASHC, MUL, DIV.
                                ;26836  ;       ENTER HERE FROM IRD,
                                ;26837  ;         WITH LA GETS DST,
                                ;26838  ;         LB GETS SRC,
                                ;26839  ;         VA AND D GET SRC OR PC (LB OR PC),
                                ;26840  ;         Q GETS IB DATA BYTES 2 AND 3,
                                ;26841  ;         PC IS ADDED BY N WHERE N IS DETERMINED BY IB,
                                ;26842  ;         SC IS SET TO 16.
                                ;26843  ;LEAVE HERE TO DP1 WITH D CONTAINS SRC OPERAND, AND REST UPDATED IF NEEDED.
                                ;26844
                                ;26845  ; -11 ADDRESSING MODES:
                                ;26846  ;
                                ;26847  ;                 R .NE. PC                       R .EQ. PC
                                ;26848  ;       MODE      NAME            SYNTAX          NAME            SYNTAX
                                ;26849  ;       0         REGISTER        R
                                ;26850  ;       1         REG DEFERRED    (R)
                                ;26851  ;       2         AUTOINCREMENT   (R)+            IMMEDIATE       #N
                                ;26852  ;       3         AUTOINC DEFER   @(R)+           ABSOLUTE        @#A
                                ;26853  ;       4         AUTODECREMENT   -(R)
                                ;26854  ;       5         AUTODEC DEFER   @-(R)
                                ;26855  ;       6         INDEX           X(R)            RELATIVE        A
                                ;26856  ;       7         INDEX DEFERRED  @X(R)           RELATIVE DEFER  @A
                                ;26857
                                ;26858  711:    ;-------------------------------;
                                ;26859  SM1:    D[INST.DEP]_CACHE,PC_PC+N,      ;SM1
                                ;26860          LA_R(DST)&LB_R(SRC),            ;RE-LATCH REG'S (MAY BE MODIFIED)
U 0711, 0000,C03F,01F0,4057,0000,0700   ;26861          QK7ID,FS/MCT,SUB/SPEC,J/DP0
                                ;26862
                                ;26863          ;-------------------------------;
                                ;26864  719:    D[INST.DEP]_CACHE,PC_PC+N,      ;SM1
                                ;26865          LA_R(DST)&LB_R(SRC),
U 0719, 0000,C03F,01F0,4057,0000,0700   ;26866          QK7ID,FS/MCT,SUB/SPEC,J/DP0
                                ;26867
                                ;26868  712:    ;-------------------------------;
                                ;26869  SM2:    R(SRC)_D+K[SP1.CON].RLOG,WORD,  ;SM2*-SM27
U 0712, 0019,4018,1580,F8C0,0000,0711   ;26870          J/SM1
                                ;26871
                                ;26872  73A:    ;-------------------------------;
                                ;26873  SM2A:   ALU_LA[INST.DEP]Q,R(DST)_ALU,   ;SM27*DM0*ACLASS
                                ;26874          SET.CC(INST),
U 073A, E01C,C00C,0180,F8CD,0070,0762   ;26875          CLR.IB0-3,PC_PC+2,J/IRD11       ;-DM07
                                ;26876
                                ;26877  77A:    ;-------------------------------;
U 077A, 0002,A03C,01C0,F800,0070,0DA9   ;26878  SM2MVB: Q_Q.SXT[BYTE],SET.CC(BYTE)      ;SM27*DM0*MOVB
                                ;26879
                                ;26880          ;-------------------------------;
                                ;26881          ALU_Q,R(DST)_ALU,WORD,
U 0DA9, E001,603C,0180,F8CD,0000,0762   ;26882          CLR.IB0-3,PC_PC+2,J/IRD11       ;-DM07
                                ;26883
                                ;26884  75A:    ;-------------------------------;
                                ;26885  SM2T:   ALU_LA[INST.DEP]Q,              ;SM27*DM0*TCLASS
                                ;26886          SET.CC(INST),
U 075A, E01C,C00C,0180,F805,0070,0762   ;26887          CLR.IB0-3,PC_PC+2,J/IRD11       ;-DM07
```

```
                                        ;26888  71A:    ;--------------------------------;
U 071A, 5000,003C,0180,F800,0000,0DAA   ;26889  SM2DMX: CLR.IB2-3                        ;SM27*-DM0
                                        ;26890
                                        ;26891          ;--------------------------------;
                                        ;26892          D_Q,
                                        ;26893          PC_PC+N,
U 0DAA, 0C00,003F,01F0,F807,0000,0700   ;26894          Q_IB.DATA,SUB/SPEC,J/DP0
                                        ;26895
                                        ;26896  713:    ;--------------------------------;
                                        ;26897  SM3:    D[WORD]_CACHE,
                                        ;26898          R(SRC)_D+K[.2].RLOG,            ;SM3*-SM37
                                        ;26899          CHK.ODD.ADDR,
U 0713, 0019,4018,0980,40C0,0C00,0D80   ;26900          J/SM5.1
                                        ;26901
                                        ;26902  71B:    ;--------------------------------;
                                        ;26903  SM37:   VA_ALU,ALU_Q.OXT[WORD],         ;SM37
U 071B, 5003,603C,0180,F800,0200,0711   ;26904          CLR.IB2-3,J/SM1
                                        ;26905
                                        ;26906  714:    ;--------------------------------;
                                        ;26907  SM4:    R(SRC)_D-K[SP1.CON].RLOG,       ;SM4*-SM47
U 0714, 0019,4004,1580,F8C0,0200,0711   ;26908          VAK/LOAD,J/SM1
                                        ;26909
                                        ;26910  71C:    ;--------------------------------;
U 071C, 0814,0038,0180,F800,0000,0DAB   ;26911  SM47:   D_PC
                                        ;26912
                                        ;26913          ;--------------------------------;
U 0DAB, 0019,0000,0980,F801,0200,0DAC   ;26914          PC&VA_D-K[.2]
                                        ;26915
                                        ;26916          ;--------------------------------;
U 0DAC, 0000,403C,0180,4000,0000,0DAD   ;26917          D[WORD]_CACHE                   ;GET INSTRUCTION ITSELF AS SRC
                                        ;26918
                                        ;26919          ;--------------------------------;
                                        ;26920          Q_D,PC_PC+N,
U 0DAD, 0000,003F,01E0,F807,0000,0700   ;26921          SUB/SPEC,J/DP0
                                        ;26922
                                        ;26923  715:    ;--------------------------------;
                                        ;26924  SM5:    R(SRC)_D-K[.2].RLOG,            ;SM5*-SM57
U 0715, 0019,4004,0980,F8C0,0200,0DAE   ;26925          VAK/LOAD
                                        ;26926
                                        ;26927          ;--------------------------------;
U 0DAE, 0000,403C,0180,4000,0C00,0DB0   ;26928  SM5X:   D[WORD]_CACHE,CHK.ODD.ADDR
                                        ;26929
                                        ;26930          ;--------------------------------;
                                        ;26931  SM5.1:  VA_ALU,ALU_D.OXT[WORD],
U 0DB0, 0003,403C,0180,F800,0200,0711   ;26932          J/SM1
                                        ;26933
                                        ;26934  71D:    ;--------------------------------;
U 071D, 0814,0038,0180,F800,0000,0DB1   ;26935  SM57:   D_PC                            ;SM57
                                        ;26936
                                        ;26937          ;--------------------------------;
U 0DB1, 0019,0000,0980,F801,0200,0DB2   ;26938          PC&VA_D-K[.2]
                                        ;26939
                                        ;26940          ;--------------------------------;
U 0DB2, 0000,403C,0180,4000,0C00,0DB3   ;26941          D[WORD]_CACHE,CHK.ODD.ADDR      ;
```

```
                                    ;26942          ;---------------------------------;
                                    ;26943          VA_ALU,ALU_D.OXT[WORD],         ;Q GETS INSTRUCTION ITSELF
U 0DB3, 0003,403C,01E0,F800,0200,0DB4 ;26944        Q_D
                                    ;26945
                                    ;26946          ;---------------------------------;
                                    ;26947          D[INST.DEP] CACHE,
U 0DB4, 0000,C03F,0180,4007,0000,0700 ;26948        PC_PC+N,SUB7SPEC,J/DP0
                                    ;26949
                                    ;26950  716:    ;---------------------------------;
                                    ;26951  SM6:    VA_ALU,ALU_Q.SXT[WORD]+LB,      ;SM6
U 0716, 500E,6014,0180,F800,0200,0711 ;26952        CLR.IB2-3,J/SM1
                                    ;26953
                                    ;26954  71E:    ;---------------------------------;
                                    ;26955  SM67:   VA_ALU,ALU_Q.SXT[WORD]+PC,
U 071E, 5016,6014,0180,F800,0200,0DB5 ;26956        CLR.IB2-3
                                    ;26957
                                    ;26958          ;---------------------------------;
                                    ;26959          D[INST.DEP]_CACHE,
                                    ;26960          PC_PC+N,
U 0DB5, 0000,C03F,01F0,4007,0000,0700 ;26961        QK7ID,FS/MCT,SUB/SPEC,J/DP0
                                    ;26962
                                    ;26963  717:    ;---------------------------------;
                                    ;26964  SM7:    VA_ALU,ALU_Q.SXT[WORD]+LB,      ;SM7
U 0717, 500E,6014,0180,F800,0200,0DAE ;26965        CLR.IB2-3,J/SM5X
                                    ;26966
                                    ;26967  71F:    ;---------------------------------;
                                    ;26968          VA_ALU,ALU_Q.SXT[WORD]+PC,      ;SM7
U 071F, 5016,6014,0180,F800,0200,0DAE ;26969        CLR.IB2-3,J/SM5X
                                    ;26970
                                    ;26971  7D4:    ;---------------------------------;
                                    ;26972  MTP1:   R6_D+K[.2].RLOG,                ;(SP)+
U 07D4, 0019,0018,0980,FA80,0000,0711 ;26973        J/SM1
                                    ;26974
                                    ;26975  77C:    ;---------------------------------;
U 077C, 0000,003D,0180,F800,0000,0E64 ;26976        CALL,J/IB.TBM                   ;IB TB MISS, GOTO FILL IB
                                    ;26977
                                    ;26978  77D:    ;---------------------------------;
U 077D, 0000,003D,0180,F800,0000,0880 ;26979        CALL,J/IB.ERR                   ;IB ERROR
                                    ;26980
                                    ;26981  77E:    ;---------------------------------;
                                    ;26982          Q_IB.DATA,                      ;IB STALL
                                    ;26983          PC_PC+N,
U 077E, 0000,003F,01F0,F807,0000,0700 ;26984        SUB/SPEC,J/DP0
```

```
                                ;26985  .TOC "  Compatibility mode    : DP0(DP1 FOR BIN INSTRUCTIONS) -  DEST MODE DECODING"
                                ;26986
                                ;26987  ;R ENTER HERE FROM IRD,
                                ;26988  ;  WITH LA GETS DST,
                                ;26989  ;  LB GETS SRC,
                                ;26990  ;  VA AND D GET SRC OR PC (LB OR PC),
                                ;26991  ;  Q GETS IB DATA BYTES 2 AND 3,
                                ;26992  ;  PC IS ADDED BY N WHERE N IS DETERMINED BY IB,
                                ;26993  ;  SC IS SET TO 16.
                                ;26994  ;IF ENTER HERE AFTER SRC ADDRESS DECODING,
                                ;26995  ;  D GETS SRC OPERAND,
                                ;26996  ;  VA IS SRC OPERAND ADDRESS,
                                ;26997  ;  PC HAS BEEN UPDATED, AND REST EITHER UPDATED FOR DST, OR NO CHANGE.
                                ;26998  ;LEAVE HERE FOR NEXT DP FOR EXECUTES WITH
                                ;26999  ;  D CONTAINS DST OPERAND, VA IS DST OPERAND ADDRESS, Q HAS SRC OPERAND.
                                ;27000
                                ;27001  ; -11 ADDRESSING MODES:
                                ;27002  ;
                                ;27003  ;                R .NE. PC                          R .EQ. PC
                                ;27004  ;       MODE     NAME              SYNTAX           NAME              SYNTAX
                                ;27005  ;       0        REGISTER          R
                                ;27006  ;       1        REG DEFERRED      (R)
                                ;27007  ;       2        AUTOINCREMENT     (R)+             IMMEDIATE         #N
                                ;27008  ;       3        AUTOINC DEFER     @(R)+            ABSOLUTE          @#A
                                ;27009  ;       4        AUTODECREMENT     -(R)
                                ;27010  ;       5        AUTODEC DEFER     @-(R)
                                ;27011  ;       6        INDEX             X(R)             RELATIVE          A
                                ;27012  ;       7        INDEX DEFERRED    @X(R)            RELATIVE DEFER    @A
                                ;27013
                                ;27014  701:    ;--------------------------------;
                                ;27015  DM1:    VA_LA,RC[T7]_ALU,                  ;DM1*-DM17
U 0701, 0000,083C,0180,F9B8,0200,0F0C ;27016            END.DP1?,J/DP1A
                                ;27017
                                ;27018  709:    ;--------------------------------;
                                ;27019  DM17:   VA_PC,RC[T7]_ALU,                  ;DM17
U 0709, 0014,0838,0180,F9B8,0200,0F0C ;27020            END.DP1?,J/DP1A
                                ;27021
                                ;27022  702:    ;--------------------------------;
                                ;27023  DM2:    VA_LA,RC[T7]_ALU,                  ;DM2*-DM27
U 0702, 0000,083C,0180,F9B8,0200,0F1C ;27024            END.DP1?,J/DP1B
                                ;27025
                                ;27026  70A:    ;--------------------------------;
U 070A, 001B,0000,0980,FAF8,0000,0DB6 ;27027  DM27:   R[R15]_0-K[.2]                ;DM27
                                ;27028
                                ;27029          ;--------------------------------;
U 0DB6, 0000,003C,0180,FA78,0000,0DB8 ;27030          LAB_R[R15]
                                ;27031
                                ;27032          ;--------------------------------;
                                ;27033          VA_LA+PC,RC[T7]_ALU,
                                ;27034          CLR.IB2-3,
U 0DB8, 5014,0814,0180,F9B8,0200,0F0C ;27035          END.DP1?,J/DP1A
                                ;27036
                                ;27037  703:    ;--------------------------------;
U 0703, 0000,003C,0180,F800,0200,0DB9 ;27038  DM3:    VA_LA                          ;DM3*-DM37
```

```
                                        ;27039          ;--------------------------------;
                                        ;27040          Q_D,
                                        ;27041          A[U_LA+K[.2].RLOG,R(DST)_ALU,
                                        ;27042          D[WORD]_CACHE,
U 0DB9, 0018,4018,09E0,40C8,0C00,0DBA   ;27043          CHK.ODD.ADDR
                                        ;27044
                                        ;27045          ;--------------------------------;
                                        ;27046  DM3V:   VA_D,RC[T7]_ALU,
                                        ;27047          D_Q,
U 0DBA, 0C01,083C,0180,F9B8,0200,0F0C   ;27048          END.DP1?,J/DP1A
                                        ;27049
                                        ;27050  70B:    ;--------------------------------;
                                        ;27051  DM37:   VA_Q,RC[T7]_ALU,                ;DM37
                                        ;27052          Q_D,
                                        ;27053          C[R.IB2-3,
U 070B, 5001,283C,01E0,F9B8,0200,0F0C   ;27054          END.DP1?,J/DP1A
```

```
                                    ;27055  704:    ;--------------------------------;
                                    ;27056  DM4:    ALU_LA-K[SP2.CON].RLOG,          ;DM4*-DM47
                                    ;27057          R(DST)_ALU,WORD,VAK/LOAD,
U 0704, 0018,4804,1980,F8C8,0200,0F14 ;27058        END.DPT?,J/DP1C
                                    ;27059
                                    ;27060  70C:    ;--------------------------------;
U 070C, 0014,0038,01C0,F800,0000,0DBB ;27061  DM47:   Q_PC                              ;DM47
                                    ;27062
                                    ;27063          ;--------------------------------;
                                    ;27064          PC&VA_Q-K[.2],
                                    ;27065          RC[T7]_ALU,
U 0DBB, 0019,2800,0980,F9B9,0200,0F0C ;27066        END.DPT?,J/DP1A
                                    ;27067
                                    ;27068  705:    ;--------------------------------;
                                    ;27069  DM5:    ALU_LA-K[.2].RLOG,               ;DM5*-DM57
                                    ;27070          R(DST)_ALU,WORD,VAK/LOAD,
U 0705, 0018,4004,0980,F8C8,0200,0DBC ;27071        J/DM5X
                                    ;27072
                                    ;27073          ;--------------------------------;
                                    ;27074  DM5X:   Q_D,
                                    ;27075          D[WORD]_CACHE,
U 0DBC, 0000,403C,01E6,4000,0C00,0DBA ;27076        CHK.ODD.ADDR,J/DM3V
                                    ;27077
                                    ;27078  70D:    ;--------------------------------;
U 070D, 0014,0038,01C0,F800,0000,0DBD ;27079  DM57:   Q_PC                              ;DM57
                                    ;27080
                                    ;27081          ;--------------------------------;
                                    ;27082          PC&VA_Q-K[.2],
U 0DBD, 0019,2000,0980,F801,0200,0DBC ;27083        J/DM5X
                                    ;27084
                                    ;27085  706:    ;--------------------------------;
                                    ;27086  DM6:    VA_LA+Q,                         ;DM6*-DM67
                                    ;27087          RC[T7]_ALU,
                                    ;27088          CLR.IB2-3,
U 0706, 501C,0814,0180,F9B8,0200,0F0C ;27089        END.DP1?,J/DP1A
                                    ;27090
                                    ;27091  70E:    ;--------------------------------;
                                    ;27092  DM67:   VA_Q+PC,                         ;DM67
                                    ;27093          RC[T7]_ALU,
                                    ;27094          CLR.IB2-3,
U 070E, 5015,2814,0180,F9B8,0200,0F0C ;27095        END.DP1?,J/DP1A                  ;GOTO CHK ODD ADDR
                                    ;27096
                                    ;27097  707:    ;--------------------------------;
                                    ;27098  DM7:    VA_LA+Q,                         ;DM7*-DM77
U 0707, 501C,0014,0180,F800,0200,0DBC ;27099        CLR.IB2-3,J/DM5X
                                    ;27100
                                    ;27101  70F:    ;--------------------------------;
                                    ;27102  DM77:   VA_Q+PC,                         ;DM77
U 070F, 5015,2014,0180,F800,0200,0DBC ;27103        CLR.IB2-3,J/DM5X
                                    ;27104
                                    ;27105  =100    ;--------------------------------;
                                    ;27106  DP1A:   D[INST.DEP]_CACHE.IBCHK,         ;-(JCLASS+DM27)
                                    ;27107          Q_D,
U 0F0C, 0000,C03F,01E0,5800,0000,0700 ;27108        SUB/SPEC,J/DP0
```

```
                                    ;27109
                                    ;-------------------------------;
                                    ;27110  DP1AA:  RC[TO]_PC,                      ;JCLASS+DM27:  SAVE PC FOR JSR
                                    ;27111          D_Q,Q_D,
U 0F0D, 0C14,003B,01E0,F980,0000,0700 ;27112          SOB/SPEC,J/DP0
                                    ;27113
                                    ;27114          ;-------------------------------;
                                    ;27115          CACHE_D[INST.DEP],             ;MOV.B + MTP
                                    ;27116          ALU_D,SET.CC(INST),
U 0F0E, 0001,C03C,0180,3000,0070,0B03 ;27117          J/STORE1
                                    ;27118
                                    ;27119          ;-------------------------------;
                                    ;27120          CACHE_D[INST.DEP],             ;MOV.B + MTP
                                    ;27121          ALU_D,SET.CC(INST),
U 0F0F, 0001,C03C,0180,3000,0070,0B03 ;27122          J/STORE1
                                    ;27123  =
                                    ;27124  =100    ;-------------------------------;
                                    ;27125  DP1C:   D[INST.DEP]_CACHE.IBCHK,       ;-(JCLASS+DM27)
                                    ;27126          Q_D,
U 0F14, 0000,C03F,01E0,5800,0000,0700 ;27127          SOB/SPEC,J/DP0
                                    ;27128
                                    ;27129          ;-------------------------------;
                                    ;27130          LAB_R(DST),                    ;JCLASS+DM27:
U 0F15, 0000,003C,0180,F848,0000,0DBE ;27131          J/DP1CC
                                    ;27132
                                    ;27133          ;-------------------------------;
                                    ;27134          CACHE_D[INST.DEP],             ;MOV.B + MTP
                                    ;27135          ALU_D,SET.CC(INST),
U 0F16, 0001,C03C,0180,3000,0070,0B03 ;27136          J/STORE1
                                    ;27137
                                    ;27138          ;-------------------------------;
                                    ;27139          CACHE_D[INST.DEP],             ;MOV.B + MTP
                                    ;27140          ALU_D,SET.CC(INST),
U 0F17, 0001,C03C,0180,3000,0070,0B03 ;27141          J/STORE1
                                    ;27142  =
                                    ;27143          ;-------------------------------;
U 0DBE, 0000,003C,0180,F988,0000,0F0D ;27144  DP1CC:  RC[T7]_LA,J/DP1AA               ;JCLASS+DM27:
```

```
                                    ;27145  =100    ;--------------------------------;
                                    ;27146  DP1B:   ALU_LA+K[SP2.CON].RLOG,          ;-(JCLASS+DM27)
                                    ;27147          R(DST)_ALU,WORD,                 ;STORE POPPED REGISTER
U 0F1C, 0018,4018,1980,F8C8,0000,0F0C ;27148        J/DP1A
                                    ;27149
                                    ;27150          ;--------------------------------;
                                    ;27151          ALU_LA+K[SP2.CON].RLOG,
                                    ;27152          R(DST)_ALU,WORD,
U 0F1D, 0018,4018,1980,F8C8,0000,0F0D ;27153        J/DP1AA                          ;JCLASS
                                    ;27154
                                    ;27155          ;--------------------------------;
                                    ;27156          CACHE_D[INST.DEP],               ;MOV.B + MTP
                                    ;27157          ALU_D,SET.CC(INST),
U 0F1E, 0001,C03C,0180,3000,0070,0DC0 ;27158        J/DP1Y
                                    ;27159
                                    ;27160          ;--------------------------------;
                                    ;27161          CACHE_D[INST.DEP],               ;MOV.B + MTP
                                    ;27162          ALU_D,SET.CC(INST),
U 0F1F, 0001,C03C,0180,3000,0070,0DC0 ;27163        J/DP1Y
                                    ;27164  =
                                    ;27165
                                    ;27166          ;--------------------------------;
                                    ;27167  DP1Y:   ALU_LA+K[SP2.CON].RLOG,          ;MOV.B+MTP
                                    ;27168          R(DST)_ALU,WORD,
U 0DC0, 0018,4018,1980,F8C8,0000,0B03 ;27169        J/STORE1
```

```
                                ;27170  .TOC "  Compatibility mode    : DP1 -  MEMORY TO R/PC FOR BIN INSTRUCTIONS"
                                ;27171
                                ;27172  ;ENTER HERE WITH LA HAS DST, D HAS SRC OPERAND.
                                ;27173  ;EXECUTE THE BINARY OPERATION INSTRUCTIONS AND GO TO IRD AFTER DONE.
                                ;27174
                                ;27175  774:    ;-------------------------------;
                                ;27176  BCL1:   ALU_LA[INST.DEP]D,R(DST)_ALU,   ;BIC.B,BIS.B,ADD,SUB,MOV
                                ;27177          SET.CC(INST),
                                ;27178          CLR.IB0-1,PC_PC+2,
U 0774, 401C,E90C,0180,F8CD,4070,0B09   ;27179          PC.MODES?,J/[OADIB
                                ;27180
                                ;27181  7F4:    ;-------------------------------;
U 07F4, 0014,0038,01C0,F800,0000,0DC1   ;27182  BCL1PC: Q_PC                     ;DST = PC
                                ;27183  =
                                ;27184          ;-------------------------------;
                                ;27185          PC&VA_ALU,ALU_Q[INST.DEP]D,
                                ;27186          SET.CC(INST),
U 0DC1, 201D,E00C,0180,F801,4270,0D86   ;27187          FLUSH.IB,J/GO.IRD
                                ;27188
                                ;27189  73F:    ;-------------------------------;
                                ;27190  BCL2:   ALU_LA[INST.DEP]D,SET.CC(INST),
                                ;27191          CLR.IB0-1,PC_PC+2,
U 073F, 401C,E90C,0180,F805,4070,0B09   ;27192          PC.MODES?,J/[OADIB
                                ;27193
                                ;27194  =01     ;-------------------------------;
U 0B01, 0804,003B,65F0,F857,1284,6700   ;27195  BCLIRD: IRD.11                   ;DST.NE.PC*-(SM47,57)
                                ;27196
                                ;27197          ;-------------------------------;
U 0B03, 2014,0038,0180,F800,4200,0D86   ;27198  STORE1: ALU_PC,FLUSH.IB,J/GO.IRD  ;DST.NE.PC*(SM47,57)
                                ;27199
                                ;27200  7BF:    ;-------------------------------;
U 07BF, 0014,0038,01C0,F800,0000,0DC2   ;27201  BCL2PC: Q_PC                     ;DST = PC
                                ;27202
                                ;27203          ;-------------------------------;
                                ;27204          ALU_Q[INST.DEP]D,SET.CC(INST),
                                ;27205          CLR.IB0-1,PC_PC+2,
U 0DC2, 401D,E00C,0180,F805,4070,0762   ;27206          J/IRD11
                                ;27207
                                ;27208  779:    ;-------------------------------;
U 0779, 0802,803C,0180,F800,0070,0DC3   ;27209  MOVB1:  D_D.SXT[BYTE],SET.CC(BYTE)  ;SIGN EXT FOR MOVB TO REG
                                ;27210
                                ;27211          ;-------------------------------;
                                ;27212          R(DST)_D,WORD,
                                ;27213          CLR.IB0-1,PC_PC+2,
U 0DC  4001,493C,0180,F8CD,4000,0B09    ;27214          PC.MODES?,J/[OADIB
                                ;27215
                                ;27216  7F9:    ;-------------------------------;
                                ;27217  MOVB1PC:PC&VA_ALU,
                                ;27218          ALU_D.SXT[BYTE].AND.K[.FFFF],
                                ;27219          SET.CC(BYTE),
U 07F9, 201A,8034,C180,F801,4270,0D86   ;27220          FLUSH.IB,J/GO.IRD
```

```
                                        ;27221  .TOC "  Compatibility mode    : DP1 (DP2 FOR BIN INSTRUCTIONS) -  MEMORY TO MEMORY,
                                        ;27222
                                        ;27223  ;HERE IS MEMORY TO MEMORY EXECUTES.
                                        ;27224  ;ENTER WITH D HAS DST OPERAND, VA CONTAINS DST OPERAND ADDRESS, AND
                                        ;27225  ;Q CONTAINS SRC OPERAND.
                                        ;27226  ;MTP INSTRUCTION IS EXCEPTION CASE.  THE DMO* IS HERE, AND -DMO* IS WITHIN
                                        ;27227  ;MOV INSTRUCTIONS.
                                        ;27228
                                        ;27229  700:    ;-------------------------------;ADDRESS ZERO FOR CM
                                        ;27230  DP0:    R(DST)_D,                       ;DST R NOT = PC
                                        ;27231          SET.CC(INST),
U 0700, 4001,C03C,0180,F8CD,4070,0762   ;27232          CLR.IB0-1,PC_PC+2,J/IRD11
                                        ;27233
                                        ;27234  708:    ;-------------------------------;
                                        ;27235  MTP:    PC&VA_D,SET.CC(INST),           ;DST R = PC
U 0708, 2001,C03C,0180,F801,4270,0D86   ;27236          FLUSH.IB,J/GO.IRD
                                        ;27237
                                        ;27238  769:    ;-------------------------------;
                                        ;27239  ACLAS7: D_D[INST.DEP]Q,                 ;ACLASS (BIN+C/CLASS) + MOVB
U 0769, 081D,C00C,0180,F800,0070,0DC9   ;27240          SET.CC(INST),J/STORE0
                                        ;27241
                                        ;27242  764:    ;-------------------------------;
                                        ;27243          D_D[INST.DEP]Q,                 ;ACLASS (BIN+C/CLASS) + MOVB
U 0764, 081D,C00C,0180,F800,0070,0DC9   ;27244          SET.CC(INST),J/STORE0
                                        ;27245
                                        ;27246  7E9:    ;-------------------------------;
                                        ;27247          D_D[INST.DEP]Q,                 ;ACLASS (BIN+C/CLASS) + MOVB
U 07E9, 081D,C00C,0180,F800,0070,0DC9   ;27248          SET.CC(INST),J/STORE0
                                        ;27249
                                        ;27250  7E4:    ;-------------------------------;
                                        ;27251          D_D[INST.DEP]Q,                 ;ACLASS (BIN+C/CLASS) + MOVB
U 07E4, 081D,C00C,0180,F800,0070,0DC9   ;27252          SET.CC(INST),J/STORE0
                                        ;27253
                                        ;27254  7C6:    ;-------------------------------;
U 07C6, 0000,003C,01F8,F800,0000,0DC4   ;27255  NEG.B7: Q_0
                                        ;27256
                                        ;27257          ;-------------------------------;
                                        ;27258          D_Q-D,
U 0DC4, 081D,E000,0180,F800,0070,0DC9   ;27259          SET.CC(INST),J/STORE0
                                        ;27260
                                        ;27261  7C9:    ;-------------------------------;
                                        ;27262  ASR.B7: D_D.SXT[INST.DEP].RIGHT,SI/ASHR,
U 07C9, 0842,C03C,0080,F800,0060,0DC5   ;27263          SET.PSL.C(AMX)
                                        ;27264
                                        ;27265          ;-------------------------------;
                                        ;27266          CACHE_D[INST.DEP],RAMX/D,AMX/RAMX,
U 0DC5, 0001,C03C,0180,3000,0070,0B03   ;27267          SET.CC(INST),J/STORE1
                                        ;27268
                                        ;27269  7CB:    ;-------------------------------;
                                        ;27270  MFP7:   LAB_R[R6],
U 07CB, 0001,C03C,0180,FA30,0070,0ABA   ;27271          ALU_D,SET.CC(INST),J/MFPX
                                        ;27272
                                        ;27273  7CC:    ;-------------------------------;
                                        ;27274  SXT7:   D_0,                            ;ASSUME POS, PREPARE ZERO TO STORE
U 07CC, 0F00,1A3C,0180,F800,0000,0C97   ;27275          PSL.CC?
```

```
                                    ;27276
                                    ;27277  =0111    ;-------------------------------;
                                    ;27278          ALU_D,SET.CC(INST),
U 0C97, 0001,C03C,0180,3000,0070,0B03 ;27279        CACHE_D[INST.DEP],J/STORE1
                                    ;27280
                                    ;27281          ;-------------------------------;
                                    ;27282          D_K[.FFFF],SET.CC(INST),
U 0C9F, 0818,C038,C180,F800,0070,0DC9 ;27283        J/STORE0
                                    ;27284
                                    ;27285  72F:    ;-------------------------------;
                                    ;27286  TCLAS7: ALU_D[INST.DEP]Q,               ;TST.B (DP1), BIT.B+CMP.B (DP2)
                                    ;27287          SET.CC(INST),
                                    ;27288          CLR.IB0-1,PC_PC+2,
U 072F, 401D,C90C,0180,F805,4070,0B09 ;27289        PC.MODES?,J/LOADIB
                                    ;27290
                                    ;27291  7AF:    ;-------------------------------;
                                    ;27292          ALU_D[INST.DEP]Q,               ;TST.B (DP1), BIT.B+CMP.B (DP2)
                                    ;27293          SET.CC(INST),
                                    ;27294          CLR.IB0-1,PC_PC+2,
U 07AF, 401D,C90C,0180,F805,4070,0B09 ;27295        PC.MODES?
                                    ;27296
                                    ;27297  =01     ;-------------------------------;
U 0B09, 0804,003B,65F0,F857,1284,6700 ;27298 LOADIB: IRD.11                         ;-(SM47,57+DM47,57)
                                    ;27299
                                    ;27300          ;-------------------------------;
U 0B0B, 001B,0000,09C0,F800,0000,0DC6 ;27301        Q_0-K[.2]
                                    ;27302
                                    ;27303          ;-------------------------------;
U 0DC6, 2015,2014,0180,F801,4200,0D86 ;27304        PC&VA_Q+PC,FLUSH.IB,J/GO.IRD    ;SM47,57+DM47,57:  ALU_PC-2
```

```
                                        ;27305  7CF:    ;--------------------------------;
                                        ;27306  SWAB7:  D_DAL.SC,ALU_D.AND.K[.FF00],    ;SWAP SRC TO BYTES 3 & 2
U 07CF, 0D19,4034,4D80,F800,0050,0DC8   ;27307          WORD,N&Z_ALU.V&C_0              ;SET CC ON RESULTING BYTE 0
                                        ;27308
                                        ;27309          ;--------------------------------;
U 0DC8, 0B00,003C,0180,F800,0000,0DC9   ;27310          D_D.SWAP,J/STORE0
                                        ;27311
                                        ;27312  7C7:    ;--------------------------------;
                                        ;27313  ROR15:  D_D.RIGHT,ALU_D,SET.PSL.C(AMX), ;
U 07C7, 0601,1A3C,6180,F800,00E4,6CAE   ;27314          SC_K[.F],PSL.CC?                ;BEN ON PSL<C>
                                        ;27315
                                        ;27316  =1110   ;--------------------------------;
                                        ;27317  ROR15X: D_D.AND.MASK,SET.CC(INST),      ;MASK OUT LMB OF BYTE/WORD
U 0CAE, 0801,C034,0180,F800,0070,0DC9   ;27318          J7STORE0
                                        ;27319
                                        ;27320          ;--------------------------------;
                                        ;27321          D_D.ORNOT.MASK,SET.CC(INST),    ;SET LMB OF BYTE/WORD
U 0CAF, 0801,C01C,0180,F800,0070,0DC9   ;27322          J7STORE0
                                        ;27323
                                        ;27324  7C8:    ;--------------------------------;
                                        ;27325  RORB7:  D_D.RIGHT,ALU_D,SET.PSL.C(AMX), ;
U 07C8, 0601,1A3C,5D80,F800,00E4,6CAE   ;27326          SC_K[.7],PSL.CC?,J/ROR15X       ;
                                        ;27327
                                        ;27328          ;--------------------------------;
U 0DC9, 0000,C03C,0180,3000,0000,0B03   ;27329  STORE0: CACHE_D[INST.DEP],J/STORE1
                                        ;27330
                                        ;27331  7CE:    ;--------------------------------;
U 07CE, 0000,003C,0180,F801,00C0,0B03   ;27332  JMP11:  PC_VA,J/STORE1
                                        ;27333
                                        ;27334  7CD:    ;--------------------------------;
                                        ;27335  JSR:    LAB_R[R6],D_Q,Q_PC,             ;
U 07CD, 0C14,0A38,01C0,FA30,0000,0F26   ;27336          SRC.PC?                         ;CHK IF SRC REG IS PC
                                        ;27337
                                        ;27338  =110    ;--------------------------------;
                                        ;27339          R6_LA-K[.2].RLOG,               ;SRC IS NOT PC
U 0F26, 0018,4004,0980,FAB0,0200,0DCA   ;27340          VAR/LOAD,J/JSR.0
                                        ;27341
                                        ;27342          ;--------------------------------;
                                        ;27343          D_Q,R6_LA-K[.2].RLOG,           ;SRC IS PC
U 0F27, 0C18,4004,0980,FAB0,0200,0DCA   ;27344          VAR/LOAD,J/JSR.0
                                        ;27345  =;END
                                        ;27346
                                        ;27347          ;--------------------------------;
                                        ;27348  JSR.0:  CACHE_D[WORD],CHK.ODD.ADDR,     ;PUSH DOWN STACK
                                        ;27349          LC_RC[T0],                      ;LATCH PC
U 0DCA, 0000,4A3C,0180,3100,0C00,0F2E   ;27350          SRC.PC?                         ;SRC IS PC?
                                        ;27351
                                        ;27352  =110    ;--------------------------------;
U 0F2E, 0010,4038,0180,F8C0,0000,0F2F   ;27353          R(SRC)_LC,WORD                  ;SRC IS NOT PC
                                        ;27354
                                        ;27355          ;--------------------------------;
                                        ;27356          PC&VA_RC[T7],                   ;SRC IS PC
U 0F2F, 2010,0038,0180,F939,4200,0D86   ;27357          FLUSH.IB,J/GO.IRD
                                        ;27358  =;END
```

```
                                        ;27359  .TOC "  Compatibility mode    : DP1 -  (-DM0)*(ASH+ASHC+MUL+DIV)"
                                        ;27360
                                        ;27361  ;HERE IS MEMORY TO REGISTER EXECUTES.
                                        ;27362  ;ENTER WITH D HAS DST OPERAND, VA CONTAINS DST OPERAND ADDRESS, AND
                                        ;27363  ;Q CONTAINS SRC OPERAND.
                                        ;27364
                                        ;27365  7AD:    ;-------------------------------;
                                        ;27366  ASH7:   SC_D.AND.K[.3F],
U 07AD, 0C19,0034,5580,F800,0082,0B12   ;27367          D_Q
                                        ;27368
                                        ;27369  =10     ;-------------------------------;
                                        ;27370  ASHU:   D_D.SXT[WORD],Q_ALU,            ;SIGN-EXTEND SRC TO D AND Q
                                        ;27371          N&Z_ALU.V&C_0,
U 0B12, 0802,543C,01C0,F800,0050,0F34   ;27372          SC?,J/ASHX
                                        ;27373
                                        ;27374          ;-------------------------------;
U 0B13, 0814,0038,0180,F800,0000,07AD   ;27375          D_PC,J/ASH7                     ;DST IS PC
                                        ;27376
                                        ;27377  =100    ;-------------------------------;
                                        ;27378  ASHX:   CLR.IB0-1,PC_PC+2,              ;SC = 0, CHK IF DM47,57
U 0F34, 4000,093C,0180,F805,4000,0B09   ;27379          PC.MODES?,J/[OADIB
                                        ;27380                                          ;NO SUCH CASE:  SC=NEG
                                        ;27381  =110    ;-------------------------------;
                                        ;27382          D_D.LEFT,SI/ZERO,               ;32 > SC > 0
U 0F36, 0500,003C,0580,F800,0084,AF47   ;27383          SC_SC-K[.1],J/ASHYY
                                        ;27384
                                        ;27385          ;-------------------------------;
                                        ;27386          D_D.RIGHT(B),SI/ASHR,           ;SC > 31, SET SC TO -SC
                                        ;27387          SC_SC-K[.3F],
U 0F37, 085C,2038,5480,F800,0084,AF3F   ;27388          J/ASHXX
                                        ;27389  =;END
                                        ;27390
                                        ;27391  =011    ;-------------------------------;
                                        ;27392  ASHX0:  R(SRC)_Q,WORD,
                                        ;27393          CLR.IB0-1,PC_PC+2,              ;END OF LOOP
U 0F3B, 4001,693C,0180,F8C5,4000,0B09   ;27394          PC.MODES?,J/[OADIB
                                        ;27395
                                        ;27396          ;-------------------------------;
                                        ;27397  ASHXX:  D_D.RIGHT(B),SI/ASHR,           ;ARITH SHIFT RIGHT
                                        ;27398          Q_Q.RIGHT,SI/ASHR,
                                        ;27399          RAMX/Q,AMX/RAMX,SET.CC(ROR),
                                        ;27400          SC_SC+1,
U 0F3F, 085D,2C38,00B0,F800,00C0,CF3B   ;27401          SC.NE.0?,J/ASHX0                ;CK IF END LOOP AND IF PC
                                        ;27402
                                        ;27403  =011    ;-------------------------------;
                                        ;27404  ASHY0:  CLR.IB0-1,PC_PC+2,
U 0F43, 4000,093C,0180,F805,4000,0B09   ;27405          PC.MODES?,J/[OADIB
                                        ;27406
                                        ;27407          ;-------------------------------;
                                        ;27408  ASHYY:  R(SRC)_D(B),                    ;ARITH SHIFT LEFT
                                        ;27409          D_D.LEFT,Q_Q.LEFT,SI/ZERO,
                                        ;27410          RAMX/Q,AMX7RAMX,
                                        ;27411          SET.CC(WORD),
                                        ;27412          SC_SC-K[.1],
U 0F47, 051D,6C38,05A8,F8C0,00F4,AF43   ;27413          SC.NE.0?,J/ASHY0
```

```
                                        ;27414  7AC:    ;--------------------------------;
                                        ;27415  ASHC7:  Q_D.AND.K[.3F],
U 07AC, 0C19,0034,55C0,F800,0000,0B1A   ;27416          D_Q
                                        ;27417
                                        ;27418  =10     ;--------------------------------;
                                        ;27419  ASHCU:  D_DAL.SC,
U 0B1A, 0D01,203C,0180,F800,0082,0DCB   ;27420          SC_Q,J/ASHCW
                                        ;27421
                                        ;27422          ;--------------------------------;
                                        ;27423          Q_D,D_PC,                       ;DST IS PC
U 0B1B, 0814,0038,01E0,F800,0000,07AC   ;27424          J/ASHC7
                                        ;27425
                                        ;27426          ;--------------------------------;
U 0DCB, 0018,0034,C1C0,F860,0000,0DCC   ;27427  ASHCW:  Q_R(SRC!1).AND.K[.FFFF]
                                        ;27428
                                        ;27429          ;--------------------------------;
                                        ;27430          D_D+Q,Q_ALU,                    ;D,Q_[SRC][SRC .OR. 1]
                                        ;27431          N&Z_ALU.V&C_0,
U 0DCC, 081D,1414,01C0,F800,0050,0F4C   ;27432          SC?,J/ASHCV
                                        ;27433
                                        ;27434  =100    ;--------------------------------;
                                        ;27435  ASHCV:  CLR.IB0-1,PC_PC+2,              ;SC = 0, CHK IF DM47,57
U 0F4C, 4000,093C,0180,F805,4000,0B09   ;27436          PC.MODES?,J/[OADIB
                                        ;27437                                          ;NO SUCH CASE:  SC = NEG
                                        ;27438  =110    ;--------------------------------;
                                        ;27439          D_D.LEFT,SI/ZERO,               ;SC > 0
U 0F4E, 0500,003C,0580,F800,0084,AF5F   ;27440          SC_SC-K[.1],J/ASHCY
                                        ;27441
                                        ;27442          ;--------------------------------;
                                        ;27443          D_D.RIGHT(B),SI/ASHR,           ;SC < 0, SET SC TO -SC
                                        ;27444          SC_SC-K[.3F],
U 0F4F, 085C,2038,5480,F800,0084,AF57   ;27445          J/ASHCX
                                        ;27446  =
                                        ;27447
                                        ;27448  =011    ;--------------------------------;
                                        ;27449  ASHCXX: D_Q,
U 0F53, CC00,003C,6D80,F800,0084,6DCD   ;27450          SC_K[.FFF0],J/ASHC70
                                        ;27451
                                        ;27452          ;--------------------------------;
                                        ;27453  ASHCX:  D_D.RIGHT(B),SI/ASHR,
                                        ;27454          Q_Q_RIGHT,SI/ASHR,
                                        ;27455          LONG,RAMX/Q,AMX/RAMX,SET.CC(ROR),
                                        ;27456          SC_SC+1,
U 0F57, 085D,2C38,00B0,F800,00C0,CF53   ;27457          SC.NE.0?,J/ASHCXX
                                        ;27458
                                        ;27459          ;--------------------------------;
U 0DCD, 0D00,003C,0180,F800,0000,0DCE   ;27460  ASHC70: D_DAL.SC
                                        ;27461
                                        ;27462          ;--------------------------------;
U 0DCE, 0001,403C,0180,F8C0,0000,0DD0   ;27463          R(SRC)_D,WORD
                                        ;27464
                                        ;27465          ;--------------------------------;
                                        ;27466          ALU_Q,R(SRC!1)_ALU,WORD,
                                        ;27467          CLR.IB0-1,PC_PC+2,
U 0DD0, 4001,693C,0180,F8E5,4000,0B09   ;27468          PC.MODES?,J/[OADIB
```

```
                                        ;27469
                                        ;27470  =011    ;-------------------------------;
                                        ;27471  ASHC71: D_Q,
U 0F5B, 0C00,003C,6D80,F800,0084,6DCD   ;27472          SC_K[.FFF0],J/ASHC70
                                        ;27473
                                        ;27474          ;-------------------------------;
                                        ;27475  ASHCY:  ALU_D(B),
                                        ;27476          RAMX/Q,AMX/RAMX,SET.CC(LONG),
                                        ;27477          D_D.LEFT,Q_Q.LEFT,SI/ZERO,
                                        ;27478          SC_SC-K[.1],
U 0F5F, 051D,2C38,05A8,F800,00F4,AF5B   ;27479          SC.NE.0?,J/ASHC71
```

```
                                ;27480  ;ENTER HERE WITH MULTIPLIER IN D, MULTIPLICAND IN Q.
                                ;27481  ;MUL_07ORSS -- D_(SS), Q_R
                                ;27482  ;NOTE THAT DST IS SS.
                                ;27483
                                ;27484  7A8:    ;-------------------------------;
                                ;27485  MUL7:   D_Q.SXT[WORD],Q_D,
                                ;27486          SC_K[.7],                       ;SET LOOP CT FOR 8
U 07A8, 0802,6D3C,5DE0,F800,0084,6F65   ;27487          D.NE.0?,J/MULU                  ;SIGN EXT M'CAND IN D
                                ;27488
                                ;27489  =101    ;-------------------------------;
U 0F65, 0F00,003C,0180,F800,0000,0DD3   ;27490  MULU:   D_0,J/MULZ                      ;PROD IS 0
                                ;27491
                                ;27492          ;-------------------------------;
                                ;27493          R[R15]_D,LONG,
U 0F67, 0001,0D3C,0180,FAF8,0000,0F6D   ;27494          D.NE.0?
                                ;27495
                                ;27496  =101    ;-------------------------------;
U 0F6D, 0000,003C,01F8,F800,0000,0DD3   ;27497          Q_0,J/MULZ                      ;PROD IS 0
                                ;27498
                                ;27499          ;-------------------------------;
                                ;27500          RC[T0]_D.LEFT,                  ;2 TIMES M'CAND IN RC<0>
                                ;27501          SI/ZERO,LONG,
U 0F6F, 0C21,003C,01E0,F980,0000,08C8   ;27502          D_Q,Q_D
                                ;27503  =
                                ;27504
                                ;27505  =0*     ;-------------------------------;
                                ;27506          LAB_R[R15],LONG,
                                ;27507          Q_0,                            ;CLR PARTIAL PROD,
U 08C8, 0000,0D3D,01F8,FA78,0000,0F7B   ;27508          CALL,SIGNS?,J/MULV              ;CHECK SIGN OF M'IER
                                ;27509
                                ;27510          ;-------------------------------;
U 08CA, 0000,0D3C,0180,F800,0000,0F76   ;27511          D31?                            ;CK D31 FOR 2**15 RANGE
                                ;27512
                                ;27513  =110    ;-------------------------------;
                                ;27514          R(SRC)_Q,WORD,                  ;STORE HIGH WORD
                                ;27515          SET.CC(WORD),
U 0F76, 0D01,603C,0180,F8C0,0070,0DD2   ;27516          D_DAL.SC,J/MUL7P                ;POS PROD, GET HIGH WORD TO D
                                ;27517
                                ;27518          ;-------------------------------;
                                ;27519          ALU_Q+K[.1],SET.CC(INST),       ;NEG PROD
                                ;27520          D_DAL.SC,                       ;GET LOW WORD TO D
U 0F77, 0D19,E014,0580,F800,0070,0DD1   ;27521          J/MUL7M
                                ;27522  =
                                ;27523
                                ;27524  =011    ;-------------------------------;
                                ;27525  MULV:   LC_RC[T0],
                                ;27526          ALU_0(A),D_D.RIGHT2,SI/ZERO,    ;D_D_0 EXT SHIFT RIGHT BY 2
U 0F7B, 0203,0C3C,0180,F900,0000,0350   ;27527          MUL?,J/MULPP                    ;Q31=0: POS M'CAND
                                ;27528
                                ;27529          ;-------------------------------;
                                ;27530          LC_RC[T0],
                                ;27531          ALU_0(A),D_D.RIGHT2,SI/ZERO,    ;D_D_0 EXT SHIFT RIGHT BY 2
U 0F7F, 0203,0C3C,0180,F900,0000,03B0   ;27532          MUL?,J/MULMP                    ;Q31=1: NEG M'CAND
                                ;27533  =
                                ;27534
```

```
                                        ;27535          ;--------------------------------;
U 0DD1, 0001,603C,0180,F8C0,0000,0DD2   ;27536  MUL7M:  R(SRC)_Q,WORD
                                        ;27537
                                        ;27538          ;--------------------------------;
                                        ;27539  MUL7P:  R(SRC!1)_D<8),WORD,              ;STORE LOW WORD OF PRODUCT
                                        ;27540          RAMX/Q,AMX/RAMX,N_AMX.Z_TST,
                                        ;27541          CLR.IB0-1,PC_PC+2,
U 0DD2, 401D,6938,0180,F8E5,4030,0809   ;27542          PC.MODES?,J/LOADIB
                                        ;27543
                                        ;27544          ;--------------------------------;
U 0DD3, 0001,E03C,0180,F8C0,0070,0DD2   ;27545  MUL2:   R(SRC)_Q,SET.CC(INST),J/MUL7P
```

```
                                ;27546  ;HERE IS MEMORY TO REGISTER EXECUTES.
                                ;27547  ;ENTER WITH D HAS DST OPERAND (DIVISOR), VA CONTAINS DST OPERAND ADDRESS, AND
                                ;27548  ;Q CONTAINS SRC OPERAND (DIVIDEND).
                                ;27549  ;
                                ;27550  ;THE DIVISION USES RESTORING ALGORITHM.
                                ;27551  ;LB IS SET TO DIVISOR (D) VIA R[R15], AND SC IS THE LOOP COUNT.
                                ;27552  ;DURING DIVIDING, D HAS THE PARTIAL DIVIDEND, AND Q HAS THE PARTIAL QUOTIENT.
                                ;27553  ;WHEN DONE, D CONTAINS THE REMAINDER, AND Q HAS THE QUOTIENT.
                                ;27554
                                ;27555  7AE:    ;-------------------------------;
                                ;27556  DIV7:   RC[TO]_Q.SXT[WORD],              ;SIGN EXT D'END
U 07AE, 0002,603C,65F8,F980,0184,6DD4 ;27557            Q_0,FE&SC_K[.10]        ;SET SHF CT
                                ;27558
                                ;27559          ;-------------------------------;
                                ;27560  DIVX:   Q_RC[TO],
U 0DD4, 0D10,0038,01C0,F900,0000,0DD5 ;27561            D_DAL.SC
                                ;27562
                                ;27563          ;-------------------------------;
U 0DD5, 0000,0D3C,0180,F800,0000,0F84 ;27564            SIGNS?                  ;D'SOR = +?
                                ;27565
                                ;27566  =100    ;-------------------------------;
U 0F84, 0C00,003C,01F8,F800,0000,0DE0 ;27567            D_Q,Q_0,J/NODIV         ;D'SOR = 0
                                ;27568
                                ;27569  =110    ;-------------------------------;
                                ;27570          R[R15]_D,LONG,                  ;D'SOR = +
U 0F86, 0C01,003C,01F8,FAF8,0000,0DD8 ;27571            D_Q,Q_0,J/DIVP
                                ;27572
                                ;27573          ;-------------------------------;
                                ;27574          R[R15]_0-D,LONG,                ;D'SOR = -
U 0F87, 0C1F,2000,01F8,FAF8,0000,0DD9 ;27575            D_Q,Q_0,J/DIVM
                                ;27576
                                ;27577          ;-------------------------------;
                                ;27578  DIVP:   D_DAL.SC,
                                ;27579          Q_R(SRC!1).AND.K[.FFFF],
U 0DD8, 0D18,0D34,C1C0,F860,0000,0CB6 ;27580            SIGNS?                  ;D'END = +?
                                ;27581
                                ;27582  =0110   ;-------------------------------;
                                ;27583          ID[TO]_D,                       ;+/+    SAVE D'END SIGN
                                ;27584          D_D[OR]Q,
                                ;27585          Q_0,LAB_R[R15],
U 0CB6, 081D,0031,C1F8,3E78,0000,01C7 ;27586            CALL,J/DIV0X
                                ;27587
                                ;27588          ;-------------------------------;
                                ;27589          ID[TO]_D,                       ;-/+    SAVE D'END SIGN
                                ;27590          D_D[OR]Q,
U 0CB7, 081D,0030,C1F8,3E78,0000,0723 ;27591            Q_0,LAB_R[R15],J/DIVMP
                                ;27592
                                ;27593          ;-------------------------------;
                                ;27594          ALU_D-K[.8000],CLK.UBCC,
                                ;27595          SC_FE,                          ;SC GETS 16.
U 0CBE, 0019,0000,4580,F800,0091,0DDC ;27596            J/DIVQ
                                ;27597  =;END
```

```
                                ;27598  =0***   ;0***------------------------------;
U 0723, 081F,2001,0180,F800,0000,01D7   ;27599  DIVMP:  D_0-D,CALL,J/DIV1X              ;
                                ;27600
                                ;27601          ;1***------------------------------;
                                ;27602          ALU_D-K[.8000]-1,CLK.UBCC,SC_FE,;
U 072B, 0019,0008,4580,F800,0091,0DDA   ;27603          J/DIVQN                         ;-/+: -QUOT, OK FOR MOST NEG #
                                ;27604
                                ;27605  DIVM:   ;--------------------------------;
                                ;27606          D_DAL.SC,                       ;D'END = +?
U 0DD9, 0D18,0D34,C1C0,F860,0000,0CC6   ;27607          Q_R(SRC!1).AND.K[.FFFF],SIGNS?  ;
                                ;27608
                                ;27609  =0110   ;0110-------------------------------;
                                ;27610          D_D[OR]Q,                       ;+/-
U 0CC6, 081D,0031,01F8,FA78,0000,01C7   ;27611          Q_0,LAB_R[R15],CALL,J/DIV0X     ;
                                ;27612
                                ;27613          ;0111------------------------------;
U 0CC7, 081D,0030,01F8,FA78,0000,0766   ;27614          D_D[OR]Q,Q_0,LAB_R[R15],J/DIVMM ;-/-
                                ;27615
                                ;27616          ;1110------------------------------;
                                ;27617          ALU_D-K[.8000]-1,CLK.UBCC,SC_FE,;
U 0CCE, 0019,0008,4580,F800,0091,0DDA   ;27618          J/DIVQN                         ;+/-: -QUOT, OK FOR MOST NEG #
                                ;27619  =
                                ;27620  =0***   ;0***------------------------------;
U 0766, 081F,2001,0180,F800,0000,01D7   ;27621  DIVMM:  D_0-D,CALL,J/DIV1X              ;
                                ;27622
                                ;27623          ;1***------------------------------;
                                ;27624          ALU_D-K[.8000],CLK.UBCC,SC_FE,  ;SC GETS 16.
U 076E, 0019,0000,4580,F800,0091,0DDC   ;27625          J/DIVQ                          ;
                                ;27626
                                ;27627          ;--------------------------------;
U 0DDA, 081F,2000,0180,F800,0000,0DDC   ;27628  DIVQN:  D_0-D                           ;NEG QUOTIENT
                                ;27629
                                ;27630  DIVQ:   ;--------------------------------;
                                ;27631          R(SRC)_D,WORD,                  ;QUOTIENT TO R
U 0DDC, 0D01,5B3C,018C,F8C0,0050,0CD7   ;27632          N&Z_ALU.V&C_0,D_DAL.SC,ALU.N?   ;
                                ;27633
                                ;27634  =0111   ;0111------------------------------;
                                ;27635          ALU_D,R(SRC!1)_ALU,WORD,        ;OVERFLOW:  Q GETS D'END SIGN
U 0CD7, 0001,403C,C1F0,2CE0,0000,0DDD   ;27636          Q_ID[T0],J/DIVQ.1               ;
                                ;27637
                                ;27638          ;1111------------------------------;
                                ;27639          ALU_D,R(SRC!1)_ALU,WORD,PC_PC+2,;REMAINDER TO R.OR.1
U 0CDF, 4001,493C,0180,F8E5,4000,0809   ;27640          CLR.IB0-1,PC.MODES?,J/LOADIB    ;
                                ;27641
                                ;27642  DIVQ.1: ;--------------------------------;
U 0DDD, 0F10,0038,0180,F900,0070,0DDE   ;27643          D_0,ALU_RC[T0],SET.CC(LONG)     ;SET PSL <N,Z,V,C> TO <N,1,1,1>
                                ;27644
                                ;27645          ;--------------------------------;
                                ;27646          ALU_D,SET.PSL.C(AMX),PC_PC+2,   ;SET PSL TO <N,1,1,0>
U 0DDE, 4001,093C,0180,F805,4060,0809   ;27647          CLR.IB0-1,PC.MODES?,J/LOADIB    ;
                                ;27648
                                ;27649          ;--------------------------------;
                                ;27650  NODIV:  ALU_RC[T0],SET.CC(LONG),PC_PC+2,;SET PSL <N,Z,V,C> TO <N,1,1,1>
U 0DE0, 4010,0938,0180,F905,4070,0809   ;27651          CLR.IB0-1,PC.MODES?,J/LOADIB    ;
```

```
                                ;27652  .CHANGE/NATIVE=1
                                ;27653  .LIST           ;Re-enable full listing
```

```
                                        ;27654  .TOC    "IE.MIC"
                                        ;27655  .TOC    "Revision 1.7"
                                        ;27656  ;        P. R. Guilbault
                                        ;27657
;27658  .NOBIN
;27659  .TOC    "       Revision History"
;27660
;27661  ; 01    Remove absolute jumps.
;27662  ;       Add label for Patch no. 094
;27663  ;       Change macro names that deal with conditions codes.
;27664  ; 00    Delete CMHALT expression. Was use originally for debug and not required.
;27665  ;       Start of history
;27666
                                        ;27667  .BIN
                                        ;27668  .NOLIST         ;Disable listing of PCS code for quickie assemblies
```

```
;27669  .TOC    ""      Interrupt & Exception : Description"
;27670
;27671
;27672  ;THIS IS THE MAIN ROUTINE TO INITIALIZE INTERRUPTS AND EXCEPTIONS.
;27673  ;ENTER THE CALL TO 'EXCPTN" IF THERE ARE PARAMETERS TO BE PUSHED DOWN INTO
;27674  ;STACK, JUMP TO 'EXCPT" IF THERE IS NO PARAMETER TO BE PUSHED DOWN,
;27675  ;FOR EXCEPTIONS.  THE TRAP CODES COULD BE IN RC(T1) IF ANY.
;27676  ;
;27677  ;FOR INTERNAL INTERRUPTS, ENTER AT "INTRPT".  "INTIO" IS EXTERNAL
;27678  ;INTERRUPT ENTRY.
;27679  ;
;27680  ;THE ROUTINE WILL UPDATE SP'S IF NECESSARY, PUSH DOWN OLD PSL
;27681  ;AND PC, CREATE NEW PSL. FOR EXCEPTIONS WITH PARAMETERS TO BE
;27682  ;PUSHED DOWN IN THE STACK, RETURN 1, WITH TRAP CODE (IF ANY) IN
;27683  ;Q REGISTER, SP LATCHED AT LA, VECTOR IN RC(TO).
;27684  ;FOR OTHER EXCEPTIONS AND INTERRUPTS, PC WILL BE GENERATED, PSL[FPD]
;27685  ;WILL BE CLEARED, AND GO TO IRD.
;27686
;27687  ;ENTER 'EXCPT2" FOR EXCEPTIONS AFTER PUSHING DOWN PARAMETERS TO STORE THE
;27688  ;LAST PARAMETER, CLEAR PSL[FPD] BIT AND GENERATE NEW PC.
```

```
                                        ;27689  .TOC     "       Interrupt & Exception : INTIO, INTRPT, EXCPTN, EXCPT"
                                        ;27690
                                        ;27691  =101
                                        ;27692  INTIO:  ;----------------------;  *** ENTER HERE FOR INTERRUPTS OF IPL 14-17 ***
                                        ;27693          ALU_0(A), VAK/LOAD,     ;  LOAD 0 TO VA
U 0F8D, 0003,003C,0180,D800,0200,0DE4   ;27694          D_INT.SUM,J/IE.2        ;  D GETS INTERRUPT SUMMARY REGISTER (ISR)
                                        ;27695
                                        ;27696  INTRPT: ;----------------------;  ENTER HERE FOR OTHER INTERRUPTS
U 0F8F, 0000,003C,35F0,2C00,0000,0DE1   ;27697          Q_ID[VECTOR]            ;  Q GETS INTERRUPT VECTOR
                                        ;27698  =;END
                                        ;27699
                                        ;27700          ;----------------------;
                                        ;27701          RC[T0]_Q.OXT[WORD],     ;  RC[T0] GETS VECTOR FOR INDEXING INTO SCB
U 0DE1, 0003,603C,3DF0,2D80,0000,0DE2   ;27702          Q_ID[PSL]               ;  Q GETS PSL FOR SAVING ON STACK
                                        ;27703
                                        ;27704          ;----------------------;
                                        ;27705          LC_RC[T0], STATE_K[.1], ;  LATCH VECTOR, PREPARE TO SET STATE TO 2
U 0DE2, 0000,003C,0580,F900,1404,6DE6   ;27706          J/IE.3                  ;  AS WE JOIN EXCEPTION CODE
                                        ;27707
                                        ;27708  IE.2:   ;----------------------;
                                        ;27709          ID[VECTOR]_D,           ;  VECTOR REG GETS ISR
U 0DE4, 0000,003C,3580,3C00,0000,0F8F   ;27710          J/INTRPT                ;
                                        ;27711
                                        ;27712  EXCPTN: ;----------------------;  *** ENTER HERE FOR EXCEPTIONS WITH PARAMETERS ***
                                        ;27713          Q_ID[PSL],              ;  Q GETS PSL
                                        ;27714          STATE_0(A),             ;  CLEAR STATE REG VIA AMX_0
U 0DE5, 0003,003C,3DF0,2D00,1408,6DE6   ;27715          LC_RC[T0]               ;  LATCH VECTOR ID IN LC
                                        ;27716
                                        ;27717          ;----------------------;
                                        ;27718  IE.3:    D_Q,RC[T2]_Q,           ;  D AND RC2 GET PSL
                                        ;27719          Q_ID[SCBB],             ;  Q GETS SCBB, S/W CONTROL BLOCK BASE REG
                                        ;27720          STATE_STATE+1,          ;  SET STATE REG TO 1 FOR PARAMETERS
U 0DE6, 0C01,203C,EDF0,2D90,1400,CDEA   ;27721          J/IE.4                  ;  GO TO SET VECTOR ADDRESS
```

```
                                ;27722  =0*
                                ;27723  EXCPT:  ;0----------------------;  *** ENTER HERE FOR EXCEPTIONS W/O PARAMETERS ***
                                ;27724          RC[PC.SV]_PC,           ;  PUT PC WHERE BAKUP.PC EXPECTS IT
                                ;27725          CALL,RLOG.EMPTY?,       ;  BACK UP R'S AND PC
U 08FC, 0014,1539,0180,F9E0,0000,0DA7   ;27726          J/BAKUP.RGS             ;
                                ;27727
                                ;27728  EXCPT0: ;1----------------------;
                                ;27729          Q_ID[PSL],              ;  Q GETS PSL
U 08FE, 0003,003C,3DF0,2C00,1408,6DE8   ;27730          STATE_0(A)              ;  CLEAR STATE REG VIA AMX_0 FOR NO PARAMETERS
                                ;27731  =;END
                                ;27732
                                ;27733  EXCPT1: ;----------------------;
                                ;27734          SC_K[.1E],              ;  SET SC FOR CLEARING TP OF PSL
U 0DE8, 0000,003C,5180,F900,0084,6DE9   ;27735          LC_RC[T0]               ;  LATCH VECTOR ID IN LC
                                ;27736
                                ;27737          ;----------------------;
                                ;27738          D_Q.AND.MASK,           ;  CLEAR TP OF PSL
                                ;27739          RC[T2]_ALU,             ;  D AND RC2 GET PSL
U 0DE9, 0801,2034,EDF0,2D90,0000,0DEA   ;27740          Q_ID[SCBB]              ;  Q GETS SCBB, S/W CONTROL BLOCK BASE REG
                                ;27741
                                ;27742  IE.4:   ;----------------------;
                                ;27743          D_D.SWAP,               ;  PSL (CURRENT MODE) IN D[1:0]
                                ;27744          VA_Q+LC,                ;  VA GETS VECTOR ADDR (BASE + VECTOR ID)
                                ;27745          FE_K[.1A],              ;  FE GETS 26. FOR POSSIBLE SETTING PSL<IS>
                                ;27746          LAB_R[SP],              ;  LATCH SP IN LA, LB
                                ;27747          MSC7INH.CM.ADDR,        ;  ALLOW VAX MODE ADDR EVEN PSL <CM> IS SET
U 0DEA, 0B11,3C14,E580,FA70,3F04,611C   ;27748          PSL.MODE?               ;  CHECK PSL<IS> BITS
                                ;27749  =11100
                                ;27750          ;00----------------------;  PSL<IS>=0: NEED TO UPDATE SP'S
                                ;27751          TRAP.ACC[1],            ;  ABORT FPA
                                ;27752          SC_D.AND.K[.3],         ;  SC GETS CURRENT MODE
U 011C, 0019,00B4,0C80,F800,0082,0DED   ;27753          J/IE.9                  ;  CHK CURRENT MODE TO UPDATE SP'S
                                ;27754
                                ;27755  ; ************************************************
                                ;27756  ; * Patch no. 028, PCS 011C trapped to WCS 1160 *
                                ;27757  ; ************************************************
                                ;27758
                                ;27759          ;01----------------------;  PSL<IS>=1: NO NEED TO UPDATE SP'S
                                ;27760          TRAP.ACC[1],            ;  ABORT FPA
                                ;27761          MSC/INH.CM.ADDR,        ;  ALLOW VAX MODE ADDR EVEN PSL <CM> IS SET
                                ;27762          D[LONG]_CACHE.P,        ;  GET VECTOR 32 BITS
U 011D, 0000,00BC,0080,C800,3C00,0DEC   ;27763          J/IE.6                  ;  GOTO SAVE VECTOR
                                ;27764
                                ;27765          ;10----------------------;  PSL<IS>=0: NEED TO UPDATE SP'S
                                ;27766          TRAP.ACC[1],            ;  ABORT FPA
                                ;27767          SC_D.AND.K[.3],         ;  SC GETS CURRENT MODE
U 011E, 0019,00B4,0C80,F800,0082,0DED   ;27768          J/IE.9                  ;  CHK CURRENT MODE TO UPDATE SP'S
```

```
                                        ;27769          ;11--------------------;  PSL<IS>=0: NEED TO UPDATE SP'S
                                        ;27770          TRAP.ACC[1],           ;  ABORT FPA
                                        ;27771          SC_D.AND.K[.3],        ;  SC GETS CURRENT MODE
U 011F, 0019,0084,0C80,F800,0082,0DED   ;27772          J/IE.9                 ;  CHK CURRENT MODE TO UPDATE SP'S
                                        ;27773  =;END
                                        ;27774  IE.6:   ;--------------------;
                                        ;27775          R[R15]_D,              ;  SAVE VECTOR IN R15
U 0DEC, 0001,0C3C,0180,FAF8,0000,0F94   ;27776          MUL?                   ;  CHK VECTOR [01:00] FOR OPERATION ID
                                        ;27777  =100
                                        ;27778  IE.7:   ;00--------------------;  USE ISP
                                        ;27779          R[SP]&VA_LA-K[.4],     ;  SP_ISP-4
U 0F94, 0018,1700,1180,FAF0,0200,0109   ;27780          STATE1?,J/IE.8         ;  IS THIS INT OR EXCEPT?
                                        ;27781
                                        ;27782          ;01--------------------;  USE ISP
                                        ;27783          R[SP]&VA_LA-K[.4],     ;  SP_ISP-4
U 0F95, 0018,1700,1180,FAF0,0200,0109   ;27784          STATE1?,J/IE.8         ;  IS THIS INT OR EXCEPT?
                                        ;27785
                                        ;27786  ; ************************************************
                                        ;27787  ; * Patch no. 043, PCS 0F95 trapped to WCS 116B *
                                        ;27788  ; ************************************************
                                        ;27789
                                        ;27790          ;10--------------------;  WCS HANDLING
U 0F96, 0000,003D,0180,F800,0000,10E0   ;27791          CALL,J/USRTRP          ;  INVOKE USER WCS ROUTINE, IF ANY
                                        ;27792
                                        ;27793          ;11--------------------;  WILL HALT AFTER SERVICING THE EXCT
                                        ;27794          STATE_STATE+K[.8],     ;  SET FLAG FOR LATTER HALT
U 0F97, 0000,003C,0180,F800,1404,8F94   ;27795          J/IE.7                 ;
                                        ;27796  =;END
                                        ;27797
                                        ;27798  =1*01   ;0--------------------;  EXCEPTIONS
                                        ;27799  IE.8:   EXCEPT.ACK,            ;  IEK/EACK, EXCEPTION ACK TO RESET PSL
                                        ;27800          D_RC[T2],              ;  D GETS PSL
U 0109, 0810,0038,0180,F910,C000,0B20   ;27801          J7CALLFPD              ;  GOTO CALL SET FPD ROUTINE
                                        ;27802
                                        ;27803          ;1--------------------;  INTERRUPTS
                                        ;27804          INTRPT.ACK,            ;  IEK/IACK, INTERRUPT ACK TO RESET PSL
                                        ;27805          D_RC[T2],              ;  D GETS PSL
U 010B, 0810,0038,0180,F910,8000,0B20   ;27806          J7CALLFPD              ;  GOTO CALL SET FPD ROUTINE
                                        ;27807  =;END
                                        ;27808
                                        ;27809          ;--------------------;  UPDATE SP'S
                                        ;27810  IE.9:   MSC/INH.CM.ADDR,       ;  ALLOW VAX MODE ADDR EVEN PSL <CM> IS SET
                                        ;27811          D[LONG]_CACHE.P,       ;  GET VECTOR 32 BITS
U 0DED, 0000,003C,01C0,C800,3C00,0DEE   ;27812          Q_LA                   ;  Q GETS SP
                                        ;27813
                                        ;27814          ;--------------------;
                                        ;27815          R[R15]_D,              ;  SAVE VECTOR IN R15
                                        ;27816          D_Q,                   ;  D GETS SP
U 0DEE, 0C01,003C,2D80,FAF8,0084,8DF0   ;27817          SC_SC+K[.28]           ;  SC GETS ID ADDRESS FOR CORRECT SP
                                        ;27818
                                        ;27819          ;--------------------;
                                        ;27820          ID(SC)_D,              ;  UPDATE SP
U 0DF0, 0000,153C,0180,3400,0000,0CEC   ;27821          ALU1-0?                ;  CHK VECTOR [01:00] FOR OPERATION ID
```

```
                                        ;27822  =1100
                                        ;27823  IE.10:  ;00--------------------;
                                        ;27824          Q_ID[KSP],              ;  USE KSP
                                        ;27825          D_RC[T2],               ;  D GETS CURRENT PSL
U 0CEC, 0810,1738,A1F0,2D10,0000,0149   ;27826          STATE1?,J/IE.11         ;  INTERRUPTS OR EXCEPTIONS?
                                        ;27827
                                        ;27828          ;01--------------------;
                                        ;27829          Q_ID[ISP],              ;  USE ISP
                                        ;27830          D_RC[T2],               ;  D GETS CURRENT PSL
U 0CED, 0810,1738,B1F0,2D10,0000,01D9   ;27831          STATE1?,J/IE.13         ;  INTERRUPTS OR EXCEPTIONS?
                                        ;27832
                                        ;27833  ; ************************************************
                                        ;27834  ; * Patch no. 044, PCS 0CED trapped to WCS 116C *
                                        ;27835  ; ************************************************
                                        ;27836
                                        ;27837          ,10--------------------;  WCS HANDLING
U 0CEE, 0000,003D,0180,F800,0000,10E0   ;27838          CALL,J/USRTRP           ;  INVOKE USER WCS ROUTINE, IF ANY
                                        ;27839
                                        ;27840          ;11--------------------;  WILL HALT AFTER SERVICING THE EXCT
                                        ;27841          STATE_STATE+K[.8],      ;  SET FLAG FOR LATTER HALT
U 0CEF, 0000,003C,0180,F800,1404,8CEC   ;27842          J/IE.10                 ;
                                        ;27843  =;END
                                        ;27844
                                        ;27845  =1*01
                                        ;27846  IE.11:  ;0---------------------;  EXCEPTIONS
                                        ;27847          EXCEPT.ACK,             ;  EXCEPTION ACK TO RESET PSL
                                        ;27848          R[SP]&VA_Q-K[.4],       ;  SP, VA_SP-4
U 0149, 0019,2000,1180,FAF0,C200,0B20   ;27849          J/CALLFPD               ;  GOTO CALL FPD ROUTINE WITH EXCEPT STACK SETUP
                                        ;27850
                                        ;27851          ;1---------------------;  INTERRUPTS
                                        ;27852          INTRPT.ACK,             ;  INTERRUPT ACK TO RESET PSL
                                        ;27853          R[SP]&VA_Q-K[.4],       ;  SP, VA_SP-4
U 014B, 0019,2000,1180,FAF0,8200,0B20   ;2785-          J/CALLFPD               ;  INTERRUPTS TO KERNEL STACK ARE STRANGE BUT LEGAL
                                        ;27855  =;END
                                        ;27856
                                        ;27857  =00
                                        ;27858  CALLFPD:;00--------------------;  CALL FPD ROUTINE WITH INTRUPT STACK SETUP
                                        ;27859          RC[PC.SV]_PC,           ;  SAVE PC
                                        ;27860          MSC/SET.NEST.ERR,       ;  SET NESTED ERR FLAG FOR POSSIBLE MM USE
U 0B20, 0014,0039,0180,F9E0,2C00,0E16   ;27861          CALL,J/SETFPD
                                        ;27862
                                        ;27863          ;01--------------------;  INTRUPT STACK/KERNEL STACK NOT VALID
                                        ;27864          D_K[.1F00],Q_0,         ;  GUESS KSNV,SETUP TO RAISE IPL TO HIGHEST PRIORITY
U 0B21, 0818,1C38,91F8,F800,0000,015D   ;27865          PSL.MODE?,J/SNV         ;  GOTO INTERRUPT/KERNEL STACK NOT VALID
                                        ;27866
                                        ;27867  =11     ;11--------------------;
                                        ;27868          CACHE_D[LONG],          ;  PUSH DOWN OLD PSL
                                        ;27869          Q_PC,                   ;  MOVE PC TO BE READY FOR PUSHING INTO STACK
                                        ;27870          LAB_R[SP],              ;  LATCH SP
                                        ;27871          SC_R[.1B],              ;  SETUP FOR CLEARING FPD BIT OF PSL
U 0B23, 0014,0D38,EDC0,3270,0084,6F9E   ;27872          D3T?,J/IE.14            ;  IF CM, PUSH PC-2 INSTEAD OF PC
                                        ;27873  =;END
                                        ;27874
```

```
                                ;27875  =1*01
                                ;27876  IE.13:  ;0----------------------;  EXCEPTIONS
                                ;27877          EXCEPT.ACK,             ;  EXCEPTION ACK TO RESET PSL
                                ;27878          R[SP]&VA_Q-K[.4], SC_FE,;  SP&VA GET SP-4, SC GETS 26
U 01D9, 0019,2000,1180,FAF0,C281,0DF1   ;27879          J/IE.13A        ;
                                ;27880
                                ;27881          ;1----------------------;  INTERRUPTS
                                ;27882          INTRPT.ACK,             ;  INTERRUPT ACK TO RESET PSL
U 01DB, 0019,2000,1180,FAF0,8281,0DF1   ;27883          R[SP]&VA_Q-K[.4], SC_FE ;  SP&VA GET SP-4, SC GETS 26
                                ;27884  =;END
                                ;27885
                                ;27886  IE.13A: ;----------------------;  TO SET PSL <IS>
U 0DF1, 0000,003C,3DF0,2C00,0000,0DF2   ;27887          Q_ID[PSL]               ;  GET NEW PSL
                                ;27888
                                ;27889          ;----------------------;
                                ;27890          Q_D,                    ;  Q GETS OLD PSL
U 0DF2, 0801,201C,01E0,F800,0000,0DF4   ;27891          D_Q[ORNOT]MASK          ;  D HAS NEW PSL
                                ;27892
                                ;27893          ;----------------------;
                                ;27894          ID[PSL]_D,              ;  UPDATE NEW PSL WITH <IS> BIT SET
                                ;27895          D_Q,                    ;  OLD PSL TO D
U 0DF4, 0C00,003C,3D80,3C00,0000,0B20   ;27896          J7CALLFPD               ;  GOTO CALL FPD ROUTINE WITH INT STACK SETUP
                                ;27897
                                ;27898  =110
                                ;27899  IE.14:  ;0----------------------;
                                ;27900          D_Q,                    ;  D GETS OLD PC
                                ;27901          R[SP]&VA_LA-K[.4],      ;  SP_SP-4
U 0F9E, 0C18,0000,1180,FAF0,0200,0DF5   ;27902          J/IE.15                 ;  GOTO DO PC - 1
                                ;27903
                                ;27904          ;1----------------------;  CM:  DECREMENT PC BY 2
                                ;27905          Q_Q.OXT[WORD]-K[.1],    ;  PC - 2, DO PC - 1 FIRST
U 0F9F, 001B,6000,05C0,F800,0000,0F9E   ;27906          J7IE.14                 ;
                                ;27907  =;END
                                ;27908  IE.15:  ;----------------------;
                                ;27909          D_D-K[.1],              ;  PC - 2 FOR COMP MODE, PC - 1 FOR VAX MODE
U 0DF5, 0819,1700,0580,F800,0000,002A   ;27910          STATE0?                 ;  IS THERE PARAMETERS TO BE PUSHED IN?
                                ;27911  =1*10
                                ;27912  EXCPT2: ;0----------------------;  *** NO PARAMETERS, OR 2ND ENTRY FOR PARAMETERS
                                ;27913          CACHE_D[LONG],          ;  PUSH DOWN OLD PC
                                ;27914          FE_K[.10],              ;  STORE 16. FOR LATER SHIFT WORD ON SIR
U 002A, 0000,003C,6580,3000,0104,6CF7   ;27915          J/EXCPT3                ;  GOTO GET NEW PC
                                ;27916
                                ;27917          ;1----------------------;  PARAMETERS
                                ;27918          CACHE_D[LONG],          ;  PUSH DOWN OLD PC
                                ;27919          LAB_R[SP],              ;  LATCH SP INTO LA, LB
                                ;27920          FE_K[.10],              ;  STORE 16. FOR LATER SHIFT WORD ON SIR
U 002B, 0000,003E,6580,3270,0104,6001   ;27921          RETURN1                 ;  RETURN TO PUSH DOWN PARAMETERS
                                ;27922  =;END
                                ;27923
```

```
                                    ;27924  =0111   ;----------------------;  2ND ENTRY FOR PARAMETERS
                                    ;27925  EXCPT3: Q_ID[CES],             ;NOW, CLEAR ARITH TRAP CODE, IF ANY
U OCF7, 0000,003D,31F0,2C00,0000,0E0B ;27926        CALL,J/CLR.CES
                                    ;27927
                                    ;27928          ;----------------------;RETURN FROM CLEARING CES
                                    ;27929          Q_ID[SIR], D_0,        ;  GET SIR
                                    ;27930          SC_FE,                 ;  SC GETS 16. FOR SWAP WORD OF SIR TO GET IPRA
                                    ;27931          CLR.FPD,               ;  CLR PSL<FPD> BIT
U OCFF, 0F00,173C,39F0,2C00,2081,0331 ;27932        STATE3-0?              ;  IS IT INTRPT OR EXCEPTION?
                                    ;27933
                                    ;27934  =0*01   ;00------------------;  EXCEPTION, DO NOT CLEAR IPR BITS IN SIR
                                    ;27935          D_DAL.SC,              ;  GET IPRA AT D[04:00]
                                    ;27936          PC&VA_R[R15].ANDNOT.K[.3], ;GET PC FROM VECTOR[31:02], WITH PC[1:0] SET TO 0
                                    ;27937          FLUSH.IB,              ;  FLUSH IB, LOAD VA
U 0331, 2D18,0024,0D80,FA79,4200,0DFB ;27938        J/IE.20                ;  GOTO BYPASS CLEARING IPR BIT
                                    ;27939
                                    ;27940          ;01------------------;  INTERRUPTS, CLR HIGHEST BIT OF IPR OF SIR
                                    ;27941          D_DAL.SC,              ;  GET IPRA AT D[04:00]
                                    ;27942          PC&VA_R[R15].ANDNOT.K[.3], ;GET PC FROM VECTOR[31:02], WITH PC[1:0] SET TO 0
                                    ;27943          FLUSH.IB,              ;  FLUSH IB, LOAD VA
U 0333, 2D18,0024,0D80,FA79,4200,0DF8 ;27944        J/IE.19                ;  GOTO CLEAR IPR BIT
                                    ;27945
                                    ;27946          ;10------------------;  VECTOR HALT
                                    ;27947          PC&VA_RC[TO],          ;  PC GETS VECTOR ID
U 0339, 0010,0038,0180,F901,0200,0DF6 ;27948        J/IE.18                ;
                                    ;27949
                                    ;27950          ;11------------------;  VECTOR HALT
U 033B, 0010,0038,0180,F901,0200,0DF6 ;27951        PC&VA_RC[TO]           ;  PC GETS VECTOR ID
                                    ;27952  =;END
                                    ;27953
                                    ;27954  IE.18:  ;----------------------;
                                    ;27955          D_K[.7],               ;  HALT CODE IS 7 FOR VECTOR[1:0] = 3
U 0DF6, 0818,0038,5D80,F800,0000,039F ;27956        J/HALT.ERR             ;  GOTO ERROR HALT
                                    ;27957
                                    ;27958  IE.19:  ;----------------------;
                                    ;27959          START.IB,              ;  START IB FOR NXT INST
U 0DF8, 3019,0034,8D80,F800,0082,0DF9 ;27960        SC_D.AND.K[.1F]        ;  SC GETS IPRA
                                    ;27961
                                    ;27962          ;----------------------;
U 0DF9, 0801,2034,0180,F800,0000,0DFA ;27963        D_Q.AND.MASK           ;  CLEAR HIGHEST IPR (INTRPT PRIORITY REQ)
                                    ;27964
                                    ;27965          ;----------------------;
                                    ;27966          ID[SIR]_D,             ;  STORE SIR
                                    ;27967          MSC/CLR.NEST.ERR,      ;  CLR NESTED ERR FLAG
U 0DFA, 0000,003C,3980,3C00,2800,00AB ;27968        J/IB.FILL              ;  GOTO FILL IB
                                    ;27969
                                    ;27970  IE.20:  ;----------------------;
                                    ;27971          START.IB,              ;  START IB FOR NXT INST
                                    ;27972          MSC/CLR.NEST.ERR,      ;  CLR NESTED ERR FLAG
U 0DFB, 3000,003C,0180,F800,2800,00AB ;27973        J/IB.FILL              ;  GOTO FILL IB
```

```
                                    ;27974  INOVFL: ;----------------------;  INTEGER OVERFLOW
                                    ;27975          D_Q.OR.K[.10],Q_D,      ;  SET ARITH TRAP CODE TO #1
                                    ;27976          SET.V,                  ;  SET V BUT LEAVE OTHER CC'S ALONE
U 0DFC, 0819,2030,65E0,F800,0020,0E02 ;27977        J/CPUST                 ;  GOTO RESTORE UPDATED CPU ERROR/STATUS REG
                                    ;27978
                                    ;27979  INDIV0: ;----------------------;  INTEGER DIVIDE BY 0
                                    ;27980          D_Q.OR.K[.20],Q_D,      ;  SET ARITH TRAP CODE TO #2
                                    ;27981          SET.V,                  ;  SET V BUT LEAVE OTHER CC'S ALONE
U 0DFD, 0819,2030,75E0,F800,0020,0E02 ;27982        J/CPUST                 ;  GOTO RESTORE UPDATED CPU ERROR/STATUS REG
                                    ;27983
                                    ;27984  DIVBY0: ;----------------------;  FLOATING DIVIDE BY 0
                                    ;27985          Q_ID[CES],              ;  Q GETS CPU ERROR/STATUS REG
U 0E00, 0003,0028,31F0,2C00,0050,0E01 ;27986        ALU_-1,N&Z_ALU.V&C_0    ;  SET PSL <N,Z>
                                    ;27987
                                    ;27988          ;----------------------;
                                    ;27989          D_Q.OR.K[.40],Q_D,      ;  SET ARITH TRAP CODE TO #4
                                    ;27990          SET.V,                  ;  NOW PSL COND CODES <N,Z,V,C> ARE <1,0,1,0>
U 0E01, 0819,2030,31E0,F800,0020,0E02 ;27991        J/CPUST                 ;  GOTO RESTORE UPDATED CPU ERROR/STATUS REG
                                    ;27992
                                    ;27993  CPUST:  ;----------------------;
                                    ;27994          ID[CES]_D,D_Q,          ;  STORE THE UPDATED CPU ERROR/STATUS REG
                                    ;27995          MSC/INH.CM.ADDR,        ;  ALLOW VAX MODE ADDR EVEN PSL <CM> IS SET
U 0E02, 0C00,003E,3180,3C00,3C00,0001 ;27996        RETURN1                 ;  GO BACK TO UNFINISHED BUSINESS
                                    ;27997
                                    ;27998  OVFL:   ;----------------------;
                                    ;27999          Q_ID[CES],              ;  Q GETS CPU ERROR/STATUS REG
U 0E03, 0003,0028,31F0,2C00,0050,0E08 ;28000        ALU_-1,N&Z_ALU.V&C_0    ;  SET N,Z TO 1,0
                                    ;28001
                                    ;28002  ; ************************************************
                                    ;28003  ; * Patch no. 018, PCS 0E03 trapped to WCS 1155 *
                                    ;28004  ; ************************************************
                                    ;28005
                                    ;28006  IE.PA.18:
                                    ;28007          ;----------------------;
                                    ;28008          D_Q.OR.K[.30],Q_D,      ;  SET ARITH TRAP CODE TO #3
U 0E08, 0819,2030,79E0,F800,0020,0E0C ;28009        SET.V,J/CPUS            ;  NOW PSL COND CODES <N,Z,V,C> ARE <1,0,1,0>
                                    ;28010
                                    ;28011  UNDRFL: ;----------------------;
                                    ;28012          Q_ID[CES],              ;  Q GETS CPU ERROR/STATUS REG
U 0E09, 0003,003C,31F0,2C00,0050,0E0A ;28013        ALU_0(A),N&Z_ALU.V&C_0  ;  SET PSL <Z> FOR UNDERFLOW
                                    ;28014
                                    ;28015  ; ************************************************
                                    ;28016  ; * Patch no. 019. PCS 0E09 trapped to WCS 1156 *
                                    ;28017  ; ************************************************
                                    ;28018
                                    ;28019  IE.PA.18A:
                                    ;28020          ;------------------------------;
U 0E0A, 0819,2030,35E0,F800,0000,0E0C ;28021        D_Q.OR.K[.50],Q_D,J/CPUS        ; SET ARITH TRAP CODE TO #5
                                    ;28022
                                    ;28023  CLR.CES:;------------------------------;
U 0E0B, 0819,2024,CDE0,F800,0000,0E0C ;28024        D_Q.ANDNOT.K[.F0],Q_D,J/CPUS    ; CLEAR CES ARITH TRAP CODE
                                    ;28025
                                    ;28026  CPUS:   ;------------------------------;
                                    ;28027          ID[CES]_D,D_Q,                  ; STORE THE UPDATED CPU ERROR/STATUS REG
U 0E0C, 0C00,003E,3180,3C00,3C00,000F ;28028        MSC/INH.CM.ADDR,RETURNF         ; ALLOW VAX MODE ADD EVEN PSL<CM> IS SET
```

```
                                ;28029  .TOC    ''      Interrupt & Exception : ARITHT, TRACE, RSVOP1''
                                ;28030
                                ;28031  ;ARITHMETIC TRAPS -- EXCEPTIONS WITH PARAMETER (TRAP CODE)
                                ;28032
                                ;28033  03C:
                                ;28034  ARITHT: ;0----------------------;
                                ;28035          R[TO]_K[.34],           ;  STORE VECTOR ID FOR ARITH TRAPS
U 003C, 0018,0039,2980,F980,0000,0DE5   ;28036          CALL,J7EXCPTN           ;  CALL MAIN I+E RT, ENTER EXCEPTION WITH PARA
                                ;28037
                                ;28038  03D:    ;1----------------------;
U 003D, 0000,003C,31F0,2C00,0000,0E0E   ;28039          Q_ID[CES]               ;  GET CPU ERROR/STATUS REGISTER
                                ;28040  =;END
                                ;28041
                                ;28042          ;----------------------;
                                ;28043          R[SP]&VA_LA-K[.4],      ;  SP_SP-4
U 0E0E, 0018,0000,1190,FAF0,0200,0E10   ;28044          Q_Q.RIGHT2              ;  SHIFT STATUS REG BY 2
                                ;28045
                                ;28046          ;----------------------;
                                ;28047          ALU_Q.AND.K[.1E],       ;  MASK TO HAVE [04:02]
                                ;28048          D_ALU.RIGHT2,           ;  D[02:00] HAS ARITH TRAP CODE
U 0E10, 0899,2034,5180,F800,0000,002A   ;28049          J7EXCPT2                ;  GOTO PUSH DOWN LAST PARA AND CLR FPD BIT OF PSL
                                ;28050
                                ;28051
                                ;28052  ;TRACE TRAP FAULT -- EXCEPTION
                                ;28053
                                ;28054  024:
                                ;28055  TRACE:  ;----------------------;
                                ;28056          R[TO]_K[.28],           ;  VECTOR ID FOR TRACE TRAP, PASS TO MAIN I+E RT
U 0024, 0018,0038,2D80,F980,0000,08FC   ;28057          J/EXCPT                 ;  GOTO MAIN I+E RT, ENTER AT EXCEPTION W/O PARA
                                ;28058
                                ;28059  ;RESERVED OPERAND FAULT -- EXCEPTION
                                ;28060
                                ;28061  106:
                                ;28062  RSVOPR: ;----------------------;
                                ;28063          R[TO]_K[.18],           ;  VECTOR ID FOR RSV OPD TRAP PASSED TO MAIN I+E RT
U 0106, 0018,1138,7D80,F980,0000,0D07   ;28064          FPD?                    ;  SEE IF EXCEPTION WAS CAUSED FROM POLY
                                ;28065
                                ;28066  =0111
                                ;28067          ;0----------------------;  FPD SET - MUST BE RESERVED COEFFICIENT IN POLY
U 0D07, 0014,0038,01C0,F9E0,0000,0D0F   ;28068          Q_PC, R[PC.SV]_PC       ;  SAVE PC WHERE BAKUP.PC NEEDS IT
                                ;28069
                                ;28070  ; ************************************************
                                ;28071  ; * Patch no. 007, PCS 0D07 trapped to WCS 1146 *
                                ;28072  ; ************************************************
                                ;28073
                                ;28074          ;1----------------------;  FIRST PART DONE NOT SET - NOT POLY
U 0D0F, 0000,003C,0180,F800,0000,08FC   ;28075          J/EXCPT                 ;  PROCESS AS NORMAL EXCEPTION W/O PARAMS
                                ;28076  =;END
                                ;28077  =0*
                                ;28078  RSVOP1: ;0*----------------------;
U 0908, 0000,003D,0180,F800,0000,0EB8   ;28079          CALL, J/BAKUP.PC        ;  BACK UP THE PC BUT NOT THE RLOG REGISTERS
                                ;28080
                                ;28081          ;1*----------------------;
U 090A, 0000,003C,0180,F800,0000,08FE   ;28082          J/EXCPT0                ;  ENTER THE 'NO PARAMETERS'' CODE IN THE MIDDLE
                                ;28083  =;END
```

```
                                    ;28084  .TOC     ''      Interrupt & Exception : SNV, KSNV''
                                    ;28085
                                    ;28086  =11101
                                    ;28087  SNV:    ;0--------------------;  INTERRUPT STACK NOT VALID
                                    ;28088          D_K[.4],               ;  HALT CODE IS 4 FOR ISNV
U 015D, 0818,0038,1180,F800,0000,039F ;28089        J/HALT.ERR             ;
                                    ;28090
                                    ;28091          ;1--------------------;  KERNEL STACK NOT VALID
                                    ;28092          D_D.SWAP,              ;  D GETS 001F000C, READY TO RAISE IPL TO 1F
U 015F, 0B00,003C,3DF0,2C00,0000,0E11 ;28093        Q_ID[PSL]              ;  GET PSL
                                    ;28094  =;END
                                    ;28095
                                    ;28096  KSNV:   ;--------------------;  KERNEL STACK NOT VALID -- EXCEPTION
U 0E11, 081D,0030,0180,F800,0000,0E12 ;28097        D_D[OR]Q               ;  RAISE IPL TO 1F
                                    ;28098
                                    ;28099          ;--------------------;
U 0E12, 0018,0038,0180,F980,0000,0E13 ;28100        RC[TO]_K[.8]           ;  VECTOR ID FOR KSNV IS 08, PASS TO MAIN I+E RT
                                    ;28101
                                    ;28102          ;--------------------;
                                    ;28103          ID[PSL]_D,             ;  RESTORE UPDATED PSL
U 0E13, 0000,003C,3D80,3C00,0000,08FE ;28104        J/EXCPTO               ;  GOTO MAIN I+E RT, ENTER AT EXCEPTION W/O PARAS
                                    ;28105
                                    ;28106  .LIST           ;Re-enable full listing
                                    ;28107
                                    ;28108  ;THE FOLLOWING LOCATION IS THE USER WCS MICROCODE DISPATCH LOCATION. THE WCS
                                    ;28109  ;LOADER OVERWRITES THIS LOCATION WITH THE FOLLOWING JUMP TO THE USERS WCS
                                    ;28110  ;MICROCODE ENTRY POINT.
                                    ;28111  ;
                                    ;28112  ;10E0:  ;--------------------------------;
                                    ;28113  ;       ALU_R(SP1)+K[ZERO].RLOG,        ; This is to un-do the clobbering of
                                    ;28114  ;                                       ; RLOG that occured as a result of
                                    ;28115  ;                                       ; calling the BAKUP.RGS routine.
                                    ;28116  ;       J/MYCODE                        ; MYCODE is the user's entry point.
                                    ;28117
                                    ;28118  10E0:
                                    ;28119  USRTRP: ;--------------------------------; HERE FROM CODE=2 VECTOR
U 10E0, 0000,003E,0180,F800,0000,0001 ;28120        RETURN1                         ; PROCESS AS CODE 3
                                    ;28121
                                    ;28122  .NOLIST         ;Disable listing of PCS code for quickie assemblies
```

```
                                    ;28123  .TOC    ""      Interrupt & Exception : Compatibility mode faults''
                                    ;28124
                                    ;28125  ;-11 MODE FAULTS -- EXCEPTIONS WITH PARAMETERS
                                    ;28126
                                    ;28127  790:
                                    ;28128  CMRSV:  ;----------------------;  COMP MODE RESERVED INSTRUCTIONS: CODE IS 0
                                    ;28129          RC[T1] ALU,ALU_0(A),    ;  NOTE K[ZERO] IS FOR VAX MODE ONLY
U 0790, 0003,003C,0180,F988,0000,0910 ;28130        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28131
                                    ;28132  721:
                                    ;28133  CMBPT:  ;----------------------;  COMP MODE BPT INST TRAP
                                    ;28134          RC[T1] K[.1],           ;  TRAP CODE IS 1
U 0721, 0018,0038,0580,F988,0000,0910 ;28135        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28136
                                    ;28137  722:
                                    ;28138  CMIOT:  ;----------------------;  COMP MODE IOT INST TRAP
                                    ;28139          RC[T1] K[.2],           ;  TRAP CODE IS 2
U 0722, 0018,0038,0980,F988,0000,0910 ;28140        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28141
                                    ;28142  788:
                                    ;28143  CMEMT:  ;----------------------;  COMP MODE EMT INST TRAP
                                    ;28144          RC[T1] K[.3],           ;  TRAP CODE IS 3
U 0788, 0018,0038,0D80,F988,0000,0910 ;28145        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28146
                                    ;28147  798:
                                    ;28148  CMTRAP: ;----------------------;  COMP MODE TRAP INST TRAP
                                    ;28149          RC[T1] K[.4],           ;  TRAP CODE IS 4
U 0798, 0018,0038,1180,F988,0000,0910 ;28150        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28151
                                    ;28152  7DD:
                                    ;28153  CMIL:   ;----------------------;  COMP MODE ILLEGAL INST TRAP
                                    ;28154          RC[T1] K[.4]+1,         ;  TRAP CODE IS 5
U 07DD, 001B,0010,1180,F988,0000,0910 ;28155        J/CMFAULT               ;  GOTO COMP MODE TRAPS/ABORTS
                                    ;28156
                                    ;28157  10E:
                                    ;28158  CMODD:  ;----------------------;  COMP MODE ODD ADDRESS TRAP
U 010E, 0018,0038,0580,F988,0000,0910 ;28159        RC[T1] K[.6]            ;  TRAP CODE IS 6
```

```
                                    ;28160  ;HERE FOR COMPATABILITY MODE FAULTS, WITH CODE IN RC[T1]
                                    ;28161
                                    ;28162  =0*
                                    ;28163  CMFAULT::;0----------------------;
                                    ;28164          STATE_0(A),             ;  CLR STATE, TO GET READY TO SET 1
                                    ;28165          RC[PC.SV]_PC,           ;  PUT PC WHERE BAKUP.PC EXPECTS IT
                                    ;28166          CALL,RLOG.EMPTY?,       ;  BACKUP REG'S AND PC
U 0910, 0017,1539,0180,F9E0,1408,6DA7 ;28167        J/BAKUP.RGS             ;
                                    ;28168
                                    ;28169          ;1----------------------;
U 0912, 0018,0038,7980,F980,0000,08E0 ;28170        RC[T0]_K[.30]           ;  VECTOR ID FOR COMP MODE IS 30
                                    ;28171  =;END
                                    ;28172
                                    ;28173  =0
                                    ;28174  IE.PA.94:
                                    ;28175          ;0----------------------;
                                    ;28176          Q_ID[PSL],              ;  Q GETS PSL
                                    ;28177          STATE_STATE+1,          ;  SET STATE REG TO 1 FOR PARAMETERS
U 08E0, 0000,003D,3DF0,2C00,1400,CDE8 ;28178        CALL,J/EXCPT1           ;  CALL MAIN I+E RT, ENTER AT EXCEPTION WITH PARAS
                                    ;28179
                                    ;28180          ;1----------------------;
U 08E1, 0018,0000,1180,FAF0,0200,0E14 ;28181        R[SP]&VA_LA-K[.4]       ;  SP_SP-4
                                    ;28182  =;END
                                    ;28183
                                    ;28184          ;----------------------;
                                    ;28185          D_RC[T1],               ;  D GETS TRAP CODE
U 0E14, 0810,0038,0180,F908,0000,002A ;28186        J/EXCPT2                ;  GOTO PUSH THE PARAMETER
```

```
                                    ;28187  .TOC    ''       Interrupt & Exception : SETFPD''
                                    ;28188
                                    ;28189  ;SETFPD ROUTINE TO SETUP FIRST PART DONE
                                    ;28190  ; PRESERVES D, DESTROYS Q
                                    ;28191
                                    ;28192          ;-----------------------;
U 0E16, 0000,003C,81F0,2C00,0000,0E18   ;28193  SETFPD: Q_ID[USTACK]             ;GET RETURN ADDRESS
                                    ;28194
                                    ;28195          ;-----------------------;
                                    ;28196          D_Q,                    ;READY TO WRITE RETURN ADDR BACK
                                    ;28197          Q_D,                    ;SAVE D TEMPORARILY
U 0E18, 0C00,003C,01E0,F800,2400,0E19   ;28198          SET.FPD                 ;SET THE FLAG
                                    ;28199
                                    ;28200          ;-----------------------;
U 0E19, 0000,003C,8180,3C00,0000,0E1C   ;28201          ID[USTACK]_D            ;PUT BACK FOR THIS RETURN
                                    ;28202
                                    ;28203          ;-----------------------;
                                    ;28204          ID[FPDA]_D,             ;ALSO SAVE FOR FAULT TIME
                                    ;28205          D_Q,                    ;RESTORE D
U 0E1C, 0C00,003E,B580,3C00,0000,0003   ;28206          RETURN3
```

```
                                    ;28207  .TOC    ''      Change modes          : CHMK, CHME, CHMS, CHMU''
                                    ;28208
                                    ;28209  ;OPCODES       BC      CHMK    CHANGE MODE TO KERNEL
                                    ;28210  ;              BD      CHME    CHANGE MODE TO EXECUTIVE
                                    ;28211  ;              BE      CHMS    CHANGE MODE TO SUPERVISOR
                                    ;28212  ;              BF      CHMU    CHANGE MODE TO USER
                                    ;28213
                                    ;28214  288:
                                    ;28215  CHM:    ;-----------------------;
                                    ;28216          Q_ID[PSL],              ;  GET PSL
                                    ;28217          R[T1]_D.SXT[WORD],      ;  SAVE CODE SIGN EXTENDED
U 0288, 0002,493C,3DF0,2D88,0000,0B2A ;28218        IR1?                    ;  CK IR[01] BIT
                                    ;28219
                                    ;28220  =10     ;0---------------------;  CHMK, CHME
                                    ;28221          SC_K[ZERO],FE_K[ZERO],  ;  SET SC, FE TO 0
                                    ;28222          R[R15]_Q,D_Q,           ;  R[R15], D GET PSL
U 0B2A, 0C01,3B3C,1980,FAF8,0184,6CBD ;28223        IR0?,J7CHM.1            ;  CHMK, CHME, CHMS OR CHMU
                                    ;28224
                                    ;28225          ;1---------------------;  CHMS, CHMU
                                    ;28226          SC_K[.2],FE_K[.2],      ;  SET SC, FE TO 2
                                    ;28227          R[R15]_Q,D_Q,           ;  R[R15], D GET PSL
U 0B2B, 0C01,3B3C,0980,FAF8,0184,6CBD ;28228        IR0?                    ;  CHMK, CHME, CHMS OR CHMU
                                    ;28229  =;END
                                    ;28230
                                    ;28231  =1101
                                    ;28232  CHM.1:  ;0---------------------;
                                    ;28233          D_D.SWAP,               ;  BYTE SWAP PSL
U 0CBD, 0E00,003C,0180,F800,0000,0E1D ;28234        J7CHM.2                 ;  GOTO GET CURRENT MODE
                                    ;28235
                                    ;28236          ;1---------------------;
                                    ;28237          FE_SC+1,                ;  ADD 1 TO FE
U 0CBF, 0B00,003C,0180,F800,0100,CE1D ;28238        D_D.SWAP                ;  BYTE SWAP PSL
                                    ;28239  =;END
                                    ;28240
                                    ;28241  CHM.2:  ;-----------------------;
                                    ;28242          SC_D.AND.K[.3],         ;  SC GETS CURRENT MODE[01:00]
                                    ;28243          STATE_FE,               ;  SAVE CHM* ID
U 0E1D, 0019,1C34,0D80,F800,1482,61FD ;28244        PSL.MODE?               ;  IS PSL<IS> BIT SET?
                                    ;28245
                                    ;28246  =11101  ;0---------------------;
                                    ;28247          D_K[.A],                ;  HALT CODE TO CONSOLE IS #A
U 01FD, 0818,0038,F580,F800,0000,039F ;28248        J7HALT.ERR              ;  <IS> IS 1: HALT
                                    ;28249
                                    ;28250          ;1---------------------;
                                    ;28251          EALU_SC-FE,             ;  CURRENT MODE - NEW MODE
                                    ;28252          CLK.DBCC,               ;  LATCH EALU COND CODES
                                    ;28253          R[T0]_K[SC],            ;  SAVE CURRENT MODE
U 01FF, 0018,0038,1D81,F980,0010,AE20 ;28254        SS_ALUT5                ;  CLR SS TO MINIMIZE EALU.N CONSTRAINED BLOCK
                                    ;28255  =;END
                                    ;28256
                                    ;28257          ;-----------------------;
                                    ;28258          SC_SC+K[.28],           ;  SC GETS KSP ID BUS ADDRESS AS BASE ADDRESS
                                    ;28259          D_R[SP],Q_R[SP],        ;  D, Q GET SP
U 0E20, 0800,123C,2DC0,FA70,0084,8696 ;28260        EALU.N?                 ;  WHICH MODE IS LOWER?
```

```
                                    ;28261  =011*   ;0----------------------;
                                    ;28262          SC_FE,                  ;  SC GETS NEW MODE
                                    ;28263          ID[SC]_D,               ;  UPDATE CURRENT SP ON ID BUS
U 0696, 0000,003C,0180,3400,0081,0E21 ;28264        J/CHM.4                 ;  GOTO GET MIN(NEW, CURRENT MODES)
                                    ;28265
                                    ;28266  CHM.5:  ;1----------------------;
                                    ;28267          RC[T4]_Q-K[.4],         ;  SP_SP-4
                                    ;28268          FE_K[.4],               ;  SETUP TO RETURN SP TO VA AFTER PRB.W RTN
U 069E, 0019,2000,1180,F9A0,0104,6E22 ;28269        J/CHM.6                 ;  GOTO PUSH PSL INTO STACK
                                    ;28270  =;END
                                    ;28271
                                    ;28272  CHM.4:  ;----------------------;
                                    ;28273          RC[T0]_K[SC],           ;  GET NEW MODE (MIN OF NEW & CURRENT MODES)
U 0E21, 0018,1738,1D80,F980,0000,0B30 ;28274        STATE1=0?               ;  BENCH ON CURRRENT MODE TO UPDATE SP'S
                                    ;28275
                                    ;28276  =00
                                    ;28277          ;00--------------------;
                                    ;28278          Q_ID[KSP],              ;  GET RIGHT SP
U 0B30, 0000,003C,A1F0,2C00,0000,069E ;28279        J7CHM.5                 ;  GOTO PUSH OLD PSL
                                    ;28280
                                    ;28281          ;01--------------------;
                                    ;28282          Q_ID[ESP],              ;  GET RIGHT SP
U 0B31, 0000,003C,A5F0,2C00,0000,069E ;28283        J7CHM.5                 ;  GOTO PUSH OLD PSL
                                    ;28284
                                    ;28285          ;10--------------------;
                                    ;28286          Q_ID[SSP],              ;  GET RIGHT SP
U 0B32, 0000,003C,A9F0,2C00,0000,069E ;28287        J7CHM.5                 ;  GOTO PUSH OLD PSL
                                    ;28288
                                    ;28289          ;11--------------------;
                                    ;28290          Q_ID[USP],              ;  GET RIGHT SP
U 0B33, 0000,003C,ADF0,2C00,0000,069E ;28291        J7CHM.5                 ;  GOTO PUSH OLD PSL
                                    ;28292  =;END
                                    ;28293
                                    ;28294  CHM.6:  ;----------------------;
                                    ;28295          D_R[SP],                ;  GET SP TO SAVE
U 0E22, 0800,003C,7D80,FA70,0084,6E28 ;28296        SC_K[.18]               ;  SC GETS 24. FOR SHIFTING CURRENT MODE
                                    ;28297
                                    ;28298          ;----------------------;
                                    ;28299          ID[T0]_D,               ;  SAVE SP
U 0E28, 0810,0038,C180,3D00,0000,0E29 ;28300        D_RC[T0]                ;  D GETS MIN(NEW MODE, CURRENT MODE)
                                    ;28301
                                    ;28302          ;----------------------;
                                    ;28303          EXCEPT.ACK,             ;  RESET PSL
                                    ;28304          D_D+K[.8],              ;  D GETS MIN(NEW MODE, CURRENT MODE) & FPD
U 0E29, 0819,0014,01F8,F800,C000,0E2C ;28305        Q_0                     ;  CLR Q FOR DAL
                                    ;28306
                                    ;28307          ;----------------------;
                                    ;28308          D_DAL.SC,               ;  D GETS MODE IN [25:24]
U 0E2C, 0D00,003C,3DF0,2C00,0000,0B40 ;28309        Q_ID[PSL]               ;  Q GETS NEWLY CREATED PSL
```

```
                                    ;28310  =00     ;00--------------------;
                                    ;28311          D_D[OR]Q,              ;  SET CURRENT MODE OF NEW PSL
U 0B40, 081D,0031,0180,F800,0000,0E16 ;28312        CALL, J/SETFPD         ;  CALL SET UP FPD
                                    ;28313
                                    ;28314          ;01--------------------;  WRITE ERR: SERVICE IT
                                    ;28315          D_R[R15],              ;  GET BACK OLD PSL WHICH HAS NOT FPD SET
                                    ;28316          Q_ID[T0],              ;  GET BACK THE SAVED R[SP]
U 0B41, 0800,003C,C1F0,2E78,0000,0E40 ;28317        J7CHM.8                ;  GOTO RESET PSL TO OLD VALUE
                                    ;28318
                                    ;28319          ;10--------------------;  READ ERR: SERVICE IT
                                    ;28320          D_R[R15],              ;  GET BACK OLD PSL WHICH HAS NOT FPD SET
                                    ;28321          Q_ID[T0],              ;  GET BACK THE SAVED R[SP]
U 0B42, 0800,003C,C1F0,2E78,0000,0E40 ;28322        J7CHM.8                ;  GOTO RESET PSL TO OLD VALUE
                                    ;28323
                                    ;28324          ;11--------------------;
U 0B43, 0010,0038,01C0,F920,0000,0E2D ;28325        Q_RC[T4]               ;  GET BACK SP
                                    ;28326  =;END
                                    ;28327
                                    ;28328          ;----------------------;
                                    ;28329          VA_Q-K[.8],Q_ALU,      ;  PROBE NEXT 2 LONG WORDS TO BE PUSHED DOWN
U 0E2D, 0019,2000,01C0,F800,0281,08E4 ;28330        SC_FE                  ;  SC, FE HAVE #4, FOR QUAD PROBE AND RETURN RC 4
                                    ;28331
                                    ;28332  =0      ;0---------------------;
                                    ;28333          ID[PSL]_D,             ;  RESTORE NEW PSL
                                    ;28334          D_Q.RIGHT,SI/ZERO,     ;  D HAS ADDR/2 FOR PRB.W
U 08E4, 0841,203D,3D80,3C00,0000,0D0D ;28335        CALL,J/PRB.W           ;  PROBE NEXT QUAD
                                    ;28336
                                    ;28337          ;1---------------------;  RETURN: PROBE OK
                                    ;28338          Q_D,                   ;  GET BACK THE SAVED SP (IN RC 4)
                                    ;28339          D_R[R15],              ;  GET BACK OLD PSL
U 08E5, 0800,003C,01E0,FA78,1480,0E30 ;28340        SC_STATE               ;  SC GETS MODE OF CHM*
                                    ;28341  =;END
                                    ;28342          ;----------------------;
                                    ;28343          CACHE_D[LONG],         ;  PUSH PSL INTO STACK
                                    ;28344          SC_SC+K[.10],          ;  SETUP FOR CHM* ID * 4
U 0E30, 0000,003C,6580,3000,0184,8E31 ;28345        FE_EALU
                                    ;28346
                                    ;28347          ;----------------------;
                                    ;28348          R[SP]&VA_Q-K[.4],      ;  SP_SP-4
                                    ;28349          SC_SC+FE,              ;  GET CHM* VECTOR / 2
U 0E31, 0019,2000,1180,FAF0,0380,8E32 ;28350        FE_SC+FE               ;
                                    ;28351
                                    ;28352          ;----------------------;
                                    ;28353          D_PC,                  ;  D GETS PC TO BE PUSHED INTO STACK
                                    ;28354          SC_SC+FE,              ;  SC HAS CHM* VECTOR ID NOW
U 0E32, 0814,0038,0180,FA70,0080,8E33 ;28355        LAB_R[SP]              ;  LATCH SP IN LA, LB
                                    ;28356
                                    ;28357          ;----------------------;
                                    ;28358          CACHE_D[LONG],         ;  PUSH DOWN PC INTO STACK
U 0E33, 0010,0038,01C0,3108,0000,0E34 ;28359        Q_RC[T1]               ;  Q GETS CODE SIGN EXTENDED
                                    ;28360
                                    ;28361          ;----------------------;
                                    ;28362          D_Q,                   ;  D GETS SXT CODE
U 0E34, 0C18,0000,1180,FAF0,0200,0E35 ;28363        R[SP]&VA_LA-K[.4]      ;  SP_SP-4
```

```
                                    ;28364          ;----------------------;
U 0E35, 0000,003C,0180,3000,0000,0E38   ;28365          CACHE_D[LONG]          ;  PUSH DOWN THE SXT CODE INTO STACK
                                    ;28366
                                    ;28367          ;----------------------;
U 0E38, 0000,003C,EDF0,2C00,0000,0E39   ;28368          Q_ID[SCBB]             ;  GET S/W CONTROL BLOCK BASE REG
                                    ;28369
                                    ;28370          ;----------------------;
                                    ;28371          CLR.FPD,               ;  CLEAR FPD OF PSL
U 0E39, 0019,2014,1D80,F800,2200,0E3C   ;28372          VA_Q+K[SC]             ;  GET NEW PC
                                    ;28373
                                    ;28374          ;----------------------;
U 0E3C, 0000,003C,0180,C800,0000,0E3D   ;28375          D[LONG]_CACHE.P        ;  GET VECTOR
                                    ;28376
                                    ;28377          ;----------------------;
                                    ;28378          ALU_D.ANDNOT.K[.3],    ;  SET NEW PC
                                    ;28379          PC&VA_ALU,             ;
                                    ;28380          FLUSH.IB,              ;  FLUSH IB, LOAD PC AND VA, ACCEPT INTERRUPTS
U 0E3D, 2019,0024,0D80,F801,4200,00AB   ;28381          J/IB.FILL              ;  GOTO FILL IB, THEN NXT INST
                                    ;28382
                                    ;28383  CHM.8:  ;----------------------;
                                    ;28384          ID[PSL]_D,             ;  MEM FAILS, STORE OLD PSL
U 0E40, 0001,203C,3D80,3EF0,0000,0E41   ;28385          R[SP]_Q                ;  RESET SP TO OLD VALUE
                                    ;28386
                                    ;28387          ;----------------------;
                                    ;28388          D_RC[PTE.VA],          ;  GET BACK SAVED PARAMETERS FOR ERR
U 0E41, 0810,0038,0180,F950,0000,0EA9   ;28389          J/M.FLT.PRB            ;  GOTO MEM MANAGEMENT ERR HANDLING
                                    ;28390
                                    ;28391  .LIST           ;Re-enable full listing
```

```
                                        ;28392  .TOC    'MMFW.MIC''
                                        ;28393  .TOC    'Revision 1.1''
                                        ;28394  ;        P. R. Guilbault
                                        ;28395
;28396  .NOBIN
;28397  .TOC    ''      Revision History''
;28398
;28399  ; 01    Remove absolute jumps.
;28400  ; 00    Created from MMCOM.MIC, MMGP.MIC, PGBRY2.MIC, MMFLT2.MIC, PROBEV.MIC, MBIT2.MIC, GTPTE2.MIC, and MMFLT.MIC
;28401  ;       Start of history
;28402
                                        ;28403  .BIN
                                        ;28404  .NOLIST         ;Disable listing of PCS code for quickie assemblies
```

```
;28405  .TOC    ''      Memory management     : MEMORY MANAGEMENT FIRMWARE DESCRIPTION''
;28406
;28407          ;************************************************
;28408          ;                                               *
;28409          ;       STAR MEMORY MANAGEMENT FIRMWARE (MMFW)  *
;28410          ;                                               *
;28411          ;************..***********************************
;28412
;28413  ;       MEMORY MANAGEMENT MICRO-TRAPS
;28414  ;       ----------------------------
;28415  ;       THE MEMORY MANAGEMENT MICRO-CODE IS INITIATED BY ANY ONE OF THE
;28416  ;       MEMORY MANAGEMENT MICRO-TRAPS.  THESE MICRO-TRAPS OCCUR ONLY ON
;28417  ;       VIRTUAL MEMORY REFERENCES AND MME=1(MME=0/1 IF UNALIGNED DATA TRAP).
;28418  ;       THE ROUTINES USED TO HANDLE THE MICRO-TRAPS ARE:
;28419  ;
;28420  ;               1)  TRANSLATION BUFFER MISS MICRO-TRAP ROUTINE, ''TBUF.MISS''
;28421  ;                   ROUTINE TO FETCH THE PTE REQUIRED AND PUT IT INTO
;28422  ;                   THE TRANSLATION BUFFER.
;28423  ;
;28424  ;               2)  SET MBIT MICRO-TRAP ROUTINE, 'SET.MBIT''
;28425  ;                   SETS PTE<M> BIT FOR THE PAGE BEING REFERENCED
;28426  ;
;28427  ;               3)  PROTECTION MICRO-TRAP, 'PROT.CHK''
;28428  ;                   DOES SETUP NECESSARY TO TRANSFER CONTROL TO THE FAULT
;28429  ;                   HANDLING ROUTINE.
;28430  ;
;28431  ;               4)  PAGE BNDRY MICRO-TRAP ROUTINE, 'PG.BDRY''
;28432  ;                   PROBES THE 2ND PAGE NEEDED TO COMPLETE THE TRAPPED
;28433  ;                   REFERENCE TO DETERMINE IF IT CAN BE DONE.
;28434  ;
;28435  ;               5)  UNALIGNED DATA MICRO-TRAP ROUTINE, 'M.UD.BGN''
;28436  ;                   DOES THE 2ND REFERENCE NEEDED TO COMPLETE A
;28437  ;                   TRAPPED REFERENCE THAT IS NOT ALIGNED WITHIN LONG WORD
;28438
;28439  ;       THE ROUTINE INITIATED BY THE MICRO-TRAP ATTEMPTS TO COMPLETE THE TRAPPED
;28440  ;       REFERENCE.  IF IT IS SUCCESSFUL, IT RESTORES THE INTERENAL
;28441  ;       REGISTERS ,THOSE NOT RESERVED FOR MEMORY MANAGEMENT, THAT IT
;28442  ;       USED AND RETURNS CONTROL TO THE MICRO-INSTRUCTION THAT WAS BEING
;28443  ;       ADDRESSED BY THE TRAPPED MICRO-INSTRUCTION.  IF IT CANNOT COMPLETE
;28444  ;       THE TRAPPED REFERENCE BECAUSE SOFTWARE INTERVENTION IS REQUIRED,
;28445  ;       IT WILL INITIATE A MEMORY MANAGEMENT FAULT BY TRANSFERING CONTROL
;28446  ;       TO MEMORY MANAGEMENT FAULT HANDLING ROUTINE.
;28447
;28448  ;       I-BUFFER TBUF-MISS AND PROTECTION VIOLATION
;28449  ;       -------------------------------------------
;28450  ;       ENTRY, ''IB.TBM'', INTO MMFW IS USED TO FETCH THE PTE NEEDED
;28451  ;       BY THE I-BUFFER TO CONTINUE FETCHING INSTRUCTION STREAM AND LOAD IT
;28452  ;       INTO THE INSTRUCTION BUFFER.
;28453
;28454  ;       ENTRY, 'M.FLT.25'', INTO MMFW IS SET-UP AS A FPD RETURN ADDRESS IN CASE
;28455  ;       PSL<FPD>=1.  SINCE MEMORY MANAGEMENT EXCEPTIONS ALWAYS RESULT IN
;28456  ;       RESTARTING THE INSTRUCTION ON I-STREAM REFERENCES, THE FACT THAT
;28457  ;       PSL<FPD>=1 IS IGNORED ON I-BUFFER MISSES.  THIS ENTRY IS  USED TO SET-
;28458  ;       UP AND MERGE INTO THE NORMAL FAULT HANDLING ROUTINE FLOWS FOR FPD=0.
```

```
;28459  ;       MEMORY MANAGEMENT FAULTS
;28460  ;---------------------------------------
;28461  ;       IF A TRAPPED REFERENCE CANNOT BE SUCCESSFULLY COMPLETED AND
;28462  ;       REQUIRES SOFTWARE INTERVENTION, THE 'MEMORY MANAGEMENT FAULT
;28463  ;       HANDLING" ROUTINE WILL, IN CONCERT WITH THE I&E MICRO-CODE,
;28464  ;       SETUP AND INITIATE A MEMORY MANAGEMENT FAULT.  I-BUFFER FAULTS
;28465  ;       AND PROBE INSTR. FAULTS ARE ALSO HANDLED BY THIS ROUTINE.
;28466
;28467  ;       THIS ROUTINE WILL SET-UP THE NECESSARY PARAMETERS FOR THE FAULT,
;28468  ;       BACK-UP THE PC, RESET PSL(TP),  AND BACK-UP THE  GENERAL REGISTERS
;28469  ;        (IF REQUIRED) AS  REQUIRED TO HANDLE A MEMORY MANAGEMENT FAULT.
;28470
;28471
;28472  ;       MEMORY MANAGEMENT FAULTS ON STRING INSTRUCTIONS
;28473  ;-------------------------------------------------------
;28474  ;       ON STRING INSTRUCTIONS, IF THE PSL(FPD) BIT IS SET, THE MEMORY
;28475  ;       MANAGEMENT FAULT HANDLING ROUTINE WILL TRANSFER CONTROL
;28476  ;       TO MICRO-INSTRUCTION SPECIFIED BY THE ADDRESS IN ID[FPDA] OR'ED WITH
;28477  ;       1/2.  BEFORE DOING THIS, THE ROUTINE BACKS-UP THE PC AND RESTORES
;28478  ;       THE INTERNAL REGISTERS, THOSE NOT RESERVED FOR MMFW, TO THEIR CONTENTS
;28479  ;       ON THE MICRO-TRAP.
;28480
;28481  ;       NOTE:  THE MICRO-STACK HAS UNPREDICTABLE CONTENTS AFTER ARRIVING AT THE
;28482  ;       THE ADDRESS SPECIFIED ABOVE.
;28483
;28484  ;       GENERAL COMMENTS
;28485  ;-----------------------------------------------------------------
;28486  ;       FOR ADDITIONAL INFORMATION, GO TO ABOVE SPECIFIED MICRO-ROUTINES
```

```
                                ;28487  .TOC    ""      Memory management     : TBUFF MISS MICRO-TRAP ROUTINE''
                                ;28488
                                ;28489          ;*********************************************
                                ;28490          ;                                           *
                                ;28491          ; TRANSLATION BUFFER MISS MICRO-TRAP ROUTINE*
                                ;28492          ;                                           *
                                ;28493          ;*********************************************
                                ;28494
                                ;28495          ;THIS ROUTINE IS INITIATED BY A TBUF-MISS MICRO-TRAP.  IT
                                ;28496          ;FETCHES THE REQUIRED PTE, PERFORMS THE NECESSARY OPERATIONS
                                ;28497          ;AND CHECKS ON THE PTE, AND LOADS IT INTO THE TRANSLATION
                                ;28498          ;BUFFER.  IT THEN RETRIES THE TRAPPED MEMORY REFERENCE AND
                                ;28499          ;RETURNS CONTROL TO THE TRAPPED MICRO-FLOW.
                                ;28500
                                ;28501          ;PROCESSOR REGISTERS USED BY THIS ROUTINE ARE SAVED BY IT, AND
                                ;28502          ;RESTORED TO THEIR ENTRY VALUES BEFORE EXITING THIS ROUTINE.
                                ;28503          ;THIS ROUTINE CAN BE FAULTED BY A FETCHED PTE NOT BEING VALID, AND BY
                                ;28504          ;ACCESS VIOLATIONS.  SEE VAX SRM CHAPTER #5 FOR MORE DETAIL.
                                ;28505
                                ;28506          ;IF A MEMORY MANAGEMENT FAULT IS ENCOUNTERED, CONTROL IS TRANSFERED
                                ;28507          ;TO A FPD SETUP ROUTINE IF PSL(FPD) BIT IS SET.  THE PROCESSOR
                                ;28508          ;REGISTERS USED BY THIS ROUTINE (D,Q,SC,LC,VA, & PC) ARE RESTORED
                                ;28509          ;TO THEIR ENTRY VALUES.   THE USTACK'S CONTENTS ARE NOT
                                ;28510          ;PREDICTABLE
                                ;28511
                                ;28512  105:
                                ;28513  TBUF.MISS:
                                ;28514          ;----------------------;
                                ;28515          STOP.IB,               ;  STOP IBUF MEMORY REFERENCES
                                ;28516          ID[D.SV]_D,            ;  SAVE 'D'' REGISTER IN ID TEMPORARY
                                ;28517          RC[PC.SV]_PC,          ;  SAVE 'PC'' IN RC REGISTER SET
U 0105, 1C14,0038,B980,3DE0,0000,0E42   ;28518          D_Q                    ;  MOVE 'Q'' TO 'D'' REGISTER
                                ;28519
                                ;28520          ;----------------------;
                                ;28521          ID[Q.SV]_D,            ;  SAVE ENTRY CONTENTS OF Q IN ID TEMPORARY
U 0E42, 0010,0038,BD80,3DC0,0000,0B48   ;28522          RC[LC.SV]_LC           ;  SAVE LC ENTRY CONTENTS
                                ;28523
                                ;28524  =00                            ; CONSTRAINT BLOCK FOR CALL AND RETURN SEQUENCES
                                ;28525          ;00--------------------;
                                ;28526          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28527          SET.NEST.ERR,          ;  FLAG THAT PC CONTENTS NOW IN RC <PC.SV>
                                ;28528          PC_VA,                 ;  LOAD PC WITH TRAPPED VIRTUAL ADDRESS
                                ;28529          RC[SC.SV]_K[SC],       ;  SAVE ENTRY CONTENTS OF SC IN RC REG SET
U 0B48, 0018,1C39,1D80,09E9,2C00,023C   ;28530          CALL,PSL.MODE?,J/GET.PTE ;  GO GE'T PTE FOR TRAPPED VIRTUAL ADDRESS
                                ;28531
                                ;28532          ;01--------------------;  *** RETURN FROM GET.PTE ***
                                ;28533          RC[VA.SV]_PC,          ;  SAVE ENTRY CONTENTS OF VA IN RC REG SET
                                ;28534          TEST.TB.WCHK,          ;  SET-UP TEST FOR CHECKING PTE<M>
                                ;28535          CALL,                  ;
U 0B49, 0014,1139,0180,11C8,0000,0D6C   ;28536          LAST.REF?,J/TEST.PTE.M ;  GO HANDLE PTE <M> BIT
```

```
                                    ;28537          ;10----------------------;  *** MEM MANAGEMENT FAULT RETURN ***
                                    ;28538          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28539          RC[VA.SV]_PC,           ;  SAVE ENTRY VA CONTENTS
U 0B4A, 0014,1138,0180,09C8,0000,0D7C ;28540        LAST.REF?, J/M.FLT      ;
                                    ;28541
                                    ;28542          ;11----------------------;  *** RETURN FROM TEST.PTE[M] ***
                                    ;28543          Q_ID[D.SV],             ;  GET ENTRY CONTENTS FOR D REG
                                    ;28544          PC&VA_RC[PC.SV],        ;  RESTORE PC TO ENTRY CONTENTS
U 0B4B, 0010,0038,B9F0,2D61,2A00,0E48 ;28545        CLR.NEST.ERR            ;  RESET FLG INDICATING PC IN RC<PC.SV>
                                    ;28546  =;END
                                    ;28547
                                    ;28548  M.TBM.A: ;----------------------;  ***  SET.MBIT RTN ENTRY ***
                                    ;28549          VA_RC[VA.SV],           ;  RESTORE ENTRY CONTENTS TO VA REG
                                    ;28550          D_Q,                    ;  RESTORE ENTRY CONTENTS TO D REG
                                    ;28551          Q_ID[Q.SV],             ;  RESTORE ENTRY CONTENTS TO Q REG
U 0E48, 0C10,1138,BDF0,2D48,0200,0D1C ;28552        LAST.REF?,J/M.RETRY     ;  GO RETRY MEMORY REFERENCE TRAPPED
                                    ;28553
                                    ;28554
                                    ;28555  =1100   ;BRANCH ON SAVED CONTEXT OF TRAPPED MEMORY REFERENCE(BEN/LAST.REF)
                                    ;28556
                                    ;28557  M.RETRY:
                                    ;28558          ;1100----------------------;  *** READ MEMORY REFERENCE/READ CHECK ***
                                    ;28559          START.IB,               ;  RESTART IBUF
                                    ;28560          D_CACHE[RETRY.TRAP],    ;  RETRY TRAPPED REFERENCE PER SAVED CONTEXT
                                    ;28561          LC_RC[LC.SV],           ;  RESTORE ENTRY CONTENTS TO LC REG
U 0D1C, 3000,003E,0180,4140,3800,0000 ;28562        RETURNO                 ;  RETURN TO TRAPPED MICRO-FLOW
                                    ;28563
                                    ;28564          ;1101----------------------;  *** INTERLOCK READ ***
                                    ;28565          START.IB,               ;  RESTART IBUF
                                    ;28566          D_CACHE.LK[RETRY.TRAP], ;  RETRY REFERENCE PER SAVED CONTEXT
                                    ;28567          LC_RC[LC.SV],           ;  RESTROE ENTRY CONTENTS TO LC REG
U 0D1D, 3000,003E,0180,7140,3800,0000 ;28568        RETURNO                 ;  RETURN TO TRAPPED MICRO-FLOW
                                    ;28569
                                    ;28570          ;1110----------------------;  READ MEMORY REFERENCE/WRITE CHECK ***
                                    ;28571          START.IB,               ;  RESTART IBUF
                                    ;28572          D_CACHE.WCHK[RETRY.TRAP],;  RETRY TRAPPED REFERENCE PER SAVED CONTEXT
                                    ;28573          LC_RC[LC.SV],           ;  RESTORE ENTRY CONTENTS TO LC
U 0D1E, 3000,003E,0180,5140,3800,0000 ;28574        RETURNO                 ;  RETURN TO TRAPPED MICRO-FLOW
                                    ;28575
                                    ;28576          ;1111----------------------;  *** WRITE MEMORY REFERENCE ***
                                    ;28577          START.IB,               ;  RESTART IBUF
                                    ;28578          CACHE[RETRY.TRAP]_D,    ;  RETRY TRAPPED REFERENCE PER SAVED CONTEXT
                                    ;28579          LC_RC[LC.SV],           ;  RESTORE ENTRY CONTENTS TO LC REG
U 0D1F, 3000,003E,0180,3140,3800,0000 ;28580        RETURNO                 ;  RETURN TO TRAPPED MICRO-FLOW
```

```
                                ;28581  .TOC    ""      Memory management     : SET MBIT MICRO-TRAP ROUTINE''
                                ;28582
                                ;28583          ;*****************************
                                ;28584          ;                            *
                                ;28585          ;SET MBIT MICRO-TRAP ROUTINE*
                                ;28586          ;                            *
                                ;28587          ;*****************************
                                ;28588
                                ;28589          ;THIS ROUTINE IS INITIATED BY A MEMORY REFERENCE CAUSING THE
                                ;28590          ;FIRST MODIFY TO A PAGE.  IT FETCHES THE REQUIRED PTE, SETS THE PTE <M>
                                ;28591          ;BIT IN THE PAGE TABLE ENTRY REFERENCED BY THE VIRTUAL ADDRESS IN VA AND
                                ;28592          ;LOADS IT INTO THE TRANSLATION BUFFER.  IT THEN
                                ;28593          ;RETRIES THE TRAPPED MEMORY REFERENCE AND
                                ;28594          ;RETURNS CONTROL TO THE TRAPPED MICRO-FLOW.
                                ;28595
                                ;28596          ;PROCESSOR REGISTERS USED BY THIS ROUTINE ARE SAVED BY IT, AND
                                ;28597          ;RESTORE TO THEIR ENTRY VALUES BEFORE EXITING THIS ROUTINE.
                                ;28598          ;SEE VAX SRM CHAPTER #5 FOR MORE DETAIL.
                                ;28599
                                ;28600  103:
                                ;28601  SET.MBIT:
                                ;28602          ;----------------------;
                                ;28603          STOP.IB,                ;  STOP IBUF FROM MAKING MEMORY REQUESTS
                                ;28604          ID[D.SV]_D,             ;  SAVE D REG IN ID TEMP REG
                                ;28605          RC[PC.SV]_PC,           ;  SAVE PC IN RC REG. SET
U 0103, 1C14,0038,B980,3DE0,0000,0E49   ;28606          D_Q                     ;  MOVE Q TO D REG.
                                ;28607
                                ;28608          ;----------------------;
                                ;28609          ID[Q.SV]_D,             ;  SAVE ENTRY CONTENTS OF Q IN ID TEMPORARY
U 0E49, 0010,0038,BD80,3DC0,0000,0B50   ;28610          RC[LC.SV]_LC            ;  SAVE LC IN RC REG SET
                                ;28611
                                ;28612
                                ;28613  =00                             ; CONSTRAINT BLOCK FOR CALL AND RETURN SEQUENCES
                                ;28614          ;00----------------------;
                                ;28615          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28616          SET.NEST.ERR,           ;  FLAG THAT PC CONTENTS NOW IN RC<PC.SV>
                                ;28617          PC_VA,                  ;  LOAD PC WITH TRAPPED VIRTUAL ADDRESS
                                ;28618          RC[SC.SV]_K[SC],        ;  SAVE SC REG IN RC REG SET
U 0B50, 0018,1C39,1D80,09E9,2C00,023C   ;28619          CALL,PSL.MODE?,J/GET.PTE ;  GO GET PTE FOR TRAPPED VIRTUAL ADDRESS
                                ;28620
                                ;28621          ;01----------------------;  *** RETURN FROM GET.PTE ***
                                ;28622          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28623          RC[VA.SV]_PC,           ;  SAVE ENTRY CONTENTS OF VA IN RC REG
U 0B51, 0014,0039,0180,09C8,0000,0259   ;28624          CALL,  J/SET.PTE.M      ;  GO SET PTE (M) BIT
                                ;28625
                                ;28626          ;10----------------------;  *** MEM MNGMT FAULT RETURN ***
                                ;28627          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28628          RC[VA.SV]_PC,           ;  SAVE ENTRY CONTENTS OF VA
U 0B52, 0014,1138,0180,09C8,0000,0D7C   ;28629          LAST.REF?,  J/M.FLT     ;
                                ;28630
                                ;28631          ;11----------------------;  *** RETURN FROM SET PTE [M] ***
                                ;28632          Q_ID[D.SV],             ;  GET ENTRY CONTENTS OF D REG
                                ;28633          PC&VA_RC[PC.SV],        ;  RESTORE PC TO ENTRY CONTENTS
                                ;28634          CLR.NEST.ERR,           ;  RESET FLG INDICATING PC IN RC (PC.SV)
U 0B53, 0010,0038,B9F0,2D61,2A00,0E48   ;28635          J/M.TBM.A               ;  MERGE WITH TBUF MISS ROUTINE
```

```
                                ;28636  .TOC    ""      Memory management     : UNALIGNED DATA MICRO-TRAP ROUTINE''
                                ;28637
                                ;28638          ;*************************************
                                ;28639          ;                                   *
                                ;28640          ;UNALIGNED DATA MICRO-TRAP ROUTINE  *
                                ;28641          ;                                   *
                                ;28642          ;*************************************
                                ;28643
                                ;28644          ;THIS ROUTINE IS INITIATED BY A UNALIGNED DATA TRAP.
                                ;28645          ;IT HANDLES SECOND REFERENCES REQUIRED FOR WORD AND LONG WORD
                                ;28646          ;MEMORY REFERENCES THAT ARE NOT ALIGNED WITHIN A LONG WORD
                                ;28647          ;MEMORY BOUNDRY BUT ARE IN THE SAME PAGE.  REFERENCES
                                ;28648          ;THAT CROSS PAGE BOUNDRIES ARE HANDLED BY THE PAGE BOUNDRY
                                ;28649          ;MICRO-TRAP ROUTINE.  AFTER DOING THE SECOND REFERENCE, THIS
                                ;28650          ;ROUTINE RETURNS CONTROL TO THE TRAPPED MICRO-FLOW.
                                ;28651
                                ;28652          ;PROCESSOR REGISTERS USED BY THIS ROUTINE ARE SAVED BY IT
                                ;28653          ;RESTORED TO THEIR ENTRY CONTENTS BEFORE EXITING THIS ROUTINE
                                ;28654  101:
                                ;28655  M.UD.BGN:
                                ;28656          ;----------------------;
                                ;28657          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28658          STOP.IB,                ;  STOP IBUF FROM MAKING MEMORY REFERENCES
                                ;28659          RC[PC.SV]_PC,           ;  SAVE ENTRY CONTENTS OF PC IN RC REG
                                ;28660          SET.NEST.ERR,           ;  SET FLAG THAT PC IS IN RC (PC.SV)
U 0101, 1014,0038,0180,09E1,2C00,0E4C   ;28661          PC_VA                   ;  LOAD PC WITH TRAPPED VIRTUAL ADDRESS
                                ;28662
                                ;28663          ;----------------------;
                                ;28664          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28665          RC[VA.SV]_PC,           ;  SAVE ENTRY CONTENTS OF VA IN RC REG
                                ;28666          VA VA+4,                ;  INCREMENT TRAPPED VIRTUAL ADDRESS
U 0E4C, 0014,1138,0180,09CB,0000,0D2C   ;28667          J/M.UDREF, LAST.REF?    ;  DETERMINE TYPE OF REQUEST TRAPPED
                                ;28668
                                ;28669  =1100   ;BRANCH ON SAVED CONTEXT OF TRAPPED MEMORY REFERENCE(BEN/LAST.REF)
                                ;28670
                                ;28671  M.UDREF:
                                ;28672          ;1100----------------;  *** READ MEMORY REFERENCE/READ CHECK ***
                                ;28673          D_CACHE[SECOND.REF],    ;  DO 2ND MEMORY REFERENCE (NO TRAP)
                                ;28674          RC[LC.SV]_LC,           ;  SAVE ENTRY CONTENTS OF LC IN RC
U 0D2C, 0010,0038,0180,41C0,3000,0E4D   ;28675          J/M.UD.A                ;  GO TO RESTORE REGS
                                ;28676
                                ;28677          ;1101----------------;  ** INTERLOCK READ ***
                                ;28678          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0D2D, 0000,003D,0180,0800,0000,0EE0   ;28679          CALL,  J/M.ERROR        ;  ILLEGAL CONDITION
                                ;28680
                                ;28681          ;1110----------------;  *** READ MEMORY REFERENCE/WRITE CHECK ***
                                ;28682          D_CACHE.WCHK[SECOND.REF],;  DO 2ND MEMORY REFERENCE (NO TRAP)
                                ;28683          RC[LC.SV]_LC,           ;  SAVE ENTRY CONTENTS OF LC IN RC
U 0D2E, 0010,0038,0180,51C0,3000,0E4D   ;28684          J/M.UD.A                ;  GO TO RESTORE REGS
                                ;28685
                                ;28686          ;1111----------------;  *** WRITE MEMORY REFERENCE ***
                                ;28687          CACHE[SECOND.REF]_D,    ;  DO 2ND MEMORY REFERENCE (NO TRAP)
                                ;28688          RC[LC.SV]_LC,           ;  SAVE ENTRY CONTENTS OF LC IN RC
U 0D2F, 0010,0038,0180,31C0,3000,0E4D   ;28689          J/M.UD.A                ;  GO TO RESTORE REGS
                                ;28690  =;END
```

```
                                    ;28691  M.UD.A:
                                    ;28692          ;-----------------------;
                                    ;28693          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28694          START.IB,               ;  START IBUF AGAIN
                                    ;28695          PC&VA_RC[PC.SV],        ;  ENTRY CONTENTS TO PC
U 0E4D, 3010,0038,0180,0961,2A00,0E50   ;28696          CLR.NEST.ERR            ;  RESET FLAG NOTING PC IN RC <PC.SV>
                                    ;28697
                                    ;28698          ;-----------------------;
                                    ;28699          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E50, 0010,0038,0180,0948,0200,0E51   ;28700          VA_RC[VA.SV]            ;  ENTRY CONTENTS TO VA
                                    ;28701
                                    ;28702          ;-----------------------;
                                    ;28703          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28704          LC_RC[LC.SV],           ;  ENTRY CONTENTS TO LC
U 0E51, 0000,003E,0180,0940,0000,0000   ;28705          RETURN0                 ;  RETURN TO TRAPPED MICRO-FLOW
```

```
                                        ;28706  .TOC     ""      Memory management      : PAGE BNDRY MICRO-TRAP ROUTINE''
                                        ;28707
                                        ;28708          ;*********************************
                                        ;28709          ;                               *
                                        ;28710          ; PAGE BNDRY MICRO-TRAP ROUTINE *
                                        ;28711          ;                               *
                                        ;28712          ;*********************************
                                        ;28713
                                        ;28714
                                        ;28715          ;THIS ROUTINE IS INITIATED BY A PAGE BOUNDARY MICRO-TRAP. IT
                                        ;28716          ;PROBES THE NEXT PAGE, IF PTE IS NOT IN TBUF, FETCHES IT AND
                                        ;28717          ;DOES THE NECESSARY CHECKS TO ASSURE THAT THE MEMORY REFERENCE
                                        ;28718          ;IS ALLOWED. IF SO, THE MEMORY REFERENCES ARE DONE.
                                        ;28719
                                        ;28720          ;PROCESSOR REGISTERS USED BY THIS ROUTINE ARE SAVED AND RESTORED
                                        ;28721          ;TO THEIR ENTRY VALUES BEFORE EXITING THIS ROUTINE.
                                        ;28722
                                        ;28723          ;THIS ROUTINE CAN BE FAULTED BY A NOT VALID PTE AND BY ACCESS
                                        ;28724          ;VIOLATIONS.
                                        ;28725
                                        ;28726          ;SEE VAX, SRM, CHAPTER #5, FOR ADDITIONAL DETAILS.
                                        ;28727
                                        ;28728
                                        ;28729
                                        ;28730
                                        ;28731
                                        ;28732
                                        ;28733  102:
                                        ;28734  PG.BDRY: ;----------------------;  ***MICRO-TRAP ENTRY***
                                        ;28735          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;28736          STOP.IB,                ;  STOP IBUF FROM MAKING MEMORY REFERENCES
U 0102, 1018,0038,1D80,09E8,0000,0E52   ;28737          RC[SC.SV]_K[SC]         ;  SAVE ENTRY CONTENTS OF SC-REG
                                        ;28738
                                        ;28739          ;----------------------;
                                        ;28740          ID[D.SV]_D,             ;  SAVE ENTRY CONTENTS OF D-REG
                                        ;28741          D_Q,                    ;  SET-UP ENTRY CONTENT TO SAVE
                                        ;28742          RC[PC.SV]_PC,  PC_VA,   ;  SAVE ENTRY CONTENTS OF PC
U 0E52, 0C14,0038,B980,3DE1,2C00,0E53   ;28743          SET.NEST.ERR            ;  SET FLAG NOTING PC SAVED IN RC[PC.SV]
                                        ;28744
                                        ;28745          ;----------------------;
                                        ;28746          ID[Q.SV]_D,             ;  SAVE ENTRY CONTENTS OF Q IN D-REG
U 0E53, 0814,0038,BD80,3DC8,0000,0E54   ;28747          D&RC[VA.SV]_PC          ;  SAVE ENTRY CONTENTS OF VA-REG
                                        ;28748
                                        ;28749          ;----------------------;
                                        ;28750          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E54, 0019,0014,0180,0801,0200,0B58   ;28751          PC&VA_D+K[.8]           ;  INCREMENT VA+8 FOR PROBE OF NEXT PAGE
```

```
                                    ;28752  =00     ;CALL CONSTRAINT BLOCK
                                    ;28753
                                    ;28754          ;00--------------------;
                                    ;28755          RC[LC.SV]_LC,          ;  SAVE ENTRY CONTENTS OF LC-REG
                                    ;28756          TEST.TB.RCHK,          ;
U 0B58, 0010,1139,0180,01C0,0000,0D5C ;28757        CALL,J/PROBE.VA,LAST.REF?;  GO PROBE NEXT PAGE
                                    ;28758
                                    ;28759
                                    ;28760
                                    ;28761          ;01--------------------;  ***GET PTE RTN RETURN***
                                    ;28762          Q_ID[D.SV],            ;  GET D'S ENTRY CONTENTS
                                    ;28763          PC&VA_RC[VA.SV],       ;  LOAD PC & VA/VA'S ENTRY CONTENTS
U 0B59, 0010,0038,B9F0,2D49,0200,0E55 ;28764        J/M.PG.01              ;
                                    ;28765
                                    ;28766          ;10--------------------;  MEMORY FAULT RETURN
                                    ;28767          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B5A, 0000,113C,0180,0800,0000,0D7C ;28768        LAST.REF?,  J/M.FLT    ;
                                    ;28769
                                    ;28770          ;11--------------------;  ***TEST.MBIT RTN RETURN***
                                    ;28771          Q_ID[D.SV],            ;  GET D'S ENTRY CONTENTS
U 0B5B, 0010,0038,B9F0,2D49,0200,0E55 ;28772        PC&VA_RC[VA.SV]        ;  LOAD PC&VA/VA'S ENTRY CONTENTS
                                    ;28773  =;END
                                    ;28774
                                    ;28775  M.PG.01:;--------------------;
                                    ;28776          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28777          D_Q,                   ;  RESTORE D TO ENTRY CONTENTS
U 0E55, 0C00,113C,0180,0800,0000,0D3C ;28778        LAST.REF?              ;
                                    ;28779
                                    ;28780
                                    ;28781  =1100   ;BRANCH ON TRAPPED REFERENCE TO DECODE REFERENCE TYPE (BEN/LAST.REF)
                                    ;28782
                                    ;28783
                                    ;28784          ;1100------------------;  ***READ WITH READ CHK REF***
                                    ;28785          D_CACHE[RETRY.NO.TRAP], ;  DO FIRST MEMORY REFERENCE (NO TRAP)
U 0D3C, 0000,1D3C,0180,4000,3400,0053 ;28786        ALIGNED?, J/M.PG.03    ;
                                    ;28787
                                    ;28788          ;1101------------------;  ***INTERLOCK READ REF***
                                    ;28789          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0D3D, 0000,003D,0180,0800,0000,0EE0 ;28790        CALL,  J/M.ERROR       ;  ILLEGAL CONDITION
                                    ;28791
                                    ;28792          ;1110------------------;  ***READ/WRITE CHK REF***
                                    ;28793          D_CACHE.WCHK[RETRY.NO.TRAP],;  DO FIRST MEMORY REFERENCE
U 0D3E, 0000,1D3C,0180,5000,3400,0053 ;28794        ALIGNED?, J/M.PG.03    ;
                                    ;28795
                                    ;28796          ;1111------------------;  ***WRITE REFERENCE***
                                    ;28797          CACHE[RETRY.NO.TRAP]_D, ;  DO FIRST MEMORY REFERENCE (NO TRAP)
U 0D3F, 0000,1D3C,0180,3000,3400,0053 ;28798        ALIGNED?, J/M.PG.03    ;
```

```
                                ;28799  =10*11  ;BRANCH ON  UNALIGNED DATA (BEN/TB.TEST)
                                ;28800
                                ;28801  M.PG.03:;10*11-----------------;  *** UNALIGNED DATA***
                                ;28802          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28803          VA_VA+4,               ;  INCREMENT FOR SECOND REFERENCE
U 0053, 0000,113C,0180,0803,0000,0D4C   ;28804          LAST.REF?,J/M.PG.09    ;
                                ;28805
                                ;28806          ;11*11-----------------;  *** DATA ALIGNED***
                                ;28807          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28808          PC&VA_RC[PC.SV],       ;  ENTRY CONTENTS TO PC
                                ;28809          CLR.NEST.ERR,          ;  RESET PC FLAG NOTING PC IN RC(PC.SV)
                                ;28810          START.IB,              ;  RESTART IBUF
U 005B, 3010,0038,0180,0961,2A00,0E5A   ;28811          J/M.PG.13              ;
                                ;28812  =;END
                                ;28813
                                ;28814  =1100   ;BRANCH ON TRAPPED REFERENCE TO DETERMINE REFERENCE TYPE
                                ;28815
                                ;28816  M.PG.09:;1100--------------------;  ***READ WITH READ CHK REF***
                                ;28817          D_CACHE[SECOND.REF],   ;  RETRY TRAPPED MEMORY REFERENCE
U 0D4C, 0000,003C,0180,4000,3000,0E58   ;28818          J/M.PG.11              ;
                                ;28819
                                ;28820          ;1101--------------------;  ***INTERLOCK READ REF***
                                ;28821          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0D4D, 0000,003D,0180,0800,0000,0EE0   ;28822          CALL, J/M.ERROR        ;  ILLEGAL CONDITION
                                ;28823
                                ;28824          ;1110--------------------;  ***READ/WRITE CHK REF***
                                ;28825          D_CACHE.WCHK[SECOND.REF],;  RETRY TRAPPED MEMORY REFERENCE
U 0D4E, 0000,003C,0180,5000,3000,0E58   ;28826          J/M.PG.11              ;
                                ;28827
                                ;28828          ;1111--------------------;  ***WRITE REFERENCE***
                                ;28829          CACHE[SECOND.REF]_D,   ;  RETRY TRAPPED MEMORY REFERENCE
U 0D4F, 0000,003C,0180,3000,3000,0E58   ;28830          J/M.PG.11              ;
                                ;28831
                                ;28832  M.PG.11: ;----------------------;
                                ;28833          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28834          PC&VA_RC[PC.SV],       ;  RESTORE ENTRY CONTENTS TO PC
                                ;28835          CLR.NEST.ERR,          ;
U 0E58, 3010,0038,0180,0961,2A00,0E5A   ;28836          START.IB               ;  RESTART IBUFF
                                ;28837
                                ;28838  M.PG.13:;----------------------;
                                ;28839          VA_RC[VA.SV],          ;  ENTRY CONTENTS TO VA-REG
U 0E5A, 0010,0038,BDF0,2D48,0200,0E5C   ;28840          Q_RC[Q.SV]             ;  ENTRY CONTENTS TO Q-REG
                                ;28841
                                ;28842          ;----------------------;
                                ;28843          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;28844          LC_RC[LC.SV],          ;  RESTORE LC TO ENTRY CONTENTS
U 0E5C, 0000,003E,0180,0940,0000,0000   ;28845          RETURN0                ;  RETURN TO TRAPPED U-INSTRUCTION
```

```
                                ;28846  .TOC    ""      Memory management     : PROTECTION MICRO-TRAP ROUTINE''
                                ;28847
                                ;28848          ;************************;
                                ;28849          ;                        ;
                                ;28850          ; PROTECTION MICRO-TRAP  ;
                                ;28851          ;                        ;
                                ;28852          ;************************;
                                ;28853
                                ;28854          ;THIS ROUTINE DOES THE NECESSARY SET-UP TO MERGE INTO
                                ;28855          ;THE NORMAL MEMORY MANAGEMENT FAULT HANDLING FLOWS.
                                ;28856
                                ;28857  104:
                                ;28858  PROT.CHK: ;----------------------;
U 0104, 0018,0038,1D80,F9E8,0000,0E5E   ;28859          RC[SC.SV]_K[SC]         ;  SAVE SC-REG
                                ;28860
                                ;28861          ;----------------------;
                                ;28862          RC[PC.SV]_PC,           ;  SAVE CURRENT PC VALUE
                                ;28863          SET.NEST.ERR,           ;  NOTE CURRENT PC IN RC[PC.SV]
                                ;28864          PC_VA,                  ;  MOVE VA REG TO PC REG
                                ;28865          ID[D.SV]_D,             ;  SAVE D-REG
U 0E5E, 0C14,0038,B980,3DE1,2C00,0E60   ;28866          D_Q                     ;  MOVE Q-REG TO D-REG
                                ;28867
                                ;28868  M.PROT3:;----------------------;
                                ;28869          RC[VA.SV]_PC,           ;  SAVE VA'S ENTRY CONTENTS
U 0E60, 0014,0038,BD80,3DC8,0000,0E61   ;28870          ID[Q.SV]_D              ;  SAVE ENTRY CONTENTS OF Q-REG
                                ;28871
                                ;28872  ; ************************************************
                                ;28873  ; * Patch no. 035, PCS 0E60 trapped to WCS 1167 *
                                ;28874  ; ************************************************
                                ;28875
                                ;28876          ;----------------------;
U 0E61, 0014,0038,0180,F9F0,0000,0E62   ;28877          RC[VA.REF]_PC           ;  SET-UP FAULTING VIRTUAL PARA
                                ;28878          ;
                                ;28879
                                ;28880  MM.PA.35:
                                ;28881          ;----------------------;
                                ;28882          D_K[.8],                ;  SET-UP 2ND FAULT PARAMETER
U 0E62, 0818,0038,0180,F800,0000,0EA2   ;28883          J7M.FLT.LC              ;  MERGE INTO NORMAL FAULT FLOWS
                                ;28884
                                ;28885  IB.PROT:;----------------------;
                                ;28886          RC[PC.SV]_PC,           ;  SAVE CURRENT PC VALUE
                                ;28887          SET.NEST.ERR,           ;  NOTE CURRENT PC IN RC[PC.SV]
                                ;28888          PC_VIBA,                ;  MOVE VA REG TO PC REG
                                ;28889          ID[D.SV]_D,             ;  SAVE D-REG
                                ;28890          D_Q,                    ;  MOVE Q-REG TO D-REG
U 0E63, 0C14,0038,B980,3DE2,2C00,0E60   ;28891          J7M.PROT3               ;  MERGE INTO PROTECTION CHECK FLOW
```

```
                                        ;28892  .TOC     "       Memory management      : IBUF PTE FETCH ROUTINE"
                                        ;28893
                                        ;28894          ;****************************************;
                                        ;28895          ;                                        ;
                                        ;28896          ; IBUF PTE FETCH AND FILL ROUTINE        ;
                                        ;28897          ;                                        ;
                                        ;28898          ;****************************************;
                                        ;28899
                                        ;28900  ;DESCRIPTION/ALGORITHM:
                                        ;28901
                                        ;28902          ;THIS ROUTINE IS USED TO FETCH PTE'S WHEN THE IBUF
                                        ;28903          ;GETS A TBUF-MISS ON ATTEMPTING TO TRANSLATE A VIRTUAL
                                        ;28904          ;ADDRESS IN THE VIBA REGISTER.
                                        ;28905
                                        ;28906          ;THIS ROUTINE WILL FETCH THE REQUIRED PTE AND, IF IT
                                        ;28907          ;IS VALID, WILL LOAD IT INTO THE TBUF AND INITIATE A
                                        ;28908          ;IBUF FILL MEMORY REFERENCE.
                                        ;28909
                                        ;28910          ;IF THE PTE IS NOT VALID OR AN ACCESS VIOLATION IS ENCOUNTERED
                                        ;28911          ;THIS ROUTINE WILL BE FAULTED.
                                        ;28912
                                        ;28913
                                        ;28914  ;CALLING SEQUENCE:      CALL,  J/IB.TBM
                                        ;28915
                                        ;28916  ;INPUT REQUIREMENTS:
                                        ;28917
                                        ;28918          ;THIS ROUTINE   SAVES SC, PC, D, & Q
                                        ;28919          ;REGISTERS IN THEIR RESPECTIVE RC AND ID REGISTER SET
                                        ;28920          ;SAVE LOCATIONS  AND RESTORES THEM BEFORE EXITING
                                        ;28921
                                        ;28922  ;STORAGE/REGISTER ALLOCATION:
                                        ;28923
                                        ;28924          ;VA <- VIBA
                                        ;28925          ;RC[PC.SV] <- PC
                                        ;28926          ;RC[SC.SV] <- SC-REG
                                        ;28927          ;RC[PTE.VA] <- EXCEPTION PARA. IF FAULT
                                        ;28928          ;ID[D.SV] <- D-REG
                                        ;28929          ;ID[FPDA] <- (M.FLT.23) AS FPD ADDR IF FAULT
                                        ;28930
                                        ;28931  ;RETURN(S) TO CALLER:   RETURN2
                                        ;28932
                                        ;28933
                                        ;28934  IB.TBM: ;----------------------;
                                        ;28935          SET.NEST.ERR,           ;  SET NESTED ERROR FLAG
                                        ;28936          STOP.IB,                ;
                                        ;28937          RC[PC.SV]_PC,           ;  SAVE PC IN RC REG
                                        ;28938          PC_VIBA,                ;  GET VIBA
                                        ;28939          ID[D.SV]_D,             ;  SAVE CONTENTS OF D-REG
U 0E64, 1C14,0038,B980,3DE2,2C00,0E66   ;28940          D_Q                     ;  MOVE CONTENTS OF Q-REG TO D-REG
                                        ;28941
                                        ;28942  IB.TBM2:;----------------------;
                                        ;28943          VA_PC,                  ;  MOVE CONTENTS OF VIBA TO BOTH VA AND PC-REGS
U 0E66, 0014,0038,BD80,3C00,0200,0B60   ;28944          ID[Q.SV]_D              ;  SAVE CONTENTS OF Q-REG
```

```
                                    ;28945  =00     ;CALL CONSTRAINT BLOCK
                                    ;28946
                                    ;28947  M.IB.PTE: ;00--------------------;
                                    ;28948          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28949          RC[SC.SV]_K[SC],         ;  SAVE ENTRY CONTENTS OF SC-REG
                                    ;28950          CALL,                    ;
U 0B60, 0018,1C39,1D80,09E8,0000,023C ;28951        PSL.MODE?,J/GET.PTE      ;  GO GET PTE
                                    ;28952
                                    ;28953          ;01--------------------;
                                    ;28954          START.IB,                ;  RESTART IBUF
                                    ;28955          SC_RC[SC.SV],            ;  RESTORE SC TO ENTRY CONTENTS
                                    ;28956          MCT/READ.V.NEWPC,        ;  DO XLATION AND GET NEXT IBUF LONG WORD
U 0B61, 3010,0038,0180,6168,0082,0E68 ;28957        J/M.IB.01                ;
                                    ;28958
                                    ;28959          ;10--------------------;
                                    ;28960          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B62, 0059,0034,65C0,0800,0000,0B63 ;28961        Q_D.AND.K[.10].RIGHT     ;
                                    ;28962
                                    ;28963          ;------------------------;
                                    ;28964          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;28965          RC[PTE.VA]_D.OR.Q,       ;  SAVE 2ND PARAMTER IN RC REG SET
U 0B63, 001D,0030,0180,09D0,0000,0B68 ;28966        J/M.FLT.23               ;  GO TO FAULT HANDLING ROUTINE
                                    ;28967  =;END
                                    ;28968
                                    ;28969  M.IB.01:
                                    ;28970          ;------------------------;
                                    ;28971          PC&VA_RC[PC.SV],         ;  RESTORE PC TO ENTRY VALUE
U 0E68, 0010,0038,B9F0,2D61,0200,0E69 ;28972        Q_ID[D.SV]               ;
                                    ;28973
                                    ;28974          ;------------------------;
                                    ;28975          D_Q,                     ;  RESTORE D TO ENTRY CONTENTS
                                    ;28976          C[R.NEST.ERR,            ;  CLEAR NESTED ERROR FLAG
                                    ;28977          Q_ID[Q.SV],              ;  RESTORE Q TO ENTRY CONTENTS
U 0E69, 0C00,003E,BDF0,2C00,2800,0002 ;28978        RETURN2                  ;  RETURN TO CALLER
                                    ;28979  =00
                                    ;28980  M.FLT.23: ;00--------------------;  SET-UP FPDA ON USTACK
                                    ;28981          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B68, 0000,003D,0180,0800,0000,0E6A ;28982        CALL,   J/M.FLT.25       ;
                                    ;28983
                                    ;28984          ;01--------------------;
                                    ;28985          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B69, 0000,003C,0180,0800,0000,0D8F ;28986        J/M.FLT.03               ;  MERGE BACK INTO NORMAL FLOW
                                    ;28987
                                    ;28988          ;10--------------------;
                                    ;28989          MCT/MEM.NOP,             ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B6A, 0000,003C,0180,0800,0000,0D8F ;28990        J/M.FLT.03               ;  MERGE BACK INTO NORMAL FLOW
                                    ;28991  =;END
```

```
                                    ;28992  M.FLT.25: ;----------------------;
U 0E6A, 0000,003C,81F0,2C00,0000,0E6C ;28993          Q_ID[USTACK]           ;  READ FPDA OFF USTACK
                                    ;28994
                                    ;28995          ;----------------------;
                                    ;28996          MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E6C, 0C00,003C,0180,0800,0000,0E6D ;28997          D_Q                    ;
                                    ;28998
                                    ;28999          ;----------------------;
                                    ;29000          ID[FPDA]_D,            ;  FINISH SET-UP OF FPDA
U 0E6D, 0000,003C,B580,3C00,0000,0EA9 ;29001          J/M.FLT.PRB            ;  MERGE INTO NORMAL FAULT FLOW
                                    ;29002
                                    ;29003
                                    ;29004
                                    ;29005
                                    ;29006
                                    ;29007
                                    ;29008  ;SUBROUTINE TO HANDLE IBUF TBMISS WITHOUT DESTROYING VA
                                    ;29009  ; USED BY BRANCH INSTRUCTIONS FOR GETTING BDEST
                                    ;29010  ; Q GETS CLOBBERED.
                                    ;29011
                                    ;29012          ;-------------------------------;
                                    ;29013  IB.TBR: SET.NEST.ERR,STOP.IB,
                                    ;29014          RC[PC.SV]_PC,                  ;SAVE PC DURING REFILL
                                    ;29015          PC_VA,                         ;GET OUT VA
U 0E6E, 1014,0038,B980,3DE1,2C00,0914 ;29016          ID[D.SV]_D                     ;SAVE CURRENT D REGISTER
                                    ;29017
                                    ;29018  =0*     ;-------------------------------;
                                    ;29019          D_PC,                          ;GET VALUE FROM VA
                                    ;29020          PC_VIBA,                       ;NOW LOAD PC WITH MISS ADDRESS
U 0914, 0814,0039,0180,F802,0000,0E66 ;29021          CALL,J/IB.TBM2                 ;GO REFILL TRANSLATION BUFFER
                                    ;29022
                                    ;29023          ;-------------------------------;
                                    ;29024          VA_Q,                          ;RESTORE FAULTED VA
                                    ;29025          Q_IB.BDEST,                    ;
U 0916, 7001,2B3E,01F0,F800,0200,0000 ;29026          IB.TEST?,RETURN0               ;
```

```
;29027  .TOC    ""      Memory management     : PROBE VIRTUAL ADDRESS ROUTINE''
;29028
;29029          ;****************************
;29030          ;                            *
;29031          ; PROBE VIRTUAL ADDRESS RTN*
;29032          ;                            *
;29033          ;****************************
;29034
;29035          ;DESCRIPTION:
;29036          ;-----------------------------
;29037
;29038          ;THIS ROUTINE IS USED TO CHECK THE VIRTUAL ADDRESS IN VA-REG
;29039          ;TO DETERMINE IF THE REQUIRED PTE IS IN THE TBUF, TO FETCH
;29040          ;IT IF NOT; AND TO CHECK FOR PROTECTION VIOLATION.
;29041
;29042          ;THIS ROUTINE CAN BE FAULTED BY PTE NOT VALID, AND BY
;29043          ;PROTECTION VIOLATIONS.
;29044
;29045          ;CALLING SEQUENCE FOR ROUTINE:  'TEST.TB.RCHK,
;29046          ;-----------------------------  CALL,  J/PROBE.VA,  LAST.REF?''
;29047
;29048          ;INPUT REQUIREMENTS TO USE THIS ROUTINE
;29049          ;-----------------------------
;29050
;29051          ;SAME AS 'GET. PTE'' ROUTINE
;29052
;29053
;29054          ;REGISTER ALLOCATION:
;29055          ;-----------------------------
;29056
;29057          ;RC[VA.REF] <- VIRTUAL ADDRESS
;29058          ;D.REG <- FAULT PARAMETER IF FAULT
;29059
;29060          ;OUTPUTS & RESULTS (IF PTE(S) IS VALID)
;29061          ;-----------------------------
;29062
;29063          ;THIS ROUTINE'S OUTPUTS ARE THE SAME AS GET.PTE & SET.MBIT
;29064          ;IF THEY ARE CALLED, OTHERWISE SAME AS ON INPUT ON RETURN
;29065          ;TO CALLING SEQUENCE.
;29066
;29067          ;RETURN TO CALLER
;29068          ;-----------------------------
;29069
;29070          ;THIS ROUTINE RETURNS IN THE FOLLOWING WAYS:
;29071
;29072          ;       1)  RETURN1     TB-HIT OR GETPTE ROUTINE RETURN
;29073
;29074          ;       2)  RETURN2     MEM FAULT RETURN
;29075
;29076          ;       3)  RETURN3     TEST.PTE.M ROUTINE RETURN
```

```
                                        ;29077  =1100   ;BRANCH ON TRAPPED REFERENCE TO DECODE ACCESS TYPE SPECIFIED IE RCHK
                                        ;29078
                                        ;29079  PROBE.VA:
                                        ;29080          ;1100-------------------;  *** READ CHECK ***
                                        ;29081          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29082          RC[VA.REF]_PC,          ;  SAVE VA BEING PROBED IN RC
U 0D5C, 0014,1D38,0180,09F0,0000,0178   ;29083          J/M.PG.15, TB.TEST?     ;
                                        ;29084
                                        ;29085          ;1101-------------------;  *** ILLEGAL ENTRY ***
                                        ;29086          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0D5D, 0000,003D,0180,0800,0000,0EE0   ;29087          CALL, J/M.ERROR         ;
                                        ;29088
                                        ;29089          ;1110-------------------;  *** REDUNDANT WRITE CHECK ***
                                        ;29090          RC[VA.REF]_PC,          ;  SAVE VA BEING PROBED IN RC
                                        ;29091          TEST.TB.WCHK,           ;
U 0D5E, 0014,0038,0180,11F0,0000,0E70   ;29092          J/M.PG.20               ;
                                        ;29093
                                        ;29094          ;1111-------------------;  *** REDUNDANT WRITE CHECK ***
                                        ;29095          RC[VA.REF]_PC,          ;  SAVE VA BEING PROBED IN RC
                                        ;29096          TEST.TB.WCHK,           ;  SET-UP FOR TB. TEST
U 0D5F, 0014,0038,0180,11F0,0000,0E70   ;29097          J/M.PG.20               ;
                                        ;29098
                                        ;29099
                                        ;29100
                                        ;29101  =11*00  ;BRANCH ON TEST.TB.RCHK RESULTS (BEN)/TB.TEST)
                                        ;29102
                                        ;29103  M.PG.15: ;11*00-----------------;  *** TB-HIT ***
                                        ;29104          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0178, 0000,003E,0180,0800,0000,0001   ;29105          RETURN1                 ;
                                        ;29106
                                        ;29107          ;11*01-----------------;  *** ILLEGAL ENTRY ***
                                        ;29108          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0179, 0000,003D,0180,0800,0000,0EE0   ;29109          CALL, J/M.ERROR         ;
                                        ;29110
                                        ;29111          ;11*10-----------------;  *** PROTECTION VIOLATION ***
                                        ;29112          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29113          D_K[.8],                ;  EXCEPTION PARA TO D-REG
U 017A, 0818,1138,0180,0800,0000,0D7C   ;29114          LAST.REF?,  J/M.FLT     ;
                                        ;29115
                                        ;29116
                                        ;29117          ;11*11-----------------;  *** TB MISS ***
                                        ;29118          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 017B, 0000,1C3C,0180,0800,0000,023C   ;29119          J/GET.PTE, PSL.MODE?    ;  JUMP TO GET.PTE ROUTINE
                                        ;29120                                  ;  AND RETURN TO CALLER AFTER GETTING THE PTE.
                                        ;29121
                                        ;29122  ; ************************************************
                                        ;29123  ; * Patch no. 042, PCS 017B trapped to WCS 1168 *
                                        ;29124  ; ************************************************
```

```
                                        ;29125          ;PROBE/WRITE CHECK MICRO-FLOW
                                        ;29126          ;---------------------
                                        ;29127
                                        ;29128  M.PG.20: ;---------------------;  *** TB-HIT ***
                                        ;29129          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E70, 0000,1D3C,0180,0800,0C00,0238   ;29130          J/M.PG.22, TB.TEST?     ;  DO TB TEST
                                        ;29131
                                        ;29132
                                        ;29133  =11*00  ;BRANCH ON TEST.TB. WCHK RESULTS (BEN/TB.TEST)
                                        ;29134
                                        ;29135  M.PG.22: ;11*00-----------------;  *** TB-HIT ***
                                        ;29136          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0238, 0000,003E,0180,0800,0000,0001   ;29137          RETURN1                 ;
                                        ;29138
                                        ;29139          ;11*01-----------------;  *** 1ST MODIFY TO PAGE ***
                                        ;29140          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0239, 0000,003C,0180,0800,0000,0B70   ;29141          J/M.PG.25               ;
                                        ;29142
                                        ;29143          ;11*10-----------------;  *** PROTECTION VIOLATION ***
                                        ;29144          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29145          D_K[.C],                ;  FAULT PARAMETER TO D-REG
U 023A, 0818,1138,8580,0800,0000,0D7C   ;29146          LAST.REF?,  J/M.FLT     ;
                                        ;29147
                                        ;29148          ;11*11-----------------;  *** TB-MISS ***
                                        ;29149          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 023B, 0000,003C,0180,0800,0000,0B70   ;29150          J/M.PG.25               ;
                                        ;29151  =;END
                                        ;29152
                                        ;29153  =00     ;CALL SEQUENCE  CONSTRAINT BLOCK
                                        ;29154
                                        ;29155  M.PG.25: ;00--------------------;
                                        ;29156          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B70, 0000,1C3D,0180,0800,0000,023C   ;29157          CALL,PSL.MODE?,J/GET.PTE ;  GO GET PTE
                                        ;29158
                                        ;29159          ;01--------------------;  *** RETURN FROM GET.PTE ***
                                        ;29160          TEST.TB.WCHK,           ;  GO SET PTE <M> IF NECESSARY
U 0B71, 0000,113C,0180,1000,0000,0D6C   ;29161          J/TEST.PTE.M,LAST.REF?  ;  THEN RETURN DIRECTLY TO CALLER
                                        ;29162
                                        ;29163
                                        ;29164          ;10--------------------;  *** MEM FAULT RETURN ***
                                        ;29165          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B72, 0000,113C,0180,0800,0000,0D7C   ;29166          LAST.REF?,  J/M.FLT     ;
                                        ;29167  =;END
```

```
;29168  .TOC    ""      Memory management     : PTE<M> TEST & SET ROUTINE""
;29169
;29170          ;*******************************
;29171          ;                              *
;29172          ; PTE<M> TEST & SET ROUTINE    *
;29173          ;                              *
;29174          ;*******************************
;29175
;29176          ;DESCRIPTION:
;29177          ;-------------------------------------------------------------
;29178          ;THIS ROUTINE IS USED TO SET THE PTE<M> BIT IN THE PTE IN THE TBUF
;29179          ;ADDRESSED BY THE VIRTUAL ADDRESS IN THE VA REGISTER & THE INDIRECTLY
;29180          ;ADDRESSED MIRROR IMAGE OF IT IN MAIN MEMORY.  IT ALSO
;29181          ;CHECKS TO SEE IF THIS FUNCTION IS REQUIRED.  SEE VAX SRM
;29182          ;CHAPTER #5 FOR MORE DETAIL.
;29183
;29184          ;CALLING SEQUENCE:
;29185          ;-------------------------------------------------------------
;29186          ;THERE ARE TWO (2) ENTRY POINTS AND CALLING SEQUENCES FOR THIS ROUTINE
;29187          ;
;29188          ;       1) 'TEST.TB.WCHK,
;29189          ;           CALL, LAST.REF?, J/TEST.PTE.M''      USED TO TEST PTE<M>
;29190          ;                                                AND SET IF REQUIRED
;29191          ;
;29192          ;       2)  CALL, J/SET.PTE.M''          SETS PTE<M> IN TBUF & PAGE
;29193          ;                                        TABLE ADDRESSED BY VA REGISTER
;29194          ;
;29195
;29196          ;INPUT REQUIREMENTS:
;29197          ;-------------------------------------------------------------
;29198          ;THIS ROUTINE EXPECTS ALL REGISTERS TO BE AS LEFT BY FETCH PTE ROUTINE
;29199
;29200          ;REGISTER ALLOCATION:
;29201          ;-------------------------------------------------------------
;29202          ;  RC[VA.REF] <- VIRTUAL ADDRESS
;29203          ;  RC[PTE.PA] ASSUMED TO CONTAIN SPTE'S PHYSICAL ADDRESS
;29204          ;  RC[PTE.VA] ASSUMED TO CONTAIN PXPTE'S VIRTUAL ADDRESS
;29205          ;  RC[PTE.MASK] ASSUMED TO CONTAIN PTE'S MBZ MASK
;29206          ;  RC[SC.SV] ASSUMED TO CONTAIN VALUE TO RESTORE SC-REG
;29207
;29208          ;RETURN TO CALLER:
;29209          ;-------------------------------------------------------------
;29210          ;THIS ROUTINE RETURNS TO THE CALLER USING, 'RETURN3''
;29211
;29212          ;OUTPUTS & RESULTS:
;29213          ;-------------------------------------------------------------
;29214          ;THE ONLY REGISTERS NOT LEFT THE SAME AS ON ENTRY, TO THIS
;29215          ;ROUTINE, ARE:
;29216          ;               1) D-REG, PTE HAS MBIT SET
;29217          ;               2) SC-REG, RESTORED TO VALUE IN RC[SC.SV]
```

```
                                        ;29218  =1100   ;TEST FOR LAST REFERENCE SPECIFYING A WRITE CHECK (BEN/LAST.REF)
                                        ;29219
                                        ;29220  TEST.PTE.M:
                                        ;29221          ;1100--------------------;  **READ & RCHK**
                                        ;29222          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29223          SC_RC[SC.SV],           ;  RESTORE SC-REG,
U 0D6C, 0010,003A,0180,0968,0082,0003   ;29224          RETURN3                 ;  RETURN TO MAIN FLOW,
                                        ;29225                                  ;  SETTING OF PTE<M> NOT REQUIRED
                                        ;29226
                                        ;29227          ;1101--------------------;  **INTERLOCK READ (WCHK)***
                                        ;29228          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29229          RC[VA.REF]_PC,          ;  SAVE VA BEING TESTED,
U 0D6D, 0014,1D38,0180,09F0,0000,0258   ;29230          J/M.MBIT, TB.TEST?      ;  WCHK SO CHECK FOR
                                        ;29231
                                        ;29232  TEST.PTE.W:
                                        ;29233          ;1110--------------------;  **READ & WCHK**
                                        ;29234          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29235          RC[VA.REF]_PC,          ;  SAVE VIRTUAL ADDRESS
U 0D6E, 0014,1D38,0180,09F0,0000,0258   ;29236          J/M.MBIT, TB.TEST?      ;  WCHK, SO CHECK FOR
                                        ;29237                                  ;  PTE<M> NOT BEING SET
                                        ;29238
                                        ;29239          ;1111--------------------;  **WRITE & WCHK**
                                        ;29240          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29241          RC[VA.REF]_PC,          ;  SAVE VIRTUAL ADDRESS
U 0D6F, 0014,1D38,0180,C9F0,0000,0258   ;29242          J/M.MBIT, TB.TEST?      ;  WCHK, SO CHECK FOR
                                        ;29243                                  ;  PTE<M> NOT BEING SET
                                        ;29244  =;END
                                        ;29245
                                        ;29246  =11*00  ;CHECK FOR PROTECTION VIOLATION AND PTE<M>=1 (BEN/TB.TEST)
                                        ;29247
                                        ;29248  M.MBIT: ;11*00-----------------;  ***  TB.HIT  **
                                        ;29249          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29250          SC_RC[SC.SV],           ;
U 0258, 0010,003A,0180,0968,0082,0003   ;29251          RETURN3                 ;  PTE HAS PTE<M> SET ALREADY,
                                        ;29252                                  ;  RETURN TO CALLING ROUTINE
                                        ;29253  SET.PTE.M:
                                        ;29254          ;11*01-----------------;  ***1ST MODIFY TO THIS PAGE***
                                        ;29255          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29256          SC_K[.1A],              ;  SET-UP SC REG TO SET M-BIT
U 0259, 0000,1C3C,E580,0800,0084,642F   ;29257          J/M.SVA, VA31?          ;  VIRTUAL ADDRESS IN WHICH ADDR SPACE?
                                        ;29258
                                        ;29259          ;11*10-----------------;  ***ACCESS VIOLATION FAULT***
                                        ;29260          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29261          D_K[.C],                ;  SET WRITE/MODIFY INTENT
U 025A, 0818,0038,8580,0800,0000,0D7C   ;29262          J/M.FLT                 ;  IN EXCEPTION PARAMETER
                                        ;29263
                                        ;29264          ;11*11-----------------;  ***ILLEGAL CONDITION***
                                        ;29265          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 025B, 0000,003D,0180,0800,0000,0EE0   ;29266          CALL, J/M.ERROR         ;
                                        ;29267  =;END
```

```
                                ;29268          ;ENTRY TO ROUTINE TO SET PTE<M> BIT
                                ;29269          ;**********************************
                                ;29270
                                ;29271  =01111  ;IS ADDRESS IN SYSTEM OR PROCESS SPACE (BEN/TB.TEST<VA31>)
                                ;29272
                                ;29273  M.SVA:  ;01111-----------------;  ***SPTE***
                                ;29274          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29275          VA_RC[PTE.PA],          ;  GET SPTE'S PHYSICAL ADDRESS & PUT INTO VA
U 042F, 0010,0038,0180,0958,0200,0E78   ;29276          J/M.MB.SY               ;  CONTINUE SPTE<M> SET ROUTINE
                                ;29277
                                ;29278          ;11111-----------------;  ***P0PTE OR P1PTE***
                                ;29279          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29280          VA_RC[PTE.VA],          ;  PXPTE'S ADDRESS TO VA REG
U 043F, C010,0038,0180,0950,3E00,0E71   ;29281          MSC/INH.CM.ADDR         ;  INHIBIT CM ADDRESS ZERO EXTENTION
                                ;29282  =;END                           ;
                                ;29283
                                ;29284          ;CODE TO SET PXPTE FOR A PROCESS SPACE VIRTUAL ADDRESS
                                ;29285
                                ;29286          ;----------------------;
                                ;29287          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29288          D_D.ORMCT.MASK,         ;  SET PTE<M>
U 0E71, 0801,001C,0180,0800,3C00,0E72   ;29289          MSC/INH.CM.ADDR         ;  INHIBIT ZERO EXTEND OF ADDR IN VA
                                ;29290
                                ;29291          ;----------------------;
                                ;29292          CACHE_D[LONG].NOCHK,    ;  WRITE PXPTE WITH M=1 INTO CACHE
                                ;29293          MSC/INH.CM.ADDR,        ;  INHIBIT CM ZERO EXTENTION OF ADDR IN VA
U 0E72, 0000,003C,0180,2800,3C00,0E73   ;29294          J/M.MB.A                ;
                                ;29295
                                ;29296  M.MB.A:
                                ;29297          ;----------------------;
                                ;29298          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E73, 0014,0038,0180,0800,0200,0E74   ;29299          VA_PC                   ;  VIRTUAL ADDR BACK TO VA
                                ;29300
                                ;29301          ;----------------------;
                                ;29302          Q_D,                    ;  MOVE PTE TO Q-REG
                                ;29303          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29304          D_D.AND.RC[PTE.MASK],   ;  MASK OUT MBZ BITS
U 0E74, 0811,0034,01E0,0978,3C00,0E75   ;29305          MSC/INH.CM.ADDR         ;
                                ;29306
                                ;29307          ;----------------------;
                                ;29308          SC_RC[SC.SV],  D_Q,     ;  RESTORE SC-REG TO ENTRY VALUE
                                ;29309          ID[TBUF]_D,             ;  LOAD MODIFIED PXPTE INTO TBUF
U 0E75, 0C10,003A,4180,3D68,0082,0003   ;29310          RETURN3                 ;  AT LOCATION SPECIFIED BY PXVA
                                ;29311                                  ;  IN VA REG. RETURN TO CALLER
```

```
                                        ;29312          ;CODE TO SET SPTE FOR SYSTEM SPACE VIRTUAL ADDRESS
                                        ;29313
                                        ;29314  M.MB.SY: ;----------------------;
                                        ;29315          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                        ;29316          D_D.ORNOT.MASK,         ;  SET SPTE<M> BIT AND INHIBIT
U 0E78, 0801,001C,0180,0800,3C00,0E79   ;29317          MSC/INH.CM.ADDR         ;  CM ZERO EXTENTION OF ADDR IN VA,
                                        ;29318                                  ;  (PTE.PA).
                                        ;29319
                                        ;29320          ;----------------------;
                                        ;29321          CACHE.P_D[LONG],        ;  WRITE MODIFIED SPTE WITH
                                        ;29322          MSC/INH.CM.ADDR,        ;  M=1 INTO CACHE.
U 0E79, 0000,003C,0180,A800,3C00,0E73   ;29323          J/M.MB.A                ;  INHIBIT CM ZERO EXTENTION OF VA ADDR.
                                        ;29324
                                        ;29325                                  ;  MERGE INTO PXPTE SET<M> ROUTINE
                                        ;29326                                  ;  TO LOAD TBUF WITH MODIFIED
                                        ;29327                                  ;  SPTE.
```

```
;29328  .TOC     ""      Memory management     : FETCH PTE ROUTINE""
;29329
;29330          ;**********************
;29331          ;                      *
;29332          ;  FETCH PTE ROUTINE  *
;29333          ;                      *
;29334          ;**********************
;29335
;29336          ;DESCRIPTION:
;29337          ;-------------------------------------------------------------
;29338          ;THIS ROUTINE IS USED TO FETCH PTE'S FOR P0, P1, & SYSTEM
;29339          ;SPACE PAGE.  SEE VAX SRM, CHAPTER #5.
;29340          ;FOR ADDITIONAL DETAILS ON PTE RELATED OPERATIONS.
;29341
;29342          ;CALLING SEQUENCE FOR ROUTINE:    CALL,  PSL.MODE?,  J/GET.PTE
;29343          ;-------------------------------------------------------------
;29344
;29345          ;INPUT REQUIREMENTS TO USE ROUTINE:
;29346          ;-------------------------------------------------------------
;29347          ;       1.  VA & PC REGISTERS, THE VIRTUAL ADDRESS REQUIRING
;29348          ;           THE PTE.
;29349
;29350          ;       2.  OTHER REGISTERS USED BY THIS ROUTINE THAT MUST BE
;29351          ;           SAVED AND RESTORED BY USER OF THIS ROUTINE ARE Q,
;29352          ;           D, SC, PC, LC.  PC SHOULD BE SAVED IN
;29353          ;           RC(PC.SV) AND FLAG NOTING THAT PC IS IN RC SHOULD BE SET.
;29354          ;           SC IS RESTORED BY THIS ROUTINE USING RC[SC.SV]
;29355          ;           ON RETURN1'S (NOT DONE ON RETURN2).
;29356
;29357          ;REGISTER ALLOCATION AND USEAGE:
;29358          ;-------------------------------------------------------------
;29359          ;           ID REGISTERS USED BY THIS ROUTINE ARE
;29360          ;               ID[P0BR]        (CONTENTS ASSUMED RIGHT ADJUSTED)
;29361          ;               ID(P0LR)
;29362          ;               ID(P1BR),  (LENGTH + 2**21)
;29363          ;               ID(P1LR)
;29364          ;               ID(SBR)
;29365          ;               ID(SLR)
;29366
;29367          ;           RC REGISTERS USED BY THIS ROUTINE
;29368          ;               RC[VA.REF] <- VIRTUAL ADDRESS FOR PTE BEING REQUESTED
;29369          ;               RC[PTE.VA] <- VIRTUAL ADDRESS FOR PXPTE REFERENCE
;29370          ;               RC[PTE.PA] <- PHYSICAL ADDRESS FOR SPTE REFERENCE
;29371          ;               RC[PTE.MASK] ASSUMED TO CONTAIN 'FC1FFFFF'
```

```
;29372          ;OUTPUTS & RESULTS OF ROUTINE (IF PTE(S) IS VALID)
;29373          ;-----------------------------------------------------------
;29374          ;THE RESULTS OF THIS ROUTINE ARE
;29375          ;
;29376          ;       1.  VA & PC REGISTERS ARE NOT CHANGED.  SC IS LOADED WITH
;29377          ;           RC[SC.SV]
;29378          ;
;29379          ;       2.  RC REGISTERS SET-UP ARE.
;29380          ;
;29381          ;               RC[PTE.VA]      P0/P1 PTE VIRTUAL ADDR.
;29382          ;               RC[VA.REF]      VIRTUAL ADDR NEEDING PTE
;29383          ;               RC[PTE.PA]      SYS PTE PHYSICAL ADDR.
;29384          ;
;29385          ;       3.  PTE(S) LOADED INTO TRANSLATION BUFFER FOR VIRTUAL
;29386          ;           ADDRESS PRESENTED TO ROUTINE PER VA.
;29387          ;
;29388          ;       4.  "D" REGISTER CONTAINS THE FIRST LEVEL PTE AS FETCHED
;29389          ;           FROM MAIN MEMORY.
;29390          ;
;29391          ;
;29392          ;IF THE FETCHED PTE(S) IS NOT VALID, OR A LENGTH VIOLATION IS
;29393          ;DETECTED DURING THIS ROUTINE, THE APPROPRIATE FAULT WILL BE
;29394          ;GENERATED AND THIS SUBROUTINE WILL BE ABORTED.  THE FAULT HANDLING
;29395          ;MICRO-CODE EXPECTS THE CURRENT PC TO BE SAVED IN RC(PC.SV).
;29396          ;
;29397          ;RETURNS TO CALLER:
;29398          ;-----------------------------------------------------------
;29399          ;THIS ROUTINE RETURNS TO THE CALLING SEQUENCE AT:
;29400          ;
;29401          ;       1)  RETURN1     -USED IF PTE FETCHED AND LOADED INTO TBUF
;29402          ;                       ;
;29403          ;       2)  RETURN2     -USED IF MEMORY MANAGEMENT FAULT
;29404          ;                        D-REG HAS FAULT PARAMETER WITH
;29405          ;                                <00>  LENGHT VIOLATION
;29406          ;                                <01>  PXPT OR SPT REFERENCE
;29407          ;                                <02>  WRITE OR MODIFY
;29408          ;                                      (TO BE SET BY CALLER)
;29409          ;                                <03>  ACCESS VIOLATION
;29410          ;                                <04>  POSSIBLE READ ACCESS VIOL, USER
;29411          ;                                      MUST HANDLE BY SETTING <03> IF
;29412          ;                                      HE WANTS TO TAKE A VIOL FAULT
;29413          ;                                <05>  POSSIBLE WRITE ACCESS VIOL, USER
;29414          ;                                      MUST HANDLE BY SETTING <03> IF
;29415          ;                                      HE WANTS TO TAKE A VIOL FAULT
```

```
                                    ;29416          ;CHECK FOR COMPATIBILITY MODE
                                    ;29417
                                    ;29418  =11100  ;PSL(CM)?(GEN/PSL.MODE)
                                    ;29419
                                    ;29420  GET.PTE:
                                    ;29421          ;11100-----------------;  *PSL(CM) ENTRY*
                                    ;29422          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29423          SC_K[.FFF6],            ;
                                    ;29424          D_PC,          Q_0,     ;  SET-UP ADDRESS IN D REG
U 023C, 0814,0038,DDF8,0800,0084,6E7A ;29425        J/M.CM0XT               ;  SO THAT IT CAN BE ZERO EXTENDED
                                    ;29426
                                    ;29427          ;11101-----------------;  * NOT PSL(CM) ENTRY *
                                    ;29428          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29429          D_PC.LEFT,  Q_0,        ;  START EXTRACTION OF VPN
                                    ;29430          SC_K[.FFF6],            ;  FROM VA & DECODE ADDR.
U 023D, 0834,1C38,DDF8,0800,0084,6447 ;29431        J/M.TYPE,VA31-30?       ;  SPACE TYPE.
                                    ;29432
                                    ;29433          ;11110-----------------;  *REDUNDANT ENTRY / NOT PSL(CM)*
                                    ;29434          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29435          D_PC.LEFT,  Q_0,        ;
                                    ;29436          SC_K[.FFF6],            ;
U 023E, 0834,1C38,DDF8,0800,0084,6447 ;29437        J/M.TYPE,VA31-30?       ;
                                    ;29438
                                    ;29439          ;11111-----------------;  *REDUNDANT ENTRY / NOT PSL(CM)*
                                    ;29440          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29441          D_PC.LEFT,  Q_0,        ;
                                    ;29442          SC_K[.FFF6],            ;
U 023F, 0834,1C38,DDF8,0800,0084,6447 ;29443        J/M.TYPE,VA31-30?       ;
                                    ;29444  =;END
                                    ;29445
                                    ;29446          ;COMPATIBILITY MODE ADDRESS ZERO EXTEND
                                    ;29447          ;***************************************
                                    ;29448
                                    ;29449  M.CM0XT:;-----------------------;
                                    ;29450          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29451          SC_SC+1,                ;
                                    ;29452          PC&VA_D.0XT[WORD],      ;  ZERO EXTEND 16 BIT VIRTUAL ADDR
                                    ;29453          D_ALU,                  ;
U 0E7A, 0803,403C,0180,0801,0280,C45F ;29454        J/M.CM.MG               ;  RETURN TO P0-SPACE FLOW
```

```
                                ;29455          ;DECODE ADDRESS TYPE; SYS, P0, OR P1 SPACE
                                ;29456
                                ;29457  =00111  ;VA<31-30>?(BEN/PSL. MODE)
                                ;29458
                                ;29459  M.TYPE:
                                ;29460          ;00111-----------------;  *REDUNDANT SYS SPACE ADDRESS*
                                ;29461          RC[VA.REF]_PC,          ;
                                ;29462          Q_ID[SLR],  D_DAL.SC,   ;
U 0447, 0D14,0038,F9F0,2DF0,0000,0E94   ;29463          J7M.SYS                 ;
                                ;29464
                                ;29465          ;01111-----------------;  *SYS SPACE ADDRESS*
                                ;29466          RC[VA.REF]_PC,          ;  SAVE VA, RIGHT ADJUST VPN,
                                ;29467          Q_ID[SLR],  D_DAL.SC,   ;  AND GET SLR INTO Q REG
U 044F, 0D14,0038,F9F0,2DF0,0000,0E94   ;29468          J7M.SYS                 ;
                                ;29469
                                ;29470          ;10111-----------------;  *P1 SPACE ADDRESS*
                                ;29471          RC[VA.REF]_PC,          ;  SAVE VIRTUAL ADDR OF REFERENCE
                                ;29472          Q_ID[P1LR],  D_DAL.SC,  ;  RIGHT ADJUST VPN, GET P1LR
U 0457, 0D14,0038,F5F0,2DF0,0000,0E7B   ;29473          J7M.P1                  ;
                                ;29474
                                ;29475  M.CM.MG:
                                ;29476          ;11111-----------------;  *P0 SPACE ADDRESS*
                                ;29477          RC[VA.REF]_PC,          ;  SAVE VA, RIGHT ADJUST VPN,
                                ;29478          Q_ID[P0LR],  D_DAL.SC,  ;  & GET P0LR INTO Q REG.
U 045F, 0D14,0038,F1F0,2DF0,0000,0E7D   ;29479          J7M.P0                  ;
                                ;29480  =;END
                                ;29481
                                ;29482          ;FETCH PTE FOR P1 VIRTUAL ADDRESS SPACE
                                ;29483          ;***************************************
                                ;29484
                                ;29485  M.P1:   ;---------------------;  SETUP TEST FOR LENGTH
                                ;29486          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E7B, 001D,2008,01C0,0800,0000,0E7C   ;29487          Q_Q-D-1                 ;  VIOLATION
                                ;29488
                                ;29489          ;---------------------;
                                ;29490          Q_ID[P1BR],  D_D.LEFT2, ;  SHIFT VPN LEFT 2 (DREG),
U 0E7C, 0100,0D3C,95F0,2C00,0000,0FA3   ;29491          J7M.PXLT,  Q31?         ;  LOAD P1BR INTO Q-REG,
                                ;29492                                  ;  & TEST FOR LENGTH VIOLATION
                                ;29493
                                ;29494  M.P0:   ;---------------------;  SETUP TEST FOR LENGTH
                                ;29495          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E7D, 001D,0000,01C0,0800,0000,0E80   ;29496          Q_D-Q                   ;  VIOLATION
                                ;29497
                                ;29498          ;---------------------;  SHIFT VPN LEFT 2,
                                ;29499          D_D.LEFT2,              ;  LOAD P0BR INTO Q REG,&
                                ;29500          Q_ID[P0BR],             ;  TEST FOR LENGTH VIOLATION
U 0E80, 0100,0D3C,91F0,2C00,0000,0FA3   ;29501          J7M.PXLT,  Q31?         ;
```

```
                                ;29502  =011    ;TEST FOR LENGTH VIOLATION (BEN/SIGNS(Q31))
                                ;29503
                                ;29504  M.PXLT: ;011--------------------;  *LENGTH VIOLATION FAULT*
                                ;29505          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29506          D_K[.9],                ;
U 0FA3, 0818,003A,D980,0800,0000,0002   ;29507          RETURN2                 ;
                                ;29508
                                ;29509
                                ;29510          ;111--------------------;  *NO LENGTH VIOLATION*
                                ;29511          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29512          D&VA_D+Q,               ;  CALCULATE PX PTE VIRTUAL ADDR.
U 0FA7, 081D,0014,0180,0800,3E00,0E81   ;29513          MSC/INH.CM.ADDR         ;
                                ;29514  =;END
                                ;29515
                                ;29516          ;----------------------;  ***SET-UP TEST OF PXPTE'S***
                                ;29517          RC[PTE.VA]_D,           ;  VIRTUAL ADDRESS & SAVE IT
                                ;29518          TEST.TB.RCHK,           ;
U 0E81, 0001,003C,0180,01D0,3C00,0E84   ;29519          MSC/INH.CM.ADDR         ;
                                ;29520
                                ;29521          ;----------------------;  ***TEST PXPTE VIRTUAL ADDRESS***
                                ;29522          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29523          MSC/INH.CM.ADDR,        ;  ALSO INHIBIT CM ADDRESS FORMAT
U 0E84, 0000,1D3C,0180,0800,3C00,025C   ;29524          TB.TEST?                ;
                                ;29525
                                ;29526  =11100  ;TEST FOR TB-MISS ONLY (BEN/TB.TEST)
                                ;29527
                                ;29528  M.PXTST: ;11100-----------------;  ***TBUF HIT***
                                ;29529          D[LONG]_CACHE.NOCHK,    ;
                                ;29530          MSC/INH.CM.ADDR,        ;  SPTE FOR PXPTE'S VIRTUAL
U 025C, 0000,003C,0180,4800,3C00,0E8A   ;29531          J/M.PXLD                ;  ADDRESS IS RESIDENT. READ PXPTE
                                ;29532
                                ;29533          ;11101-----------------;  **REDUNDANT, TBUF HIT***
                                ;29534          D[LONG]_CACHE.NOCHK,    ;
                                ;29535          MSC/INH.CM.ADDR,        ;
U 025D, 0000,003C,0180,4800,3C00,0E8A   ;29536          J/M.PXLD                ;
                                ;29537
                                ;29538          ;11110-----------------;  REDUNDANT, TBUF HIT
                                ;29539          D[LONG]_CACHE.NOCHK,    ;
                                ;29540          MSC/INH.CM.ADDR,        ;
U 025E, 0000,003C,0180,4800,3C00,0E8A   ;29541          J/M.PXLD                ;
                                ;29542
                                ;29543  ;INITIATE FETCH OF SPTE FOR PXPTE'S VIRTUAL ADDRESS
                                ;29544
                                ;29545          ;11111-----------------;  ***PXPTE NOT IN TBUF***
                                ;29546          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 025F, 0000,003C,0180,0801,0000,0E85   ;29547          PC_VA                   ;  PTE.VA TO BOTH PC & VA
                                ;29548  =;END
```

```
                                ;29549          ;----------------------;  START EXTRACTING VPN FROM PTE.VA
                                ;29550          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29551          D_PC.LEFT,  Q_0,        ;  CLEAR PTE.VA<31> & Q-REG
U 0E85, 0834,0038,DDF8,0800,0084,6B78   ;29552  SC_K[.FFF6]             ;  SET UP TO RIGHT ADJUST VPN
                                ;29553
                                ;29554  =00     ;00--------------------;
                                ;29555          Q_ID[SLR],  D_DAL.SC,   ;  SYS LENGTH REG TO Q & VPN TO D-REG
U 0B78, 0D00,003D,F9F0,2C00,0000,0E94   ;29556  CALL,  J/M.SYS          ;  RIGHT ADJUSTED;  MERGE WITH SYS PTE
                                ;29557                                  ;  FETCH ROUTINE.
                                ;29558
                                ;29559          ;01--------------------;  ***RETURN FROM SPTE FETCH ROUTINE***
                                ;29560          D[LONG]_CACHE.NOCHK,    ;
                                ;29561          MSC/INH.CM.ADDR,        ;  FETCH PXPTE
U 0B79, 0000,003C,0180,4800,3C00,0E8A   ;29562  J/M.PXLD                ;  MERGE BACK INTO PXPTE FETCH ROUTINE
                                ;29563
                                ;29564          ;10--------------------;  ***FAULT RETURN*******
                                ;29565          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B7A, 0819,0030,0980,0800,0000,0B7B   ;29566  D_D.OR.K[.2]             ;  UPDATE ERROR CODE TO INCLUDE
                                ;29567                                  ;  PXPTE SPT REFERENCE
                                ;29568
                                ;29569          ;----------------------;
                                ;29570          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0B7B, 0819,0034,6180,0800,0000,0E88   ;29571  D_D.AND.K[.F]            ;  PXPTE REFERENCE, RESET USER ACCESS CK
                                ;29572
                                ;29573          ;----------------------;
                                ;29574          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29575          PC&VA_RC[VA.REF],       ;  RELOAD PC AND VA TO ENTRY CONTENTS
U 0E88, 0010,003A,0180,0971,0200,0002   ;29576  RETURN2                 ;  FAULT RETURN TO CALLER
                                ;29577
                                ;29578  M.PXLD: ;----------------------;
                                ;29579          Q_D,                    ;  ORGINAL PTE TO Q-REG
                                ;29580          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E8A, 0811,0034,01E0,0978,0000,0E8C   ;29581  D_D.AND.RC[PTE.MASK]     ;  MASK OUT MBZ BITS
                                ;29582
                                ;29583          ;----------------------;  RESTORE VA REGISTER TO ORIGINAL
                                ;29584          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29585          PC&VA_RC[VA.REF],       ;  CONTENTS & TEST PXPTE FOR
U 0E8C, 0010,1D38,0180,0971,0200,04AF   ;29586  PTE.VALID?,  J/M.PXPTE   ;  VALIDITY, NOT NO-ACCESS,&
                                ;29587                                  ;  MZB FIELD.
                                ;29588
                                ;29589  =01111  ;TEST IF PTE IN D-REGISTER IS VALID (BEN/TB.TEST<NOT PTE VALID>)
                                ;29590
                                ;29591  M.PXPTE:
                                ;29592          ;01111-----------------;  *PXPTE IS OK*
                                ;29593          ID[TBUF]_D,  D_Q,       ;  LOAD PXPTE INTO TRANSLATION
                                ;29594          SC_RC[SC.SV],           ;  BUFFER AND RETURN TO
U 04AF, 0C10,003A,4180,3D68,0082,0001   ;29595  RETURN1                 ;  CALLING REOUTINE
                                ;29596
                                ;29597          ;11111-----------------;  *PXPTE NOT VALID FAULT*
                                ;29598          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 04BF, 0000,003C,8D80,0800,0084,6E8E   ;29599  SC_K[.1F]               ;  SETUP TO SET VALID BIT FOR TEST
                                ;29600  =;END
```

```
                                    ;29601          ;-----------------------;
                                    ;29602          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E8E, 0801,001C,0180,0800,0000,0E90 ;29603        D_D.ORNOT.MASK          ;  SET VALID BIT FOR TEST
                                    ;29604
                                    ;29605          ;-----------------------;  PROT AT LEAST KERNAL R/W IN PTE(NE 0+1)
U 0E90, 0000,003C,4180,3C00,0000,0E91 ;29606        ID[TBUF]_D              ;  WRITE BAD PTE INTO TBUF FOR ACCESS CHECK
                                    ;29607
                                    ;29608          ;-----------------------;
U 0E91, 0000,003C,0180,0000,0000,0E92 ;29609        TEST.TB.RCHK            ;  CHECK FOR READ ACCESS CHECK
                                    ;29610
                                    ;29611          ;-----------------------;
                                    ;29612          D_0,                    ;
                                    ;29613          TEST.TB.WCHK,           ;  SETUP FOR WRITE ACESS CHECK
U 0E92, 0F00,1D3C,0180,1000,0000,029D ;29614        TB.TEST?                ;  BRANCH ON READ ACCESS TEST
                                    ;29615
                                    ;29616  =11101  ;BRANCH ON TRANS TEST<TB MISS + ACCESS VIOL>
                                    ;29617
                                    ;29618          ;11101----------------;  NO READ ACCESS VIOLATION
                                    ;29619          ID[TBUF]_D,             ;  CLEAR TBUF LOCATION
U 029D, 0000,1D3C,4180,3C00,0000,02DD ;29620        TB.TEST?, J/M.ACHK.1    ;  TEST FOR WRITE VIOL
                                    ;29621
                                    ;29622          ;11111----------------;
U 029F, 0000,003C,4180,3C00,0000,0E93 ;29623        ID[TBUF]_D              ;  CLEAR TBUF LOCATION
                                    ;29624
                                    ;29625          ;-----------------------;
                                    ;29626          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29627          D_K[.30],               ;  NOTE BOTH READ AND WRITE VIOL POSSIBLE
U 0E93, 0818,003A,7980,0800,0000,0002 ;29628        RETURN2                 ;  RETURN TO CALLER
                                    ;29629
                                    ;29630
                                    ;29631
                                    ;29632
                                    ;29633
                                    ;29634          ;FETCH PTE FOR SYSTEM VIRTUAL ADDRESS SPACE
                                    ;29635          ;*******************************************
                                    ;29636
                                    ;29637  M.SYS:  ;-----------------------;
                                    ;29638          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 0E94, 001D,0000,01C0,0800,0000,0E96 ;29639        Q_D-Q                   ;  SET-UP FOR LENGTH VIOLATION
                                    ;29640                                  ;  TEST
                                    ;29641
                                    ;29642          ;-----------------------;
                                    ;29643          Q_ID[SBR],              ;  SHIFT VPN LEFT 2,
                                    ;29644          D_D.LEFT2,              ;  LOAD SBR INTO Q-REG,
U 0E96, 0100,0D3C,99F0,2C00,0000,0FAB ;29645        J/M.SYSLT, Q31?         ;  TEST FOR LENGTH VIOLATION
```

```
                                    ;29646  =011    ;TEST FOR LENGTH VIOLATION (BEN/SIGNS(Q31))
                                    ;29647
                                    ;29648  M.SYSLT: ;011-------------------;  *LENGTH VIOLATION*
                                    ;29649           MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29650           D_K[.9],               ;
U 0FAB, 0818,003A,D980,0800,0000,0002 ;29651         RETURN2                ;
                                    ;29652                                  ;
                                    ;29653
                                    ;29654           ;111-------------------;  *PASSED LENGTH TEST*
                                    ;29655           MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29656           RC[PTE.PA]&VA_D+Q,     ;  CALCULATE SPTE PHYSICAL ADDRESS
                                    ;29657           MSC/INH.CM.ADDR,       ;  AND LOAD INTO VA & RC<PTE.PA>
U 0FAF, 001D,0014,0180,09D8,3E00,0E98 ;29658         J/SYS.NXT              ;
                                    ;29659  =;END
                                    ;29660
                                    ;29661  SYS.NXT: ;----------------------;
                                    ;29662           D[LONG]_CACHE.P,       ;
U 0E98, 0000,003C,0180,C800,3C00,0E99 ;29663         MSC/INH.CM.ADDR        ;  FETCH SPTE INTO D-REGISTER
                                    ;29664                                  ;  & INHIBIT CM ADDRESS FORMAT
                                    ;29665
                                    ;29666           ;----------------------;
                                    ;29667           Q_D,                   ;  ORIGINAL PTE TO Q-REG
                                    ;29668           MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29669           D_D.AND.RC[PTE.MASK],  ;  MASK OUT MBZ BITS
U 0E99, 0811,0034,01E0,0978,3C00,0E9A ;29670         MSC/INH.CM.ADDR        ;
                                    ;29671
                                    ;29672           ;----------------------;
                                    ;29673           MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29674           VA_PC,  MSC/INH.CM.ADDR,;  RESTORE SVA INTO VA &
U 0E9A, 0014,1D38,0180,0800,3E00,04CF ;29675         J/M.SPTE,  PTE.VALID?  ;  TEST SPTE FOR VALIDITY,
                                    ;29676                                  ;  NOT NO ACCESS, AND MBZ FIELD.
                                    ;29677
                                    ;29678  =01111  ;TEST IF SPTE IN D-REGISTER IS VALID (BEN/TB.TEST<NOT PTE.VALID>)
                                    ;29679
                                    ;29680  M.SPTE:
                                    ;29681           ;01111-----------------;  *SPTE VALID*
                                    ;29682           ID[TBUF]_D,  D_Q,      ;
                                    ;29683           MSC/INH.CM.ADDR,       ;  LOAD SPTE INTO TBUF
                                    ;29684           SC_RC[SC.SV],          ;
U 04CF, 0C10,003A,4180,3D68,3C82,0001 ;29685         RETURN1                ;
                                    ;29686
                                    ;29687           ;11111-----------------;  *SPTE INVALID*
                                    ;29688           MCT/MEM.NOP,           ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 04DF, 0000,0D3C,0180,0800,0000,0FB6 ;29689         D31?                   ;
                                    ;29690  =;END
```

```
                                ;29691  =110    ;CHECK TO SEE IF PXPTE IS VALID (BEN/SIGNS(D31))
                                ;29692
                                ;29693          ;110--------------------;  ***PTE NOT VALID***
                                ;29694          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29695          SC_K[.4],       D_0,    ;  START SETUP FOR TEST OF AT LEAST 'K R/W'
                                ;29696          Q_D,                    ;  PTE TO Q-REG
U OFB6, OF00,003C,11E0,0800,0084,6E9B   ;29697          J7M.ACHKS               ;
                                ;29698
                                ;29699          ;111--------------------;  ***ACCESS VIOLATION ON PTE***
                                ;29700          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29701          D_K[.8],                ;  D(03) SET TO NOTE ACCESS
U OFB7, 0818,003A,0180,0800,0000,0002   ;29702          RETURN2                 ;  VIOLATION FOR EXCEPTION ROUTINE
                                ;29703  =;END
                                ;29704
                                ;29705  M.ACHKS:
                                ;29706          ;------------------------;
                                ;29707          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29708          SC_K[.1F],              ;  SETUP TO SET VALID BIT FOR TEST
U OE9B, 0D00,003C,8D80,0800,0084,6E9C   ;29709          D_BAL.SC                ;  SHIFT IN PROT<03:01> FOR KERN R/W CHK
                                ;29710
                                ;29711          ;------------------------;
                                ;29712          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29713          D_Q.ORNOT.MASK,         ;  SET VALID BIT FOR TEST
                                ;29714          MSC/INH.CM.ADDR,        ;
U OE9C, 0801,2D1C,0180,0800,3C00,0FBD   ;29715          D.NE.0?                 ;  TEST FOR KERN R/W ACCESS
                                ;29716
                                ;29717  =101    ;BRANCH ON DREG NOT EQUAL TO ZERO
                                ;29718
                                ;29719          ;101--------------------;
                                ;29720          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                ;29721          D_K[.8],                ;  FLAG AN ACCESS VIOLATION
U OFBD, 0818,003A,0180,0800,0000,0002   ;29722          RETURN2                 ;  RETURN TO CALLER
                                ;29723
                                ;29724          ;111--------------------;  PROT AT LEAST KERNAL R/W IN PTE(NE 0+1)
                                ;29725          ID[TBUF]_D,             ;  WRITE BAD PTE INTO TBUF FOR ACCESS CHECK
U OFBF, 0000,003C,4180,3C00,3C00,0E9D   ;29726          MSC/INH.CM.ADDR         ;
                                ;29727
                                ;29728          ;------------------------;
                                ;29729          TEST.TB.RCHK,           ;  CHECK FOR READ ACCESS CHECK
U OE9D, 0000,003C,0180,0000,3C00,0EA0   ;29730          MSC/INH.CM.ADDR         ;
                                ;29731
                                ;29732          ;------------------------;
                                ;29733          D_0,                    ;
                                ;29734          MSC/INH.CM.ADDR,        ;
                                ;29735          TEST.TB.WCHK,           ;  SETUP FOR WRITE ACESS CHECK
U OEA0, OF00,1D3C,0180,1000,3C00,02BD   ;29736          TB.TEST?                ;  BRANCH ON READ ACCESS TEST
```

```
                                    ;29737  =11101  ;BRANCH ON TRANS TEST<TB MISS + ACCESS VIOL>
                                    ;29738
                                    ;29739          ;11101-----------------;  NO READ ACCESS VIOLATION
                                    ;29740          ID[TBUF] D,             ;  CLEAR TBUF LOCATION
                                    ;29741          MSC/INH.CM.ADDR,        ;
U 02BD, 0000,1D3C,4180,3C00,3C00,02DD ;29742        TB.TEST?,  J/M.ACHK.1   ;  TEST FOR WRITE VIOL
                                    ;29743
                                    ;29744          ;11111-----------------;
                                    ;29745          ID[TBUF] D,             ;  CLEAR TBUF LOCATION
U 02BF, 0000,003C,4180,3C00,3C00,0EA1 ;29746        MSC/INH.CM.ADDR         ;
                                    ;29747
                                    ;29748          ;-----------------------;
                                    ;29749          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29750          D_K[.30],               ;  NOTE BOTH READ AND WRITE VIOL POSSIBLE
U 0EA1, 0818,003A,7980,0800,0000,0002 ;29751        RETURN2                 ;  RETURN TO CALLER
                                    ;29752
                                    ;29753  =11101  ;BRANCH ON TRANS TEST<TB MISS + ACCESS VIOL>
                                    ;29754
                                    ;29755  M.ACHK.1:
                                    ;29756          ;11101-----------------;  NO WRITE ACCESS VIOLATION
                                    ;29757          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
U 02DD, 0000,003E,0180,0800,0000,0002 ;29758        RETURN2                 ;  RETURN TO CALLER
                                    ;29759
                                    ;29760          ;11111-----------------;  WRITE ACCESS VIOL  POSSIBLE
                                    ;29761          MCT/MEM.NOP,            ;  !!!! DONOT ALLOW IPA RE-FILL CYCLES !!!!
                                    ;29762          D_K[.20],               ;  NOTE ACCESS VIOL POSSIBLE
U 02DF, 0818,003A,7580,0800,0000,0002 ;29763        RETURN2                 ;  RETURN TO CALLER
```

```
;29764  .TOC     ""      Memory management      : MEMORY  ANAGEMENT FAULT HANDLING ROUTINE''
;29765
;29766          ;*****************************************
;29767          ;                                        *
;29768          ;MEMORY MANAGEMENT FAULT HANDLING ROUTINE *
;29769          ;                                        *
;29770          ;*****************************************
;29771
;29772          ;THIS ROUTINE IS USED PROCESS PROTECTION VIOLATIONS, LENGTH VIOLATIONS,
;29773          ;AND PTE NOT VALID EXCEPTIONS (SEE VAX SRM CHAPTERS 5&6 FOR
;29774          ;DETAILS)
;29775
;29776          ;ROUTINE ENTRIES & INPUT REQUIREMENTS
;29777          ;------------------------------------
;29778
;29779          ;       1) ''J/M.FLT, [LAST.REF]?'' , (NORMAL ENTRY)
;29780
;29781          ;       THIS ENTRY IS USED ONLY BY MEMORY MANAGEMENT
;29782          ;       ROUTINES AND EXPECTS D,Q,SC,LC,PC,VA,&VA.REF
;29783          ;       TO BE SAVED IN EITHER RC OR ID REGISTERS.  EXCEPTION TYPE
;29784          ;       NOTED IN D_REG (D<0>=LENGTH CHECK, D<1>=PTE FAULT,
;29785          ;        D<3>=ACCESS VIOL).  ALSO SET NESTED ERROR FLAG
;29786
;29787          ;       2) ''J/M.FLT.LC'' , (LC NOT SAVED ENTRY)
;29788
;29789          ;       SAME AS 1) ABOVE EXCEPT LC HAS NOT BEEN SAVED
;29790
;29791          ;       3) ''J/M.FLT.PRB''
;29792
;29793          ;       SAME AS 1) ABOVE EXCEPT TYPE OF REFERENCE NOTED
;29794          ;       IN D_REG <2> (D<2>=WRITE OR MODIFY INTENT)
;29795
;29796          ;       4) ''J/FPD.RTN'', (FIRST PART DONE ENTRY)
;29797
;29798          ;       THIS ENTRY IS USED BY FIRST PART DONE SETUP ROUTINES AFTER
;29799          ;       THEY HAVE SET-UP TO TAKE THE MEMORY MANAGEMENT
;29800          ;       FAULT.  THIS ENTRY EXPECTS THAT RC[VA.REF] AND
;29801          ;       RC[PTE.VA] ARE TO BE UNCHANGED BY THE FPD SETUP
;29802          ;       ROUTINE.  ON ENTRY TO THIS POINT, ALL INFO NEEDED TO
;29803          ;       RESTART THE FAULTED INSTRUCTION MUST HAVE BEEN
;29804          ;       SAVED IN THE GENERAL REGISTER SET AND THE PC BACKED-UP.
;29805
;29806          ;       5) ''RETURN1'', (RETURN FROM EXCEPTION SET-UP ROUTINE
;29807          ;                        CALLED BY THIS ROUTINE)
;29808
;29809          ;       THE EXCEPTION SET-UP ROUTINE WILL RETURN TO THIS ROUTINE
;29810          ;       TO PUSH THE FAULT PARAMETERS ONTO THE
;29811          ;       STACK.  THE EXCEPTION SET-UP ROUTINE IS EXPECTED
;29812          ;       TO HAVE SET-UP THE PROPER STACK POINTER IN R14 FOR
;29813          ;       THIS ROUTINE TO USE.  THE EXCEPTION SETUP ROUTINE
;29814          ;       MUST NOT CHANGE RC [VA.REF] AND RC[PTE.VA].
```

```
                                        ;29815          ;ROUTINE EXITS
                                        ;29816          ;-----------------------
                                        ;29817
                                        ;29818          ;       1) 'RETURN 1 (ID[FPDA] ON TOP OF USTACK)',
                                        ;29819
                                        ;29820          ;       THIS ROUTINE EXITS TO THE SPECIFIED FPD SETUP ROUTINE
                                        ;29821          ;       ON WRITE REFERENCES IF PSL(FPD)=1.   WORKING REGISTERS ARE
                                        ;29822          ;       AS THEY WERE WHEN THE MEMORY REFERENCE WAS
                                        ;29823          ;       TRAPPED.  USTACK IS NOT RESTORED TO MICRO ADDRESS SAVED
                                        ;29824          ;       ON THE MICRO-TRAP.
                                        ;29825
                                        ;29826          ;       2) 'RETURN 2 (ID[FPDA] ON TOP OF USTACK)''
                                        ;29827
                                        ;29828          ;       SAME AS 1) ABOVE EXCEPT USED ON READ REFERENCES
                                        ;29829
                                        ;29830          ;       3) ''CALL, J/EXCPTN'' (CALL OF EXCEPTION SET-UP ROUTINE)
                                        ;29831
                                        ;29832          ;       THIS CALL IS USED TO ACTIVATE THE EXCEPTION SETUP
                                        ;29833          ;       ROUTINE TO SET-UP THE EXCEPTION, AND AFTER
                                        ;29834          ;       COMPLETING SAME TO RETURN TO THIS ROUTINE.  SEE
                                        ;29835          ;       5)UNDER RETURN ENTRIES ABOVE FOR EXPECTED STATE
                                        ;29836          ;       ON RETURN TO THIS ROUTINE.
                                        ;29837          ;       THIS ROUTINE LEAVES THE PROPER EXCEPTION
                                        ;29838          ;       VECTOR IN THE D_REGISTER (20 OR 24).
                                        ;29839
                                        ;29840          ;       4)  J/EH.FAULT  IF EFP FLAG IS SET
                                        ;29841
                                        ;29842          ;       RETURN TO ERROR HANDLING ROUTINE WHICH CALLED THIS
                                        ;29843          ;       ROUTINE TO PROCESS AN ERROR FAULT.
                                        ;29844
                                        ;29845
                                        ;29846  M.FLT.LC: ;--------------------;
                                        ;29847          RC[LC.SV]_LC,           ;  SAVE ENTRY CONTENTS OF LC RG
U 0EA2, 0010,1138,0180,F9C0,0000,0D7C _ ;29848          LAST.REF?               ;  WAS LAST REFERENCE A WRITE/MODIFY?
                                        ;29849
                                        ;29850  =1100   ;BRANCH ON LAST TRAPPED MEMORY REFERENCE
                                        ;29851
                                        ;29852  M.FLT:
                                        ;29853          ;1100-----------------;  *** READ/RD CHECK ***
                                        ;29854          Q_D.AND.K[.10].RIGHT,   ;  OR IN RD ACCESS FLAG
U 0D7C, 0059,0034,65C0,F800,0000,0EA8   ;29855          J7M.FLT.B               ;
                                        ;29856
                                        ;29857          ;1101-----------------;  ** INTERLOCKED READ **
                                        ;29858          D_D.OR.K[.4],           ;  SET MODIFY INTENT BIT IN PARAMETER
U 0D7D, 0819,0030,1180,F800,0000,0EA3   ;29859          J7M.FLT.A               ;
                                        ;29860
                                        ;29861          ;1110-----------------;  *** READ/WRITE CHECK ***
                                        ;29862          D_D.OR.K[.4],           ;  SET MODIFY INTENT BIT IN PARAMETER
U 0D7E, 0819,0030,1180,F800,0000,0EA3   ;29863          J7M.FLT.A               ;
```

```
                                    ;29864          ;1111-------------------;  *** WRITE REFERENCE ***
                                    ;29865          D_D.OR.K[.4],           ;  SET MODIFY INTENT BIT IN PARAMETER
U 0D7F, 0819,0030,1180,F800,0000,0EA3 ;29866        J/M.FLT.A               ;
                                    ;29867
                                    ;29868
                                    ;29869
                                    ;29870  M.FLT.A:;-----------------------;
U 0EA3, 0099,0034,75C0,F800,0000,0EA8 ;29871        Q_D.AND.K[.20].RIGHT2   ;  OR IN WR ACCESS BIT
                                    ;29872
                                    ;29873  M.FLT.B:;-----------------------;
                                    ;29874          D_ALU,                  ;  PARA TO D-REG FOR CONSOLE
U 0EA8, 081D,0030,0180,F9D0,0000,0EA9 ;29875        RL[PTE.VA]_D.OR.Q       ;  SAVE EXCEPTION PARA.
                                    ;29876
                                    ;29877
                                    ;29878  M.FLT.PRB: ;-------------------;  ***SPECIAL ENTRY FOR PRB ***
                                    ;29879          SC_K[.8],               ;  SETUP TO SHIFT FAULT PARA LEFT 8
                                    ;29880          Q_0,                    ;  CLEAR OUT Q FOR SHIFT
U 0EA9, 0000,1C3C,0078,F800,0084,627B ;29881        CONSOLE.MODE?           ;  CONSOLE MODE ?
                                    ;29882
                                    ;29883  =11011  ;TEST FOR FAULT CAUSED BY CONSOLE MODE (BEN/PSL.MODE)
                                    ;29884
                                    ;29885          ;11011------------------;  *** EXIT-TO-CONSOLE CODE ***
                                    ;29886          D_DAL.SC,               ;  SHIFT FAULT PARA LEFT 8
U 027B, 0D00,003C,0180,F800,0000,0EAC ;29887        J/M.FLT.01              ;
                                    ;29888
                                    ;29889          ;11111------------------;  *** NOT CONSOLE MODE ***
                                    ;29890          Q_ID[FPDA],             ;  LOAD ADDRESS OF FPD SET-UP ROUTINE
U 027F, 0000,003C,B5F0,2C00,0000,0918 ;29891        J/M.FLT.02              ;
                                    ;29892
                                    ;29893  M.FLT.01:;-----------------------;
                                    ;29894          D_D.OR.K[.1],           ;  SET ERROR CODE FOR MEMORY MANAGEMENT FAULT
U 0EAC, 0819,0030,0580,F800,0000,0F1B ;29895        J/M.CNSL                ;  GO TO CONSOLE MICRO-CODE
                                    ;29896
                                    ;29897  =0*     ;SETUP CALL CONSTRAINT BLOCK
                                    ;29898
                                    ;29899  M.FLT.02:;0*--------------------;
                                    ;29900          D_Q,                    ;  FPD SETUP ROUTINE ADDRESS TO D_REG
U 0918, 0C00,113D,0180,F800,0000,0D87 ;29901        CALL,  FPD?, J/M.FLT.2A ;  IS PSL(FPD) SET ?
                                    ;29902
                                    ;29903          ;1*--------------------; * RETURN FROM FPD-SET-UP & BACK UP REGS *
                                    ;29904          Q_ID[PSL],              ;  GET PSL TO RESET TP BIT
U 091A, 0000,003C,3DF0,2C00,0000,0EBC ;29905        J/M.FLT.08              ;
                                    ;29906  =;END
```

```
                                    ;29907  =0111   ;TEST FOR PSL(FPD) SET (BEN/LAST.REF)
                                    ;29908
                                    ;29909  M.FLT.2A:
                                    ;29910          ;0111--------------------;  *** PSL (FPD) SET ***
                                    ;29911          ID[USTACK]_D,           ;  PUT FPD SETUP ROUTINES ADDR ON STACK
U 0D87, 0000,003C,8180,3C00,0000,0EAD ;29912        J/M.FLT.04              ;
                                    ;29913  M.FLT.03:
                                    ;29914          ;1111--------------------;  *** PSL (FPD) NOT SET ***
U 0D8F, 0000,153C,0180,F800,0000,0DA7 ;29915        RLOG.EMPTY?,J/BAKUP.RGS ;  GO BACK-UP PC & GENERAL REGS
                                    ;29916
                                    ;29917
                                    ;29918  M.FLT.04:;----------------------;
U 0EAD, 0010,0038,0180,F961,0200,0EB0 ;29919        PC&VA_RC[PC.SV]         ;  RESTORE PC TO ENTRY CONTENTS AND LEAVE
                                    ;29920                                  ;  NESTED ERR 'ON'
                                    ;29921
                                    ;29922
                                    ;29923          ;----------------------;
U 0EB0, 0010,0038,0180,F948,0200,0EB1 ;29924        VA_RC[VA.SV]            ;  TRAPPED VIRTUAL ADDR TO VA REG
                                    ;29925
                                    ;29926          ;----------------------;
                                    ;29927          Q_ID[D.SV],             ;  ENTRY CONTENTS OF D TO Q
                                    ;29928          SC_RC[SC.SV],           ;  RESTORE ENTRY CONTENTS OF SC
U 0EB1, 0010,1138,B9F0,2D68,0082,0D9C ;29929        LAST.REF?               ;  WAS TRAPPED REFERENCE A WRITE?
                                    ;29930
                                    ;29931
                                    ;29932  =1100   ;1100--------------------;  *** READ REFERENCE ***
                                    ;29933          D_Q,                    ;  RESTORE D_REG TO ENTRY CONTENTS
                                    ;29934          Q_ID[Q.SV],             ;  RESTORE Q_REG
                                    ;29935          LC_RC[LC.SV],           ;  RESTORE LC_REG
U 0D9C, 0C00,003E,BDF0,2D40,0000,0002 ;29936        RETURN2                 ;  READ ENTRY INTO FPD ROUTINE
                                    ;29937
                                    ;29938          ;1101--------------------;  *** READ REFERENCE ***
                                    ;29939          D_Q,                    ;  (REDUNDANT)
                                    ;29940          Q_ID[Q.SV],             ;
                                    ;29941          LC_RC[LC.SV],           ;
U 0D9D, 0C00,003E,BDF0,2D40,0000,0002 ;29942        RETURN2                 ;
                                    ;29943
                                    ;29944          ;1110--------------------;  *** READ REFERENCE ***
                                    ;29945          D_Q,                    ;  (REDUNDANT)
                                    ;29946          Q_ID[Q.SV],             ;
                                    ;29947          LC_RC[LC.SV],           ;
U 0D9E, 0C00,003E,BDF0,2D40,0000,0002 ;29948        RETURN2                 ;
                                    ;29949
                                    ;29950          ;1111--------------------;  *** WRITE REFERENCE ***
                                    ;29951          D_Q,                    ;  RESTORE D_REG TO ENTRY CONTENTS
                                    ;29952          Q_ID[Q.SV],             ;  RESTORE Q_REG
                                    ;29953          LC_RC[LC.SV],           ;  RESTORE LC_REG
U 0D9F, 0C00,003E,BDF0,2D40,0000,0001 ;29954        RETURN1                 ;  WRITE ENTRY INTO FPD ROUTINE
                                    ;29955
                                    ;29956  =;END
```

```
;29957  .TOC    ""      Memory management     : BACK-UP GENERAL REGS AND PC''
;29958
;29959          ;DESCRIPTION:
;29960          ;-----------
;29961          ;THIS ROUTINE IS USED TO BACK-UP THE GENERAL REGISTERS AND
;29962          ;PC ON MEMORY MANAGEMENT, SPECIFIER, OPERAND,  AND ERROR FAULTS.
;29963
;29964
;29965          ;*********************************************************
;29966          ;                                                       *
;29967          ;                     NOTE                              *
;29968          ;                  ----------                           *
;29969          ;       THIS ROUTINE SHOULD NEVER BE PASSED THRU        *
;29970          ;       TWICE.  RLOG IS WRAPPED AROUND TO 'F'           *
;29971          ;       WHICH WILL CAUSE UNPREDICTABLE BACKING-UP       *
;29972          ;       OF THE GENERAL REGISTERS.                       *
;29973          ;                                                       *
;29974          ;       ALSO THE BACKING-UP OF THE PC MUST OCCUR        *
;29975          ;       AFTER BACKING-UP OF THE REGISTERS BECAUSE:      *
;29976          ;       IF RLOG IS EMPTY,THE ABOVE PROBLEM  MAY OCCUR.  *
;29977          ;       IF IT IS NOT EMPTY, A BACK-UP VALUE WILL BE LOST.
;29978          ;                                                       *
;29979          ;*********************************************************
;29980
;29981          ;CALLING SEQUENCES:
;29982          ;------------------
;29983          ; 1)  ''CALL, RLOG.EMPTY?, J/BAKUP.REGS''
;29984          ;     THIS ENTRY USED TO BACK-UP GENERAL REGS AND PC
;29985
;29986          ; 2)  ''CALL, J/BAKUP.PC''
;29987          ;     THIS ENTRY IS USED WHEN ONLY THE PC IS TO BE
;29988          ;     BACKED UP.  THE CURRENT VALUE OF THE PC MUST BE IN Q-REG.
;29989
;29990          ;INPUT REQUIREMENTS:
;29991          ;-------------------
;29992          ;CONTENTS OF SC, D, Q, LAB REGISTERS ARE DESTROYED BY THIS ROUTINE.
;29993          ;ALSO ON ENTRY, 2) ABOVE CURRENT VALUE OF PC MUST BE IN Q-REG.
;29994
;29995          ;OUTPUTS:  PC-DELTA LEFT IN D-REG (ZERO-EXTENDED AND RIGHT ADJUSTED)
;29996          ;-------
;29997
;29998          ;RETURN TO CALLER:      'RETURN2''
;29999          ;----------------
```

```
                                    ;30000  =0111   ;TEST RLOG STACK TO SEE IF IT IS EMPTY
                                    ;30001
                                    ;30002  BAKUP.RGS::0111----------------;  *** BACKUP REG NOTED BY RLOG ***
                                    ;30003          SC_K[.FFF8],           ;  SET UP FOR D-REG SHIFT RIGHT OF 8
                                    ;30004          D_RLOG,                ;  RLOG & PCSV TO D_REG
U 0DA7, 0800,0038,7180,F800,1C84,6EB2 ;30005        J7M.FLT.06             ;
                                    ;30006
                                    ;30007          ;1111----------------;  *** RLOG STACK EMPTY ***
                                    ;30008          Q_RC[PC.SV],           ;  CURRENT PC VALUE TO Q_REG &
U 0DAF, 0010,0038,01C0,F960,0000,0EB8 ;30009        J7M.FLT.07             ;  GO TO CODE TO BACK-UP THE PC
                                    ;30010  =:END
                                    ;30011
                                    ;30012
                                    ;30013  M.FLT.06:;-------------------;  *** BACKUP REG NOTED BY RLOG ***
                                    ;30014          Q_D.RIGHT,             ;  ALIGN ADD/SUB BIT INTO Q<15>
U 0EB2, 0D41,003C,01C0,F800,0000,0EB4 ;30015        D_DAL.SC               ;  SHIFT PCSV OUT OF D_REG
                                    ;30016
                                    ;30017
                                    ;30018          ;-------------------;
                                    ;30019          Q_Q.SXT[WORD],         ;  SIGN EXTEND Q<15> TO Q<31> TO TEST
U 0EB4, 0002,603C,F1C0,F800,0084,6EB5 ;30020        SC_K[.FFFC]            ;  SETUP FOR D_REG RT 4 SHIFT
                                    ;30021
                                    ;30022          ;-------------------;
                                    ;30023          SC_D,                  ;  RA ADDRESS TO SC
U 0EB5, 0D01,003C,0180,F800,0082,0EB6 ;30024        D_DAL.SC               ;  SHIFT OUT RA ADDRESS
                                    ;30025
                                    ;30026          ;-----------------------------;
                                    ;30027          D_D.OXT[BYTE].ANDNOT.K[.FFF0], ;  SETUP VALUE TO MODIFY RA (SC)
                                    ;30028          LAB_R(SC),             ;  RA(SC) TO LA REG
U 0EB6, 081B,8D24,6D80,F868,0000,0FC3 ;30029        Q31?                   ;  ADD VALUE TO RA (SC) ?
                                    ;30030
                                    ;30031
                                    ;30032  =011    ;IS VALUE TO BE ADDED TO LA ? (BEN/SIGNS (Q31))
                                    ;30033
                                    ;30034          ;011----------------;  ***  YES, ADD IT IN ***
                                    ;30035          R(SC) LA+D,            ;
U 0FC3, 001C,3514,0180,F8E8,0000,0DA7 ;30036        RLOG.EMPTY?,J/BAKUP.RGS ;  GO SEE IF THERE ARE MORE TO BACK-UP
                                    ;30037
                                    ;30038          ;111----------------;  *** NO, SUBTRACT IT ***
                                    ;30039          R(SC) LA-D,            ;
U 0FC7, 001C,3500,0180,F8E8,0000,0DA7 ;30040        RLOG.EMPTY?,J/BAKUP.RGS ;  GO SEE IF THERE ARE MORE TO BACK-UP
                                    ;30041
                                    ;30042  BAKUP.PC:
                                    ;30043  M.FLT.07::-------------------;  *** CODE TO BACK-UP PC ***
U 0EB8, 0801,2000,0180,F800,1C00,0EB9 ;30044        D_Q-PCSV               ;  GET AMOUNT TO BACK-UP PC
                                    ;30045
                                    ;30046          ;-------------------;
U 0EB9, 0803,803C,0180,F800,0000,0EBA ;30047        D_D.OXT[BYTE]          ;  ZERO EXTEND PC DELTA
                                    ;30048
                                    ;30049          ;-------------------;
                                    ;30050          PC&VA_Q-D,             ;  BACK-UP PC TO INSTR'S OP-CODE
U 0EBA, 001D,2002,0180,F801,0200,0002 ;30051        RETURN2                ;
```

```
                                    ;30052  FPD.RTN:;-----------------------;
U 0EBB, 0000,003E,0180,F800,0000,0002   ;30053          RETURN2               ;
                                    ;30054
                                    ;30055  M.FLT.08:;----------------------;
U 0EBC, 0000,003C,5180,F800,0084,6EBE   ;30056          SC_K[.1E]             ;  SETUP MASK TO RESET TP BIT
                                    ;30057
                                    ;30058          ;----------------------;
U 0EBE, 0801,2034,0180,F800,0000,0EC0   ;30059          D_Q.AND.MASK          ;  RESET TP BIT
                                    ;30060
                                    ;30061          ;----------------------;
U 0EC0, 0000,003C,3D80,3C00,0000,0EC1   ;30062          ID[PSL]_D             ;  PSL UPDATE
                                    ;30063
                                    ;30064          ;----------------------;
U 0EC1, 0000,003C,6DF0,2C00,0000,0EC2   ;30065          Q_ID[FAULT]           ;  GET SBI FAULT REGISTER
                                    ;30066
                                    ;30067          ;----------------------;
U 0EC2, 0000,003C,B980,F800,0084,6EC3   ;30068          SC_K[.19]             ;  SETUP MASK BIT TO TEST 'EFP'
                                    ;30069
                                    ;30070          ;----------------------;
                                    ;30071          ALU_Q.ANDNOT.MASK,    ;  TEST 'EFP' BIT
U 0EC3, 0001,2024,0180,F800,0010,0EC8   ;30072          CLK.UBCC              ;
                                    ;30073
                                    ;30074          ;----------------------;
                                    ;30075          D_RC[PTE.VA],         ;  2ND PARA. TO D-REG
U 0EC8, 0810,0138,0180,F950,0000,08E8   ;30076          Z?                    ;  WAS 'EFP' SET?
                                    ;30077
                                    ;30078  =0       ;BRANCH ON ALU.Z(BEN/Z)
                                    ;30079          ;0----------------------;
U 08E8, 0000,003C,0180,F800,0000,091E   ;30080          J/EH.FAULT            ;  ***  YES, EXIT TO ERROR HANDLING  ***
                                    ;30081
                                    ;30082          ;1----------------------;
                                    ;30083          RC[T7]_D.AND.K[.7],   ;  RESET ACCESS VIOL FLAG
U 08E9, 0019,1934,5D80,F988,0000,0DB7   ;30084          D3?                   ;  WAS IT SET ?
                                    ;30085
                                    ;30086
                                    ;30087
                                    ;30088  =0111    ;CHECK TO SEE IF ACCESS VIOL FLAG IS SET (BEN/D3-0)
                                    ;30089
                                    ;30090          ;0111-------------------;  *** NO, PTE NOT VALID ***
                                    ;30091          RC[T0]_K[.24],        ;  EXCEPTION VECTOR TO T0
U 0DB7, 0018,0038,E980,F980,0000,0EC9   ;30092          J/M.FLT.09            ;
                                    ;30093
                                    ;30094          ;1111-------------------;  *** YES, ACCESS VIOLATION ***
U 0DBF, 0018,0038,7580,F980,0000,0EC9   ;30095          RC[T0]_K[.20]         ;  EXCEPTION VECTOR TO T0
                                    ;30096
                                    ;30097
                                    ;30098  M.FLT.09:
                                    ;30099          ;----------------------;
U 0EC9, 0810,0038,0180,F970,0000,08EC   ;30100          D_RC[VA.REF]          ;  GET FAULTING VIRTUAL ADDRESS
```

```
                                    ;30101  =0      ;CALL CONSTRANT BLOCK
                                    ;30102
                                    ;30103          ;0--------------------;  *** CALL EXCEPTION SET-UP RNT ***
                                    ;30104          R[T6]_D,               ;  SAVE FAULTING VA IN 'T6'
U 08EC, 0001,003D,0180,F9B0,0000,0DE5 ;30105        CALL, J/EXCPTN         ;
                                    ;30106
                                    ;30107          ;1--------------------;  *** RETURN ***(RA[SP] IN LA)
U 08ED, 0810,0038,0180,F930,0000,0ECA ;30108        D_R[T6]                ;  GET FAULTING VIRTUAL ADDR.
                                    ;30109  =;END
                                    ;30110
                                    ;30111          ;----------------------;
U 0ECA, 0018,0000,1180,FAF0,0200,0ECB ;30112        R[SP]&VA_LA-K[.4]      ;  DECREMENT STACK POINTER
                                    ;30113
                                    ;30114          ;----------------------;
                                    ;30115          CACHE_D[LONG],         ;  PUSH FAULTED VIRTUAL ADDR
U 0ECB, 0000,003C,0180,3270,0000,0ECC ;30116        LAB_R[SP]              ;
                                    ;30117
                                    ;30118          ;----------------------;
U 0ECC, 0810,0038,0180,F938,0000,0ECE ;30119        D_R[T7]                ;  GET 2ND PARAMETER
                                    ;30120
                                    ;30121          ;----------------------;
                                    ;30122          R[SP]&VA_LA-K[.4],     ;  DECREMENT STACK POINTER
U 0ECE, 0018,0000,1180,FAF0,0200,002A ;30123        J/EXCPT2               ;  JMP BACK TO I&E ROUTINE & PUSH LAST PARA.
                                    ;30124
                                    ;30125  .LIST           ;Re-enable full listing
```

```
                                        ;30126  .TOC    "ERCODE.MIC"
                                        ;30127  .TOC    "Revision 0.3"
                                        ;30128  ;        P. R. Guilbault
                                        ;30129
;30130  .NOBIN
;30131  .TOC    "       Revision History"
;30132
;30133  ; 00    Start of history
;30134
                                        ;30135  .BIN
                                        ;30136  .NOLIST         ;Disable listing of PCS code for quickie assemblies
```

```
                                        ;30137  .TOC    ''      Error handling        : IB.ERR - IBUF error routine''
                                        ;30138
                                        ;30139  ; ************************
                                        ;30140  ; *                      *
                                        ;30141  ; *  IBUF ERROR ROUTINE  *
                                        ;30142  ; *                      *
                                        ;30143  ; ************************
                                        ;30144
                                        ;30145  ;  THIS ROUTINE CHECKS FOR THE FOLLOWING CONDITIONS THAT CAN OCCUR
                                        ;30146  ;  ON I-BUFFER MEMORY REFERENCES AND TRANSFERS CONTROL
                                        ;30147  ;  TO THE  ROUTINE THAT HANDLES THE DETECTED CONDITION ON
                                        ;30148  ;  MICRO-TRAPS.
                                        ;30149  ;
                                        ;30150  ;       A) TBUF PARITY ERROR
                                        ;30151  ;
                                        ;30152  ;       B) PROTECTION VIOLATION
                                        ;30153  ;
                                        ;30154  ;       C) RDS ERROR
                                        ;30155  ;
                                        ;30156  ;       D) SBI READ TIMEOUT OR SBI ERROR  CONFIRMATION
                                        ;30157  ;
                                        ;30158  ;  ALL THESE CONDITIONS ARE FAULTED AND ALL INSTRUCTIONS FAULTED
                                        ;30159  ;  ARE RETRYABLE
                                        ;30160  ;
                                        ;30161  ;----------------------
                                        ;30162  ;  THIS MICRO-CODE IS  ACTIVATED WHEN ONE OF THE ABOVE MACHINE CHECKS
                                        ;30163  ;  OCCUR ON A I-BUFFER MEMORY REFERENCE.
                                        ;30164  ;
                                        ;30165  ;  THE IRD FIRMWARE TRANSFERS TO THE ENTRY TO THIS ROUTINE,
                                        ;30166  ;  ''IB.ERR'', WHEN IT DETECTS  THAT ONE OF THE ABOVE
                                        ;30167  ;  CONDITIONS IS PENDING
                                        ;30168
                                        ;30169
                                        ;30170
                                        ;30171  =00     ;  CALL CONSTRAINT BLOCK
                                        ;30172  IB.ERR:
                                        ;30173          ;00--------------------;
                                        ;30174          STOP.IB,                ;
                                        ;30175          IDED.SV] D,             ;  SAVE D-REG
U 0B80, 1000,003D,B980,3C00,0000,0ED0   ;30176          CALL,  J7EH.IBUF1       ;  PUSH RETURN ADDR TO SAVE
                                        ;30177
                                        ;30178          ;01--------------------;
U 0B81, 0000,003C,0180,F800,0000,0D8F   ;30179          J/M.FLT.03              ;  MERGE BACK INTO NORMAL FLOW
                                        ;30180
                                        ;30181          ;10--------------------;
U 0B82, 0000,003C,0180,F800,0000,0D8F   ;30182          J/M.FLT.03              ;  MERGE BACK INTO NORMAL FLOW
                                        ;30183  =;  END
```

```
                                ;30184  EH.IBUF1: ;----------------------;
U 0ED0, 0000,003C,81F0,2C00,0000,0ED1   ;30185          Q_ID[USTACK]          ;  READ RETURN ADDRESS OFF OF STACK
                                ;30186                                ;
                                ;30187          ;----------------------;
                                ;30188          D_Q,                  ;
U 0ED1, 0C00,003C,4DF0,2C00,0000,0ED2   ;30189          Q_ID[TBER1]           ;  GET TBUF ERROR REG #1
                                ;30190                                ;
                                ;30191                                ;
                                ;30192          ;----------------------;
U 0ED2, 0000,003C,B580,3C00,0000,0ED3   ;30193          ID[FPDA]_D            ;  WRITE FPD ADDRESS INTO ID-REG
                                ;30194                                ;
                                ;30195          ;----------------------;
                                ;30196          ALU_Q.AND.K[.4],      ;
U 0ED3, 0019,2034,1180,F800,0010,0ED4   ;30197          CLK.UBCC              ;  TEST FOR TB PARITY ERROR
                                ;30198                                ;
                                ;30199          ;----------------------;
                                ;30200          ALU_Q.AND.K[.2],      ;
                                ;30201          CLK.UBCC,             ;  TEST FOR PROTECTION VIOLATION
U 0ED4, 0019,2134,0980,F800,0010,08F0   ;30202          Z?                    ;  SBI PARITY ERROR?
                                ;30203                                ;
                                ;30204  =0      ;  BRANCH ON ALU ZERO RESULT (BEN/Z)
                                ;30205                                ;
                                ;30206          ;00--------------------;*** TB PARITY ERROR***
                                ;30207          RC[PTE.PA]_K[.A],     ;  ERROR CODE TO SUMMARY PARA
U 08F0, 0018,0038,F580,F9D8,0000,0ED9   ;30208          J/EH.IBUF5            ;
                                ;30209                                ;
                                ;30210          ;01--------------------;
                                ;30211          Q_ID[PARITY],         ;  GET CACHE PARITY ERROR REG
U 08F1, 0000,013C,79F0,2C00,0000,08F4   ;30212          Z?                    ;  PROTECTION VIOLATION?
                                ;30213                                ;
                                ;30214  =0      ;  BRANCH ON ALU ZERO RESULT(BEN/Z)
                                ;30215                                ;
                                ;30216          ;00--------------------;*** PROTECTION VIOLATION ***
U 08F4, 0000,003C,0180,F800,0000,0E63   ;30217          J/IB.PROT             ;  MERGE INTO PROTECTION CHECK FLOW
                                ;30218                                ;
                                ;30219          ;01--------------------;
                                ;30220          ALU_Q,    WORD,       ;  TEST FOR CACHE PARITY
U 08F5, 0001,603C,0180,F800,0010,0ED5   ;30221          CLK.UBCC              ;  SET ALU<N> IF SO
                                ;30222                                ;
                                ;30223          ;----------------------;
                                ;30224          Q_ID[SBI.ERR],        ;  GET SBI ERROR REGISTER
U 0ED5, 0000,1B3C,65F0,2C00,0000,0DC7   ;30225          ALU.N?                ;  CACHE PARITY ERROR?
                                ;30226                                ;
                                ;30227  =0111   ;0111----------------;  NO CACHE PARITY ERROR
                                ;30228          ALU_Q.AND.K[.80],     ;
                                ;30229          CLK.UBCC,             ;  SET ALU<Z> IF RDS ERROR
U 0DC7, 0019,2034,4180,F800,0010,0ED8   ;30230          J/EH.IBUF3            ;
                                ;30231                                ;
                                ;30232          ;1111----------------;*** CACHE PARITY ERROR ***
                                ;30233          RC[PTE.PA]_K[.F],     ;  ERROR CODE TO SUMMARY PARAMETER
U 0DCF, 0018,0038,6180,F9D8,0000,0ED9   ;30234          J/EH.IBUF5            ;
```

```
                                ;30235  EH.IBUF3:
                                ;30236          ;----------------------;
                                ;30237          ALU_Q.AND.K[.40],       ;
                                ;30238          CLK.UBCC,               ;  TEST FOR TIME OUT ERROR
U 0ED8, 0019,2134,3180,F800,0010,08F8   ;30239          Z?                      ;  RDS ERROR?
                                ;30240
                                ;30241
                                ;30242  =0      ;  BRANCH ON ALU ZERO RESULT (BEN/Z)
                                ;30243
                                ;30244          ;00--------------------;***RDS ERROR***
                                ;30245          RC[PTE.PA]_K[.C],       ;  ERROR CODE TO SUMMARY PARA
U 08F8, 0018,0038,8580,F9D8,0000,0ED9   ;30246          J/EH.IBUF5              ;
                                ;30247
                                ;30248          ;01--------------------;
                                ;30249          ALU_Q.AND.K[.8],        ;  TEST FOR CONFIRMATION ERROR
                                ;30250          CLK.UBCC,               ;
U 08F9, 0019,2134,0180,F800,0010,0900   ;30251          Z?                      ;  IB TIME OUT?
                                ;30252
                                ;30253  =0      ;  BRANCH ON ALU ZERO RESULT  (BEN/Z)
                                ;30254
                                ;30255          ;00--------------------;***  IB READ TIME-OUT ERROR  ***
                                ;30256          RC[PTE.PA]_K[.D],       ;  ERROR CODE TO SUMMARY PARAMETER
U 0900, 0018,0038,8980,F9D8,0000,0ED9   ;30257          J/EH.IBUF5              ;
                                ;30258
                                ;30259          ;01--------------------;
U 0901, 0000,013C,0180,F800,0000,090C   ;30260          Z?                      ;  IB ERROR CONFIRMATION?
                                ;30261
                                ;30262  =0      ;  BRANCH ON ALU ZERO RESULT (BEN/Z)
                                ;30263
                                ;30264          ;00--------------------;*** IB READ CONFIRMATION ERROR ***
                                ;30265          RC[PTE.PA]_K[.D],       ;  ERROR CODE TO SUMMARY REGISTER
U 090C, 0018,0038,8980,F9D8,0000,0ED9   ;30266          J/EH.IBUF5              ;
                                ;30267
                                ;30268          ;01--------------------;
U 090D, 0000,003D,0180,F800,0000,0EE0   ;30269          CALL,   J/EH.USEQ       ;  IMPOSSIBLE ENTRY IF MACHINE WORKING
                                ;30270
                                ;30271  EH.IBUF5:
                                ;30272          ;----------------------;
                                ;30273          RC[PC.SV]_PC,           ;  SAVE PC CONTENTS
                                ;30274          PC_VIBA,                ;
                                ;30275          SET.NEST.ERR,           ;
U 0ED9, 0014,0038,0180,F9E2,2C00,0EDA   ;30276          J/EH.RTO3               ;
```

```
;30277  .TOC    ''      Error handling        : Machine check errors''
;30278
;30279  ; ******************************
;30280  ; *                            *
;30281  ; *  MACHINE CHECK ERRORS       *
;30282  ; *                            *
;30283  ; ******************************
;30284  ;
;30285  ;DESCRIPTION:
;30286  ;------------
;30287  ;  THIS FIRMWARE IS USED TO FAULT, ABORT, OR HALT THE INSTRUCTION
;30288  ;  IN PROGRESS WHEN A MACHINE CHECK OCCURS.
;30289  ;
;30290  ;  FAULT: THE INSTRUCTION IS FAULTED IF THE MACHINE CHECK IS NOT
;30291  ;         A CONTROL STORE PARITY ERROR OR A SECOND MACHINE CHECK; OR
;30292  ;         IS NOT IN MEMORY MANAGEMENT OR INTERRUPT/EXCEPTION FIRMWARE
;30293  ;         WHEN THE MACHINE CHECK OCCURS.
;30294  ;
;30295  ;  ABORT: AN INSTRUCTION IS ABORTED IF IT IS IN MEMORY MANAGEMENT
;30296  ;         OR INTERRUPT/EXCEPTION FIRMWARE WHEN THE CHECK OCCURS, AND
;30297  ;         ON A CONTROL STORE PARITY CHECKS.
;30298  ;
;30299  ;  HALT:  THE INSTRUCTION IS HALTED IF A MACHINE CHECK IS
;30300  ;         DETECTED AS HAVING OCCURRED DURING THE ERROR HANDLING
;30301  ;         FIRMWARE, OR ON A DOUBLE SBI CP ERROR, NOTED IN
;30302  ;         ID[SBI.ERR<2>].  THE MACHINE IS ALWAYS HALTED IF THE
;30303  ;         PROCESSOR IS IN CONSOLE MODE WHEN THE MACHINE
;30304  ;         CHECK OCCURRS.
;30305  ;
;30306  ;  FIRMWARE MICRO-TRAP ENTRY POINTS:
;30307  ;-----------------------------------
;30308  ;
;30309  ;  THIS FIRMWARE IS INITIATED BY ANY ONE OF THE FOLLOWING
;30310  ;  MICRO-TRAPS  A) CS PARITY ERROR
;30311  ;               B) TB PARITY ERROR
;30312  ;               C) CACHE PARITY ERROR
;30313  ;               D) RDS
;30314  ;               E) READ TIME OUT
;30315  ;               F) MICRO-SEQUENCING ERROR
;30316  ;
;30317
;30318  ;OTHER FIRMWARE INTERFACES
;30319  ;------------------------
;30320  ;
;30321  ;  THIS FIRMWARE USES THE 'MEMORY MANAGEMENT FAULT HANDLING
;30322  ;  ROUTINE'' AND THE ''GENERAL I&E ROUTINE''
;30323  ;
;30324  ;SPECIFICATIONS:
;30325  ;--------------
;30326  ;
;30327  ;SEE 'STAR MACHINE CHECK ABORT/FAULT/HALT DESCRIPTION AND
;30328  ;SPECIFICATION' FOR DETAILS ON ERROR HANDLING.
```

```
                                ;30329          ;  CONTROL STORE PARITY ERROR MICRO-TRAP ENTRY
                                ;30330          ;*********************************************
                                ;30331  10F:
                                ;30332  EH.CS01: ;----------------------;
                                ;30333          RC[PTE.PA]_K[.1],       ;  CS ERROR PARAMETER
                                ;30334          STOP.IB,                ;
                                ;30335          MCT/MEM.NOP,            ;  STOP IBUF MEMORY REQUESTS
U 010F, 1018,1138,0580,09D8,0000,0CCB   ;30336          NEST.ERR?               ;  IS NESTED ERROR FLAG SET?
                                ;30337
                                ;30338  =1011   ;  BRANCH ON NESTED ERROR FLAG, (BEN/LAST.REF<NESTED ERROR>)
                                ;30339
                                ;30340  EH.CS02:;1011----------------;***NESTED ERROR FLAG NOT SET***
                                ;30341          ID[D.SV]_D,             ;  SAVE ENTRY CONTENTS OF D-REG
                                ;30342          RC[PC.SV]_PC,  PC_VA,   ;  SAVE PC IN RC-REG
                                ;30343          SET.NEST.ERR,           ;  SET NESTED ERROR FLAG
U 0CCB, 0014,0038,B980,3DE1,2C00,0EDE   ;30344          J/EH.RT07               ;
                                ;30345
                                ;30346          ;1111----------------;***NESTED ERROR FLAG SET***
                                ;30347          ID[D.SV]_D,             ;  SAVE ENTRY CONTENTS OF D-REG
                                ;30348          PC_VA,                  ;  MOVE VA-REG TO PC-REG
U 0CCF, 0000,003C,B980,3C01,0000,0EDE   ;30349          J/EH.RT07               ;
                                ;30350
                                ;30351  10D:    ;  READ TIMEOUT MICRO-TRAP ENTRY
                                ;30352          ;*******************************
                                ;30353
                                ;30354  EH.RT01:;----------------------;
                                ;30355          STOP.IB,                ;
                                ;30356          MCT/MEM.NOP,            ;  STOP IBUFFER MEMORY REFERENCES
                                ;30357          RC[PTE.PA]_0,           ;  ERROR ID TO SUMMARY PARA.
U 010D, 1003,113C,0180,09D8,0000,0DDB   ;30358          NEST.ERR?               ;
                                ;30359
                                ;30360  =1011   ;  BRANCH ON NESTED ERROR FLAG (BEN/LAST.REF <NESTED ERROR>)
                                ;30361
                                ;30362  EH.RT02: ;1011----------------;***NESTED ERROR FLAG NOT SET***
                                ;30363          ID[D.SV]_D,  D_Q,       ;  SAVE ENTRY CONTENTS OF D-REG
                                ;30364          RC[PC.SV]_PC,  PC_VA,   ;  SAVE ENTRY  CONTENTS OF PC-REG
                                ;30365          SET.NEST.ERR,           ;  SET NESTED ERROR FLAG
U 0DDB, 0C14,0038,B980,3DE1,2C00,0EDA   ;30366          J/EH.RT03
                                ;30367
                                ;30368          ;1111----------------;***NESTED ERROR FLAG SET***
                                ;30369          ID[D.SV]_D,             ;  SAVE ENTRY CONTENTS OF D-REG
                                ;30370          PC_VA,                  ;  SET UP VA-REP FOR SAVING
U 0DDF, 0000,003C,B980,3C01,0000,0EDE   ;30371          J/EH.RT07               ;
                                ;30372  =;  END
```

```
                                        ;30373  EH.RT03:;----------------------;  DO FAULT SEQUENCE
                                        ;30374          ID[Q.SV]_D,             ;  SAVE ENTRY CONTENTS OF Q-REG
U OEDA, 0014,0038,BD80,3DC8,0000,0EDC   ;30375          RC[VA.SV]_PC            ;  SAVE ENTRY CONTENTS OF VA-REG
                                        ;30376
                                        ;30377          ;----------------------;
                                        ;30378          MCT/MEM.NOP,            ;  INHIBIT ALLOW IB MCT CODE
U OEDC, 0018,0038,1D80,09E8,0000,0EDD   ;30379          RC[SC.SV]_K[SC]         ;  SAVE ENTRY CONTENTS OF SC-REG
                                        ;30380
                                        ;30381          ;----------------------;
                                        ;30382          RC[LC.SV]_LC,           ;  SAVE ENTRY CONTENTS OF LC-REG
U OEDD, 0010,0038,49F0,2DC0,0000,0924   ;30383          Q_ID[TBER0]             ;  GET TBUF ERROR REG#0
                                        ;30384
                                        ;30385
                                        ;30386  =0      ;00--------------------;
                                        ;30387          RC[PTE.VA]_Q,           ;  SAVE TBER0'S CONTENTS IN RC
                                        ;30388          CALL,                   ;
U 0924, 0001,3C3D,0180,F9D0,0000,035B   ;30389          CONSOLE.MODE?,J/EH.INIT ;  GO & DO ERROR INITIATE SEQUENCE
                                        ;30390
                                        ;30391          ;01--------------------;
U 0925, 0000,003C,0180,F800,0000,0EA9   ;30392          J/M.FLT.PRB             ;  GO & SETUP FOR FAULT
                                        ;30393                                  ;  RETURNS TO EH.FLT01
                                        ;30394
                                        ;30395  EH.RT07:
                                        ;30396          ;----------------------;
                                        ;30397          RC[VA.SV]_PC,           ;  SAVE ENTRY CONTENTS OF VA-REG
U OEDE, 0014,0038,49F0,2DC8,0000,0928   ;30398          Q_ID[TBER0]             ;
                                        ;30399
                                        ;30400  =0
                                        ;30401          ;00--------------------;  DO ABORT SEQUENCE
                                        ;30402          RC[PTE.VA]_Q,           ;  SAVE ENTRY CONTENTS OF TBER0
                                        ;30403          CALL,                   ;
U 0928, 0001,3C3D,0180,F9D0,0000,035B   ;30404          CONSOLE.MODE?,J/EH.INIT ;
                                        ;30405
                                        ;30406          ;01--------------------;
U 0929, 0000,003C,0180,F800,0000,0EE9   ;30407          J/EH.ABORT              ;  GO & DO ABORT SEQUENCE
                                        ;30408  =;  END
```

```
                                        ;30409  10C:    ;  RDS ERROR MICRO-TRAP
                                        ;30410          ;*********************
                                        ;30411
                                        ;30412  EH.RDS: ;-----------------------;
                                        ;30413          STOP.IB,                ;
                                        ;30414          MCT/MEM.NOP,            ;  STOP IBUFFER MEMORY REFERENCE
                                        ;30415          RC[PTE.PA]_0+K[.4]+1,   ;  NOTE ERROR ID IN SUMMARY PARA
U 010C, 101B,1110,1180,09D8,0000,0DDB   ;30416          NEST.ERR?, J/EH.RT02    ;  MERGE INTO READ TIMEOUT FLOW
                                        ;30417
                                        ;30418
                                        ;30419
                                        ;30420  108:    ;  CACHE PARITY ERROR MICRO-TRAP
                                        ;30421          ;*******************************
                                        ;30422
                                        ;30423  EH.CPE: ;-----------------------;
                                        ;30424          STOP.IB, MCT/MEM.NOP,   ;  STOP IBUFFER MEMORY REFERENCE
                                        ;30425          RC[PTE.PA]_K[.3],       ;  NOTE ERROR ID IN SUMMARY PARA
U 0108, 1018,1138,0D80,09D8,0000,0DDB   ;30426          NEST.ERR?, J/EH.RT02    ;  MERGE INTO READ TIMEOUT FLOW
                                        ;30427
                                        ;30428
                                        ;30429
                                        ;30430  107:    ;  TRANSLATION BUFFER PARITY ERROR MICRO-TRAP
                                        ;30431          ;*********************************************
                                        ;30432
                                        ;30433  EH.TBE: ;-----------------------;
                                        ;30434          STOP.IB, MCT/MEM.NOP,   ;  STOP IBUFFER MEMORY REFERENCE
                                        ;30435          RC[PTE.PA]_K[.2],       ;  NOTE ERROR ID IN SUMMARY PARA
U 0107, 1018,1138,0980,09D8,0000,0DDB   ;30436          NEST.ERR?, J/EH.RT02    ;  MERGE INTO READ TIMEOUT FLOW
                                        ;30437
                                        ;30438
                                        ;30439
                                        ;30440
                                        ;30441          ;  MICRO-SEQUENCING ERROR(NOT SUPPOSE TO BE HERE)
                                        ;30442          ;*****************************************************
                                        ;30443
                                        ;30444  EH.USEQ:
                                        ;30445  M.ERROR:;-----------------------;
                                        ;30446          STOP.IB, MCT/MEM.NOP,   ;  STOP IBUFFER MEMORY REFERENCE
                                        ;30447          RC[PTE.PA]_K[.6],       ;  NOTE ERROR ID IN SUMMARY PARA
U 0EE0, 1018,1138,D580,09D8,0000,0CCB   ;30448          NEST.ERR?, J/EH.CS02    ;  MERGE INTO READ TIMEOUT FLOW
```

```
                                        ;30449  .TOC    ""      Error handling        : Error handling initialize''
                                        ;30450
                                        ;30451  ; ****************************************
                                        ;30452  ; *                                      *
                                        ;30453  ; * ERROR HANDLING INITIALIZATION ROUTINE *
                                        ;30454  ; *                                      *
                                        ;30455  ; ****************************************
                                        ;30456  ;
                                        ;30457  ;  THIS ROUTINE CHECKS THE ERROR FIRST PASS FLAG (EFP),
                                        ;30458  ;  THE CP DOUBLE SBF ERROR, AND CONSOLE MODE. IF ANY ARE
                                        ;30459  ;  'ON'' THE MACHINE IS HALTED.
                                        ;30460  ;
                                        ;30461  ;  IF THE MACHINE IS NOT HALTED, THE 'EFP'' FLAG
                                        ;30462  ;  IS SET AND A TIME-OUT OF APPROXIMATELY 154 MICRO-
                                        ;30463  ;  SECONDS IS PERFORMED.
                                        ;30464  ;
                                        ;30465  ;  CALLING SEQUENCE
                                        ;30466  ;-----------------------
                                        ;30467  ;  CALL, CONSOLE.MODE? J/EH.INIT
                                        ;30468  ;
                                        ;30469  ;  OUTPUT & RESULTS
                                        ;30470  ;-----------------------
                                        ;30471  ;  ID[FAULT<25>] BIT SET
                                        ;30472  ;
                                        ;30473  ;  BRANCH ON NOT CONSOLE MODE (BEN/PSL.MODE <-CONSOLE.MODE>)
                                        ;30474
                                        ;30475  =11011
                                        ;30476  EH.INIT:;11011-------------------;***CONSOLE MODE***
                                        ;30477          D_K[.2],                ;  ERROR CODE FOR ERROR PARK.
U 035B, 0818,0039,0980,F800,0000,0F1B   ;30478          CALL,   J/M.CNSL        ;  GO & HALT MACHINE
                                        ;30479
                                        ;30480  ;11111-----------------;
U 035F, 0000,003C,6DF0,2C00,0000,0EE1   ;30481          Q_ID[FAULT]             ;  GET 'EFP'' BIT
                                        ;30482  =;  END
                                        ;30483  ;---------------------;
U 0EE1, 0000,003C,B980,F800,0084,6EE2   ;30484          SC_K[.19]               ;  SETUP MASK TO TEST 'EFP'' BIT
                                        ;30485
                                        ;30486  ;---------------------;
U 0EE2, 0801,2024,0180,F800,0000,0EE4   ;30487          D_Q.ANDNOT.MASK         ;  TEST TO SEE IF 'EFP'' IS SET
                                        ;30488
                                        ;30489  ;---------------------;
                                        ;30490          D_Q.ORNOT.MASK,         ;
                                        ;30491          SC_K[.10],              ;  SET 'EFP'' BIT POSITION
U 0EE4, 0801,2D1C,6580,F800,0084,6FCD   ;30492          D.NE.0?                 ;  WAS IT SET?
                                        ;30493
                                        ;30494  =101    ;  BRANCH ON D.NE.0, BEN/SIGNS
                                        ;30495
                                        ;30496  ;101-------------------;
                                        ;30497          SC_SC+K[.3],            ;  SETUP MASK TO RESET FAULT
U 0FCD, 0000,003C,0D80,F800,0084,8EE5   ;30498          J/EH.INIT1              ;  POSITION IN D-REG
                                        ;30499
                                        ;30500  ;111-------------------;***'EFP'' BIT SET***
                                        ;30501          D_0+K[.4]+1,            ;  ERROR CODE FOR HALT
U 0FCF, 081B,0011,1180,F800,0000,0F1B   ;30502          CALL,J/M.CNSL           ;  GO & HALT
                                        ;30503  =;  END
```

```
                                        ;30504  EH.INIT1:
                                        ;30505          ;-----------------------;
U 0EE5, 0801,0034,0180,F800,0000,0EE6   ;30506          D_D.AND.MASK            ;  RESET BIT IN <FAULT> POS.
                                        ;30507
                                        ;30508          ;-----------------------;
U 0EE6, 0000,003C,6D80,3C00,0000,0EE8   ;30509          ID[FAULT]_D             ;  SET 'EFP' BIT IN ID[FAULT]
                                        ;30510
                                        ;30511          ;-----------------------;
                                        ;30512          D_K[.3FF],              ;  SETUP COUNT FOR TIMEOUT
U 0EE8, 0818,0038,8180,F800,0000,0FD7   ;30513          J/EH.INIT3
                                        ;30514
                                        ;30515  =101    ;  BRANCH ON D.NE.0, BEN/SIGNS
                                        ;30516
                                        ;30517  EH.INIT2:;101-------------------;***COUNT EQUAL ZERO***
U 0FD5, 0000,003E,0180,F800,0000,0001   ;30518          RETURN1                 ;  RETURN TO CALLING ROUTINE
                                        ;30519
                                        ;30520  EH.INIT3: ;111-----------------;
                                        ;30521          D_D-K[.1],              ;  DECREMENT TIMEOUT COUNT
U 0FD7, 0819,0D00,0580,F800,0000,0FD5   ;30522          D.NE.0?.J/EH.INIT2      ;  IS IT ZERO?
                                        ;30523  =;  END
```

```
                                        ;30524  .TOC     "       Error handling          : Error snapshot routine"
                                        ;30525
                                        ;30526          ;************************************
                                        ;30527          ;                                  ;
                                        ;30528          ;  ERROR SNAPSHOT ROUTINE          ;
                                        ;30529          ;                                  ;
                                        ;30530          ;************************************
                                        ;30531
                                        ;30532
                                        ;30533          ;IF ABORT SEQUENCE, ENTRY @ EH.ABORT
                                        ;30534
                                        ;30535          ;IF FAULT SEQUENCE, ENTRY @ EH.FAULT
                                        ;30536
                                        ;30537          ;THIS ROUTINE LOGS OUT ONTO THE STACK THE PARAMETERS AND INFO
                                        ;30538          ;RELATED TO THE ERROR CONDITION BEING HANDLED.  THIS INCLUDES
                                        ;30539          ;SNAPSHOTTING THE MACHINE STATE, LOGGING THIS STATE ONTO THE
                                        ;30540          ;STACK, AND RESETTING SBI.ERR<CP TIME OUT FLAGS>, CACHE PARITY,
                                        ;30541          ; AND TBER1 ID-REGS.   THEN AN EXCEPTION IS TAKEN(MACHINE CHECK).
                                        ;30542
                                        ;30543
                                        ;30544
                                        ;30545  EH.ABORT: ;--------------------;  GET SUMMARY PARAMETER
U 0EE9, 0810,0038,0180,F958,0000,0EEA   ;30546          D_RC[PTE.PA]            ;
                                        ;30547
                                        ;30548          ;----------------------;
U 0EEA, 0019,0030,CD80,F9D8,0000,091C   ;30549          RC[PTE.PA]_D.OR.K[.F0]  ;  NOTE ABORT IN SUMMARY PARA.
                                        ;30550
                                        ;30551  =0*     ;00--------------------;
                                        ;30552          Q_RC[PC.SV],            ;  LOAD Q-REP WITH SAVED PC VALUE
U 091C, 0010,0039,01C0,F960,0000,0EB8   ;30553          CALL,   J/BAKUP.PC      ;  GO BACK-UP PC TO BEGIN OF INSTR.
                                        ;30554
                                        ;30555  EH.FAULT: ;01--------------------;
U 091E, 0000,003C,31F0,2C00,0000,0EEB   ;30556          Q_ID[CES]               ;  GET CPU ERROR STATUS REGISTER
                                        ;30557
                                        ;30558          ;----------------------;
U 0EEB, 0000,003C,79B0,F800,0084,6EF0   ;30559          SC_K[.30]               ;  SET UP ADDR FOR ID[T1]
                                        ;30560
                                        ;30561          ;----------------------;
                                        ;30562          SC_SC+1,                ;  SETUP ADDR FOR T1
                                        ;30563          Q_ID[USTACK],           ;  GET TRAPPED U-ADDR
U 0EF0, 0C00,003C,81F0,2C00,0080,CEF1   ;30564          D_Q                     ;  MOVE CES TO D-REG
                                        ;30565
                                        ;30566
                                        ;30567          ;----------------------;
                                        ;30568          ID(SC)_D,               ;  SAVE CES IN T1
                                        ;30569          D_Q,                    ;
U 0EF1, 0C00,003C,0180,3400,0080,CEF2   ;30570          SC_SC+1                 ;
                                        ;30571
                                        ;30572
                                        ;30573          ;----------------------;
                                        ;30574          ID(SC)_D,               ;  SAVE U-ADDR IN T2
U 0EF2, 0000,003C,0180,3400,0080,CEF3   ;30575          SC_SC+1                 ;
```

```
                                        ;30576          ;----------------------;
                                        ;30577          D_RC[VA.SV],           ;  GET VA/VIBA SAVED CONENTS
U 0EF3, 0810,0038,B9F0,2D48,0000,0EF4   ;30578          Q_ID[D.SV]             ;  GET SAVED D-REG CONTENTS
                                        ;30579
                                        ;30580
                                        ;30581          ;----------------------;
                                        ;30582          ID(SC)_D,              ;  SAVE VA/VIBA IN T3
                                        ;30583          D_Q,                   ;
U 0EF4, 0C00,003C,0180,3400,0080,CEF5   ;30584          SC_SC+1                ;
                                        ;30585
                                        ;30586
                                        ;30587          ;----------------------;
                                        ;30588          ID(SC)_D,              ;  SAVE D-REG SAVED CONTENTS IN T4
                                        ;30589          D_RC[PTE.VA],          ;  GET SAVED TBER0 CONTENTS
U 0EF5, 0810,0038,0180,3550,0080,CEF8   ;30590          SC_SC+1                ;
                                        ;30591
                                        ;30592          ;----------------------;
                                        ;30593          ID(SC)_D,              ;  SAVE TBER0 IN T5
U 0EF8, 0000,003C,0180,3400,0080,CEF9   ;30594          SC_SC+T                ;
                                        ;30595
                                        ;30596          ;----------------------;
U 0EF9, 0000,003C,4DF0,2C00,0000,0EFA   ;30597          Q_ID[TBER1]            ;
                                        ;30598
                                        ;30599          ;----------------------;
                                        ;30600          D_Q,                   ;
U 0EFA, 0C00,003C,69F0,2C00,0000,0EFB   ;30601          Q_ID[TIME.ADDR]        ;
                                        ;30602
                                        ;30603          ;----------------------;
                                        ;30604          ID(SC)_D,              ;  SAVE TBER1 IN T6
                                        ;30605          D_Q,                   ;
U 0EFB, 0C00,003C,0180,3400,0080,CEFC   ;30606          SC_SC+1                ;
                                        ;30607
                                        ;30608          ;----------------------;
                                        ;30609          ID(SC)_D,              ;  SAVE TIME OUT ADDR IN T7
U 0EFC, 0000,003C,0180,3400,0080,CEFD   ;30610          SC_SC+T                ;
                                        ;30611
                                        ;30612          ;----------------------;
U 0EFD, 0000,003C,79F0,2C00,0000,0F00   ;30613          Q_ID[PARITY]           ;
                                        ;30614
                                        ;30615          ;----------------------;
                                        ;30616          D_Q,                   ;
U 0F00, 0C00,003C,65F0,2C00,0000,0F01   ;30617          Q_ID[SBI.ERR]          ;
                                        ;30618
                                        ;30619          ;----------------------;
                                        ;30620          ID(SC)_D,              ;  SAVE CACHE PARITY ERROR REG
U 0F01, 0000,003C,0180,3400,0080,CF02   ;30621          SC_SC+T                ;  IN T8
```

```
                                        ;30622          ;----------------------;
                                        ;30623          Q_RC[PTE.PA],          ;  GET SAVED ERROR PARAMETER
                                        ;30624          D_Q,                   ;
U 0F02, 0C10,0038,79C0,3D58,0000,0F03   ;30625          ID[PARITY]_D           ;  RESET PARITY <15>
                                        ;30626
                                        ;30627          ;----------------------;
                                        ;30628          Q_C.AND.K[.F], CLK.UBCC,;  TEST FOR CP TIME OUT TAKEN
                                        ;30629          ID(SC)_D,              ;  SAVE SBI ERROR REG IN T9
U 0F03, 0019,2034,61C0,3400,0090,CF04   ;30630          SC_SC+T                ;
                                        ;30631
                                        ;30632          ;----------------------;
                                        ;30633          D_D.AND.K[.8000],      ;  AND OUT ALL BUT ENABLE BIT
U 0F04, 0819,0134,4580,F800,0000,092C   ;30634          Z?                     ;  WAS TRAP CP RD TIME OUT?
                                        ;30635
                                        ;30636  =0      ;00--------------------;  ** NOT RD TO U-TRAP **
                                        ;30637          RC[T6]_D.OR.K[.1F00],  ;  SETUP TO RESET RD TIME OUT BIT
U 092C, 0019,0030,9180,F980,0000,0B88   ;30638          J/EH.FLT7              ;
                                        ;30639
                                        ;30640          ;01--------------------;
                                        ;30641          RC[T6]_D.OR.K[.1F00],  ;  SETUP TO RESET RD TIME OUT BIT
                                        ;30642          D_0,                   ;  ZERO FOR PENDING INTERRUPT FLAG
U 092D, 0F19,0030,9180,F980,0000,0B8A   ;30643          J/EH.FLT9              ;
                                        ;30644
                                        ;30645
                                        ;30646  =00
                                        ;30647  EH.FLT7:;00--------------------;  GO SEE IF PENDING ERROR INTERRUPT
                                        ;30648          SC_K[ZERO],            ;
U 0B88, 0000,003D,1980,F800,0084,6F10   ;30649          CALL,  J/EH.PUSH       ;
                                        ;30650
                                        ;30651          ;01--------------------;***ERROR INTERRUPT PENDING***
U 0B89, 0818,0038,4D80,F800,0000,0B8A   ;30652          D_K[.FF00]             ;  FLAG NOTING PENDING ERROR INTERRUPT TO D
                                        ;30653
                                        ;30654  EH.FLT9:;10--------------------;
                                        ;30655          D_D.OR.RC[PTE.PA],     ;  FINISH SETTING UP SUMMARY REG
U 0B8A, 0811,0030,7980,F958,0084,6F06   ;30656          SC_K[.30]              ;  SET UP TO WRITE TO
                                        ;30657  =;  END
                                        ;30658
                                        ;30659          ;----------------------;
                                        ;30660          D_RC[T6],              ;  GET RESET PARAMETER FOR SBI.ERR REG
                                        ;30661          ID(SC)_D,              ;  SUMMARY PARA /TO
U 0F06, 0810,0038,0180,3530,0080,CF08   ;30662          SC_SC+T                ;
                                        ;30663
                                        ;30664          ;----------------------;
                                        ;30665          ID[SBI.ERR]_D,         ;  RESET CP TIMEOUT AND ERROR CONFIRMATION
U 0F08, 0F00,003C,6580,3C00,0000,0F09   ;30666          D_0                    ;
                                        ;30667
                                        ;30668          ;----------------------;
U 0F09, 0000,003C,4D80,3C00,0000,0938   ;30669          ID[TBER1]_D            ;  RESET TBER1
```

```
                                        ;30670  =0      ;00--------------------;
                                        ;30671          RC[TO]_K[.4],           ;  EXCEPTION VECTOR TO RC[TO]
U 0938, 0018,0039,1180,F980,0000,0DE5   ;30672          CALL,   J/EXCPTN        ;  GO TO EXCEPTION HANDLING ROUTINE
                                        ;30673
                                        ;30674          ;01--------------------;  RETURN FROM EXCEPTION HANDLING
U 0939, 0000,003C,7980,F800,0084,6B90   ;30675          SC_K[.30]               ;  START SETUP OF SC TO 3A (HEX)
                                        ;30676
                                        ;30677  =00     ;00--------------------;
                                        ;30678          SC_SC+K[.9],            ;  PUSH INFO ONTO STACK
U 0B90, 0000,003D,D980,F800,0084,8F10   ;30679          CALL,   J/EH.PUSH       ;
                                        ;30680
                                        ;30681          ;01--------------------;***ERROR WHILE PUSHING***
                                        ;30682          D_0+K[.4]+1,            ;  HALT MACHINE
U 0B91, 081B,0011,1180,F800,0000,0F1B   ;30683          CALL,J/M.CNSL           ;
                                        ;30684
                                        ;30685          ;10--------------------;***PUSHED INFO OK***
U 0B92, 0818,0038,7580,F800,0000,0B93   ;30686          D_K[.20]                ;  START SETTING UP LENGTH PARA
                                        ;30687
                                        ;30688          ;--------------------;  FINISH SETTING UP
U 0B93, 0019,0030,01C0,F800,0000,0BA0   ;30689          Q_D.OR.K[.8]            ; LENGTH PARAMETER (40)
                                        ;30690
                                        ;30691
                                        ;30692
                                        ;30693
                                        ;30694  =00     ;00--------------------;
                                        ;30695          LAB_R[SP],              ;  GET STACK POINTER
                                        ;30696          SC_K[.30],              ;  PUSH COUNT ONTO STACK
U 0BA0, 0000,003D,7980,FA70,0084,6F13   ;30697          CALL,   J/EH.PUSH5      ;
                                        ;30698
                                        ;30699          ;01--------------------;***ERROR WHILE PUSHING***
                                        ;30700          D_0+K[.4]+1,            ;  HALT MACHINE
U 0BA1, 081B,0011,1180,F800,0000,0F1B   ;30701          CALL,J/M.CNSL           ;
                                        ;30702
                                        ;30703          ;10--------------------;***PUSHED INFO OK***
U 0BA2, 0818,0038,9180,F800,0000,0BA3   ;30704          D_K[.1F00]              ;  START DEV. "1F" FOR IPL
                                        ;30705
                                        ;30706
                                        ;30707          ;--------------------;
                                        ;30708          D_D.SWAP,               ;
U 0BA3, 0B00,003C,3DF0,2C00,0000,0F0A   ;30709          Q_ID[PSL]               ;  GET PSL
                                        ;30710
                                        ;30711          ;--------------------;
                                        ;30712          D_D.OR.Q,               ;  SET IPL FIELD TO "1F"
U 0F0A, 081D,0030,ED80,F800,0084,6F08   ;30713          SC_K[.1B]               ;  GO BACK INTO EXCEPTION FLOW
                                        ;30714
                                        ;30715          ;--------------------;
                                        ;30716          ID[PSL]_D,              ;  UPDATE PSL WITH IPL=1F
U 0F0B, 0000,003C,3D80,3C00,0000,0CF7   ;30717          J/EXCPT3                ;(SPECIAL ENTRY FOR ERROR HANDLING)
```

```
                                        ;30718  .TOC    "       Error handling        : Push parameters"
                                        ;30719
                                        ;30720          ;  PUSH PARAMETERS ON STACK & CHECK FOR SBI ERRORS
                                        ;30721          ;************************************************
                                        ;30722
                                        ;30723          ;  THIS ROUTINE PUSHES ID[T0-T9] REGS ON TO THE STACK.
                                        ;30724          ;  IT FIRST CHECKS TO FIND OUT IF THERE IS A PENDING SBI
                                        ;30725          ;  ERROR. IF THERE IS, A PENDING SBI ERROR, THE ROUTINE IS EXITED
                                        ;30726          ;  WITHOUT PUSHING THE NEXT ID REGISTER.
                                        ;30727
                                        ;30728          ;  ROUTINE CALL SEQUENCE:
                                        ;30729
                                        ;30730          ;               A) IF TO CHECK FOR PENDING SBI ERROR ONLY
                                        ;30731          ;                  SC_0,
                                        ;30732          ;                  CA[L,J/EH.PUSH
                                        ;30733          ;
                                        ;30734          ;               B) TO PUSH, AREAS & SBI ERROR CHECK ON EA ONE
                                        ;30735          ;
                                        ;30736          ;                  SC_K[.39],
                                        ;30737          ;                  CA[L, J/EH.PUSH
                                        ;30738
                                        ;30739          ;ROUTINE EXITS: RETURN1,  SBI ERROR PENDING EXIT
                                        ;30740          ;               RETURN2,  FINISHED SUCCESSFULLY (NO PENDING ERROR)
                                        ;30741
                                        ;30742  EH.PUSH: ;----------------------;  DEVELOP MASK TO CHECK FOR PENDING
U 0F10, 0000,003C,65F0,2C00,0000,0F11   ;30743          Q_ID[SBI.ERR]           ;  GET SBI ERROR REG TO TEST
                                        ;30744
                                        ;30745          ;----------------------;
                                        ;30746          ALU_Q.AND.K[.1F00],     ;  TEST FOR TIME OUT
U 0F11, 0019,2034,9180,F800,0010,0F12   ;30747          CLK.UBCC                ;  SET ALU.Z IF NO ERRORS
                                        ;30748
                                        ;30749          ;----------------------;
                                        ;30750          EALU_SC.ANDNOT.K[.EF],  ;
U 0F12, 0000,013C,3D80,F800,0014,493C   ;30751          CLK.UBCC,Z?             ;  TEST FOR EXIT CONDITION
                                        ;30752
                                        ;30753  =0      ;  BRANCH ON ALU.Z (BEN/Z)
                                        ;30754
                                        ;30755          ;00--------------------;***NO, SBI PENDING ERROR DETECTED***
U 093C, 0000,003E,0180,F800,0000,0001   ;30756          RETURN1                 ;  ERROR RETURN TO CALLER
                                        ;30757
                                        ;30758          ;01--------------------;
U 093D, 0000,123C,0180,F800,0000,0DEB   ;30759          EALU.Z?                 ;  COMPLETION EXIT? (SC=30) HEX
                                        ;30760  =;  END
                                        ;30761
                                        ;30762  =1011   ;  BRANCH ON EALU.Z (BEN/ALU <EALU.Z>)
                                        ;30763
                                        ;30764  EH.PUSH3:
                                        ;30765          ;1011------------------;
                                        ;30766          Q_ID(SC),               ;  GET NEXT PARA TO PUSH
U 0DEB, 0000,003C,01F0,2670,0000,0F13   ;30767          LAB_R[SP],J/EH.PUSH5    ;  LOAD STACKPOINTER INTO LAB-REG
                                        ;30768
                                        ;30769          ;1111------------------;***EXIT CONDITION***
                                        ;30770          D_0,                    ;
U 0DEF, 0F00,003E,0180,F800,0000,0002   ;30771          RETURN2                 ;  OK RETURN
                                        ;30772  =
```

```
                                    ;30773  EH.PUSH5: ;----------------------;
                                    ;30774          D_Q,                    ;  MOVE PARA TO D-REG
U 0F13, 0C18,0000,1180,FAF0,0200,0F18 ;30775        R[SP]&VA_LA-K[.4]       ;  PUSH STACK ADDRESS +4
                                    ;30776
                                    ;30777          ;----------------------;
U 0F18, 0000,003C,0180,3000,0000,0F19 ;30778        CACHE_D[LONG]           ;  PUSH PARA ONTO STACK
                                    ;30779
                                    ;30780          ;----------------------;
U 0F19, 0000,003C,65F0,2C00,0000,0940 ;30781        Q_ID[SBI.ERR]           ;  GET SBI ERROR REGISTER
                                    ;30782
                                    ;30783  ; ************************************************
                                    ;30784  ; * Patch no. 023, PCS 0F19 trapped to WCS 115A *
                                    ;30785  ; ************************************************
                                    ;30786
                                    ;30787  =0      ;  BRANCH ON ALU.Z (BEN/Z)
                                    ;30788
                                    ;30789  EH.PUSH7: ;00--------------------;
                                    ;30790          ALU_Q.AND.K[.2],        ;
                                    ;30791          CLK_UBCC,               ;  TEST TO SEE IF SBI BUSY
U 0940, 0019,2034,0980,F800,0010,0F1A ;30792        J/EH.PUSH9              ;
                                    ;30793
                                    ;30794          ;01--------------------;***SBI NOT BUSY***
                                    ;30795          SC_SC-K[.1],            ;  GO TO NEXT REGISTER
U 0941, 0000,003C,0580,F800,0084,AF10 ;30796        J/EH.PUSH               ;  GO & SEE IF SBI ERROR OCCURRED
                                    ;30797  =;END
                                    ;30798
                                    ;30799
                                    ;30800  EH.PUSH9: ;----------------------;
                                    ;30801          Q_ID[SBI.ERR],          ;  GET SBI ERROR REG AGAIN
U 0F1A, 0000,013C,65F0,2C00,0000,0940 ;30802        Z?,     J/EH.PUSH7      ;  WAS SBI NOT BUSY?
                                    ;30803
                                    ;30804  M.CNSL: ;----------------------;  GO TO HALT LOOP
                                    ;30805          PC&VA_RC[PC.SV],        ;
                                    ;30806          CLR.NEST.ERR,           ;
U 0F1B, 0010,0038,0180,F961,2A00,039F ;30807        J/HALT.ERR              ;
                                    ;30808
                                    ;30809  .LIST           ;Re-enable full listing
```

```
                                        ;30810  .TOC    "CNSL.MIC"
                                        ;30811  .TOC    "Revision 1.2"
                                        ;30812  ;        P. R. Guilbault
                                        ;30813
;30814  .NOBIN
;30815  .TOC    "       Revision History"
;30816
;30817  ; 01    Remove absolute jumps.
;30818  ; 00    Add PLNK2.MIC and OTHER.MIC code here
;30819  ;       Start of history
;30820
                                        ;30821  .BIN
                                        ;30822  .NOLIST          ;Disable listing of PCS code for quickie assemblies
```

```
;30823  .TOC    ""      Console interface     : DESCRIPTION"'
;30824
;30825
;30826          ;MICRO-CODE ROUTINES TO SUPPORT CONSOLE SOFTWARE.
;30827          ;ROUTINES EXPECT DATA IN RXDB, AND IN ID[T1],ID[T2]
;30828          ;AND THEY RETURN DATA IN TXDB, STATUS IN ID[D.SV],
;30829          ;AND ADDITIONAL INFORMATION IN ID[T3].
;30830          ;PC IS USED WHENEVER R15 IS REFERENCED.
;30831          ;NO EFFORT IS MADE TO SAVE INTERNAL REGISTERS.
;30832
;30833          ;INFORMATION AND PARAMETERS NEEDED FROM THE CONSOLE,
;30834          ;ARE LOADED IN ID[RXDB] AND ID[T1],ID[T2].
;30835          ;RESULTING DATA IS LOADED IN ID[TXDB] AND ID[T3],
;30836          ;AND STATUS INFORMATION IS LOADED IN ID[D.SV].
;30837
;30838  ;ROUTINE:       START-ADDRESS:  PARAMETERS:(* MEANS SUPPLIED BY CONSOLE)
;30839  ;-------------------------------------------------------------------------
;30840  ;EXAMINE MEMORY 120             ID[T1]=BYTE/WORD/LONG-PARAMETER *
;30841  ;                               ID[RXDB]=VIRTUAL ADDRESS *
;30842  ;                               ID[TXDB]=MEMORY DATA
;30843  ;                               ID[T3]=PHYSICAL ADDRESS
;30844  ;                               ID[D.SV]=STATUS-CODE
;30845  ;-------------------------------------------------------------------------
;30846  ;DEPOSIT MEMORY 121             ID[T1]=BYTE/WORD/LONG-PARAMETER *
;30847  ;                               ID[RXDB]=VIRTUAL ADDRESS *
;30848  ;                               ID[T2]=MEMORY DATA *
;30849  ;                               ID[TXDB]=PHYSICAL ADDRESS
;30850  ;                               ID[D.SV]=STATUS-CODE
;30851  ;-------------------------------------------------------------------------
;30852  ;EXAM.GEN.REG.  122             ID[RXDB]=REGISTER NUMBER *
;30853  ;                               ID[TXDB]=REGISTER DATA
;30854  ;-------------------------------------------------------------------------
;30855  ;DEPO.GEN.REG.  123             ID[RXDB]=REGISTER NUMBER *
;30856  ;                               ID[T2]=REGISTER DATA *
;30857  ;-------------------------------------------------------------------------
;30858  ;EXAM.PROC.REG. 124             ID[RXDB]=REGISTER NUMBER *
;30859  ;                               ID[TXDB]=REGISTER DATA
;30860  ;-------------------------------------------------------------------------
;30861  ;DEP.PROC.REG.  125             ID[RXDB]=REGISTER NUMBER *
;30862  ;                               ID[T2]=REGISTER DATA *
;30863  ;-------------------------------------------------------------------------
;30864  ;CONTINUE       127
;30865  ;-------------------------------------------------------------------------
;30866  ;QUAD-CLEAR     129             ID[RXDB]=QUAD-ADDRESS *
;30867  ;-------------------------------------------------------------------------
;30868  ;SBI-UNJAM      12A
;30869  ;-------------------------------------------------------------------------
```

```
                                    ;30870  .TOC     ""      Console interface     : HALT AND CONTINUE''
                                    ;30871
                                    ;30872  =11110
                                    ;30873  HALT.INST:      RC[TO]_K[.10],          ;  ERROR-CODE
U 039E, 0018,0038,6580,F980,0000,08FC ;30874            J/EXCPT                     ;  HALT-INSTR. NOT IN KERNEL MODE
                                    ;30875              ;-------------------------;
                                    ;30876  HALT.ERR:
                                    ;30877              ID[D.SV]_D,             ;  SAVE ERROR-CODE
U 039F, 1000,003C,B980,3C00,0000,00FF ;30878            STOP.IB                 ;
                                    ;30879              ;-------------------------;
                                    ;30880  OFF:
                                    ;30881  HALTLOOP:
                                    ;30882              TRAP.ACC[1],            ;  TRAP THE ACCELARATOR
                                    ;30883              CID/ACK,                ;  SET CONSOLE ACKNOWLEDGE FLAG
U 00FF, 0000,00BC,0080,1400,0000,00FF ;30884            J/HALTLOOP              ;  STAY IN LOOP UNTIL MAINT. RETURN
                                    ;30885              ;-------------------------;
                                    ;30886  127:
                                    ;30887  CONS.CONT:
                                    ;30888              ;-------------------------;  ***CONTINUE***
                                    ;30889              VA_PC,LONG,             ;
                                    ;30890              FLUSH.IB,               ;  CLEAR INSTRUCTION BUFFER
                                    ;30891              AC.LOW?,                ;  TEST FOR LOW POWER
U 0127, 2C14,0E38,0180,F800,4200,0FDB ;30892            J/CONT1
                                    ;30893              ;-------------------------;
                                    ;30894  =011        ;BRANCH ON AC LOW
                                    ;30895              ;-------------------------;
                                    ;30896  CONT1:      START.IB,               ;  RESTART INSTRUCTION BUFFER
                                    ;30897              CID/CONT,
U 0FDB, 3000,1C3C,0180,1C00,0000,03DC ;30898            CONSOLE.MODE?,J/CONT3   ;  TEST COMPATIBILITY MODE
                                    ;30899              ;-------------------------;
                                    ;30900  CONT2:      INHIBIT.IB,             ;  HOLD INSTRUCTION BUFFER
                                    ;30901              LONG,AC.LOW?,           ;  TEST AC-LOW FLAG
U 0FDF, 0000,0E3C,0180,0800,0000,0FDB ;30902            J/CONT1
                                    ;30903              ;-------------------------;
                                    ;30904  =11100      ;BRANCH ON COMPATIBILITY-BIT OF PSL
                                    ;30905              ;-------------------------;
                                    ;30906  CONT3:      LOAD.IB,                ;  LOAD NEW DATA
U 03DC, 0000,003C,0180,6005,0000,0762 ;30907            PC_PC+2,J/IRD11         ;  ADJUST PC INCOMPATIBILITY MODE
                                    ;30908              ;-------------------------;
                                    ;30909              CID/CONT,               ;  CLEAR CONSOLE MODE
U 03DD, 0000,003C,0180,1C00,0000,00AB ;30910            J/IB.FILL
                                    ;30911              ;-------------------------;
                                    ;30912              CID/CONT,               ;  CLEAR CONSOLE MODE
U 03DE, 0000,003C,0180,1C00,0000,00AB ;30913            J/IB.FILL
                                    ;30914              ;-------------------------;
                                    ;30915              CID/CONT,               ;  CLEAR CONSOLE MODE
U 03DF, 0000,003C,0180,1C00,0000,00AB ;30916            J/IB.FILL               ;
```

```
                                    ;30917  .TOC    ""      Console interface     : EXAMINE MEMORY""
                                    ;30918
                                    ;30919  120:
                                    ;30920  EXAMIN: ;-------------------------------;   ***EXAMINE MAIN MEMORY***
U 0120, 0000,003C,C5F0,2C00,0000,0F20 ;30921        Q_ID[T1],LONG                   ;  GET BYTE,WORD,LONG-PARAMETER
                                    ;30922          ;-----------------------------;
                                    ;30923          D_Q,LONG,                       ;  TEST LOW ORDER BITS
                                    ;30924          Q_ID[RXDB],                     ;  GET VIRTUAL ADDRESS
U 0F20, 0C14,0038,15F0,2DE0,0000,0F21 ;30925        RC[PC.SV]_PC                    ;  SAVE PC IN RC
                                    ;30926          ;-----------------------------;
                                    ;30927          SET.NEST.ERR,                   ;  SET NESTED ERROR BIT
                                    ;30928          VA_Q,LONG,                      ;  LOAD VA WITH ADDRESS
                                    ;30929          SC_K[.9],                       ;  SHIFT-VALUE FOR PTE
U 0F21, 0001,2C3C,D980,F800,2E84,6FE4 ;30930        BEN/MUL                         ;  TEST FOR BYTE,WORD,LONG
                                    ;30931          ;-----------------------------;
                                    ;30932  =100    ;BRANCH ON LOW TWO BITS OF PARAMETER IN ID[T1]
                                    ;30933          ;00-----------------------------;   *
                                    ;30934          D[BYTE]_CACHE,                  ;  READ BYTE
                                    ;30935          ALU_K[.3FF],                    ;
                                    ;30936          RC[T0]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FE4, 0058,8038,8180,4180,0000,0948 ;30937        J/EXAM.PTE                      ;  GET PTE
                                    ;30938          ;01-----------------------------;   *
                                    ;30939          D[WORD]_CACHE,                  ;  READ WORD
                                    ;30940          ALU_K[.3FF],                    ;
                                    ;30941          RC[T0]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FE5, 0058,4038,8180,4180,0000,0948 ;30942        J/EXAM.PTE                      ;  GET PTE
                                    ;30943          ;10-----------------------------;   *
                                    ;30944          D[LONG]_CACHE,                  ;  READ A LONGWORD
                                    ;30945          ALU_K[.3FF],                    ;
                                    ;30946          RC[T0]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FE6, 0058,0038,8180,4180,0000,0948 ;30947        J/EXAM.PTE                      ;  GET PTE
                                    ;30948  =;END   ;-------------------------------;
```

```
                                ;30949  =0
                                ;30950  EXAM.PTE:
                                ;30951          ;-------------------------------;
                                ;30952          ID[TXDB]_D,LONG,                ;  SEND MEMORY DATA TO CONSOLE
                                ;30953          Q_Q.AND.RC[TO],                 ;  MASK OUT 9 LOW BITS OF VIRTUAL ADDR
                                ;30954          PC_VA,                          ;
U 0948, 0011,2035,1DC0,3D01,0000,0F22   ;30955          CALL.J/GET.PTE.CONS             ;  GET PAGE-TABLE ENTRY
                                ;30956          ;-------------------------------;
                                ;30957          PC_VA,CLR.NEST.ERR,             ;  CLEAR NESTED ERROR BIT
                                ;30958          ID[T3]_D,LONG,                  ;  SEND PHYSICAL ADDRESS TO CONSOLE
U 0949, 0000,003C,CD80,3C01,2800,0439   ;30959          J/CONS.HALT
                                ;30960          ;-------------------------------;
                                ;30961  GET.PTE.CONS:
U 0F22, 0001,203C,49F0,2D80,0000,0F23   ;30962          RC[TO]_Q,Q_ID[TBER0]            ;  SAVE LOW 9 BITS OF ADDRESS
                                ;30963          ;-------------------------------;
                                ;30964          ALU_Q.AND.K[.1],
U 0F23, 0879,2034,0580,F800,0000,0F24   ;30965          D_ALU.LEFT2                     ;  SHIFT  MM-BIT INTO BIT 2
                                ;30966          ;-------------------------------;
U 0F24, 0000,193C,15F0,2C00,0000,0FE8   ;30967          Q_ID[RXDB],D3-0?                ;  TEST IT,  GET VIRTUAL ADDRESS
                                ;30968          ;-------------------------------;
                                ;30969  =000    ;BRANCH ON LOW BIT OF TBER0
                                ;30970          ;-------------------------------;
                                ;30971          PC_VA_RC[PC.SV],                        ;  RESTORE PC
                                ;30972          D_Q,Q_ID[SBI.ERR],              ;  RETURN ORIGINAL ADDRESS IN D
U 0FE8, 0C10,003A,65F0,2D61,0200,0001   ;30973          RETURN1                         ;  MEMORY MANAGEMENT IS OFF
                                ;30974          ;-------------------------------;
U 0FEC, 0000,1C3D,0180,F800,0000,023C   ;30975  =100    CALL,PSL.MODE?,J/GET.PTE        ;  CALL CHUCK'S ROUTINE
                                ;30976          ;-------------------------------;
                                ;30977  =101    SC_K[.9],
U 0FED, 0811,0034,D9F8,F978,0084,6FEE   ;30978          D_D.AND.RC[PTE.MASK],Q_0        ;  MASK OUT FORMER MBZ-FIELD
                                ;30979          ;-------------------------------;
U 0FEE, 0D10,0038,0180,F961,0200,0F25   ;30980  =110    PC&VA_RC[PC.SV],D_DAL.SC        ;  RELOAD PC, SHIFT PTE INTO HIGH BITS
                                ;30981  =;END   ;-------------------------------;
                                ;30982          D_D.OR.RC[TO],                  ;  OR IN THE BYTE NUMBER
U 0F25, 0811,0032,65F0,2D00,0000,0001   ;30983          Q_ID[SBI.ERR],RETURN1
                                ;30984          ;-------------------------------;
```

```
                                        ;30985  .TOC    ""      Console interface     : DEPOSIT MEMORY""
                                        ;30986
                                        ;30987  121:
                                        ;30988  DEPOSIT:
                                        ;30989          ;-------------------------------;  CONSOLE DEPOSIT MEMORY ROUTINE***
U 0121, 0000,003C,C5F0,2C00,0000,0F28   ;30990          Q_ID[T1],LONG                   ;  GET STATUS DATA
                                        ;30991          ;-------------------------------;
                                        ;30992          D_Q,Q_ID[RXDB],LONG,            ;  GET VIRTUAL ADDRESS
U 0F28, 0C14,0038,15F0,2DE0,0000,0F29   ;30993          RC[PC.SV]_PC                    ;  SAVE PC
                                        ;30994          ;-------------------------------;
                                        ;30995          SET.NEST.ERR,                   ;  SET NESTED ERROR BIT
                                        ;30996          VA_Q,LONG,                      ;  LOAD ADDRESS FOR DEPOSIT
U 0F29, 0001,203C,C9F0,2C00,2E00,0F2A   ;30997          Q_ID[T2]                        ;  GET DATA
                                        ;30998          ;-------------------------------;
                                        ;30999          D_Q,LONG,                       ;  GET DATA READY FOR MEMORY
                                        ;31000          SC_K[.9],                       ;  SHIFT VALUE FOR PHYSICAL ADDRESS
U 0F2A, 0C00,0C3C,D980,F800,0084,6FF4   ;31001          BEN/MUL                         ;  TEST PARAMETER
                                        ;31002  =100
                                        ;31003          ;00-------------------------------;
                                        ;31004          CACHE_D[BYTE],                  ;  WRITE A BYTE
                                        ;31005          ALU_K[.3FF],                    ;  GET MASK-CONSTANT
                                        ;31006          RC[TO]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FF4, 0058,8038,8180,3180,0000,0950   ;31007          J/DEPPTE                        ;  GO GET PTE
                                        ;31008
                                        ;31009          ;01-------------------------------;
                                        ;31010          CACHE_D[WORD],                  ;  WRITE A WORD
                                        ;31011          ALU_K[.3FF],                    ;  GET MASK-CONSTANT
                                        ;31012          RC[TO]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FF5, 0058,4038,8180,3180,0000,0950   ;31013          J/DEPPTE                        ;  GO GET PTE
                                        ;31014
                                        ;31015          ;10-------------------------------;
                                        ;31016          CACHE_D[LONG],                  ;  WRITE A LONG
                                        ;31017          ALU_K[.3FF],                    ;  GET MASK-CONSTANT
                                        ;31018          RC[TO]_ALU.RIGHT,               ;  GET VIRTUAL ADDRESS MASK
U 0FF6, 0058,0038,8180,3180,0000,0950   ;31019          J/DEPPTE                        ;  GO GET PTE
                                        ;31020  =;END    ;-------------------------------;
                                        ;31021  =0
                                        ;31022  DEPPTE: ;-------------------------------;
                                        ;31023          Q_Q.AND.RC[TO],                 ;  GET BYTE NUMBER
                                        ;31024          PC_VA,LONG,                     ;  RELOAD PC
U 0950, 0011,2035,01C0,F901,0000,0F22   ;31025          CALL,J/GET.PTE.CONS             ;  GET PAGE TABLE ENTRY
```

```
                                        ;31026          ;RETURN HERE WITH D=PHYSICAL ADDRESS
                                        ;31027          ;-------------------------------;
                                        ;31028  DEP.PTE.0:
                                        ;31029          CLR.NEST.ERR,                   ;  CLEAR NESTED ERROR BIT
U 0951, 0000,003C,1D80,3C00,2800,0955   ;31030          ID[TXDB]_D,J/DEPOS.30           ;  SEND PHYSICAL ADDRESS TO CONSOLE
                                        ;31031          ;-------------------------------;
                                        ;31032  =0       ;BRANCH ON ALU Z-BIT
                                        ;31033          ;-------------------------------;
                                        ;31034  DEPOS.3:
                                        ;31035          ALU_D.AND.K[.8000],             ;  BIT 1=1, SO SBI IS NOT BUSY
U 0954, 0019,0034,4580,F800,0010,0F2C   ;31036          CLK.UBCC,J/DEPOS.5              ;  CLOCK CP TIME-OUT BIT
                                        ;31037          ;-------------------------------;
                                        ;31038  DEPOS.30:
                                        ;31039          ALU_Q.AND.K[.2],CLK.UBCC,       ;  CLOCK SBI BUSY BIT
U 0955, 0019,2034,0980,F800,0010,0F2B   ;31040          J/DEPOS.4
                                        ;31041          ;-------------------------------;
                                        ;31042  DEPOS.4:        Q_ID[SBI.ERR],ALU_Q,    ;  READ SBI ERROR REGISTER AGAIN
                                        ;31043          D_ALU.LEFT3,                    ;  SHIFT BIT 12 INTO POSITION 15
U 0F2B, 08A1,213C,65F0,2C00,0000,0954   ;31044          Z?,J/DEPOS.3                    ;  TEST SBI BUSY BIT
                                        ;31045          ;-------------------------------;
                                        ;31046  DEPOS.5:
U 0F2C, 0818,0138,0980,F800,0000,0958   ;31047          D_K[.2],Z?                      ;  TEST CP TIME OUT BIT
                                        ;31048          ;-------------------------------;
                                        ;31049  =0       ;BRANCH ON ALU Z-BIT
                                        ;31050          ;-------------------------------;
U 0958, 0000,003C,0180,F800,0000,039F   ;31051          J/HALT.ERR
                                        ;31052          ;-------------------------------;
U 0959, 0F00,003C,0180,F800,0000,039F   ;31053          D_0,J/HALT.ERR
                                        ;31054          ;-------------------------------;
```

```
                                        ;31055  .TOC    ""      Console interface     : EXAMINE/DEPOSIT GPR'S""
                                        ;31056
                                        ;31057  122:
                                        ;31058  EXAMGR:
                                        ;31059          ;----------------------;  ***EXAMINE R0-R14 AND PC***
                                        ;31060          Q_ID[RXDB],           ;  GET REGISTER NUMBER
U 0122, 0814,0038,15F0,2C00,0000,0F2D   ;31061          D_PC                  ;  IN CASE OF R15
                                        ;31062          ;----------------------;
                                        ;31063          ALU_Q.XOR.K[.F],BYTE, ;  CHECK FOR R15
U 0F2D, 0019,A020,6180,F800,0010,0F30   ;31064          CLK.UBCC              ;
                                        ;31065          ;----------------------;
                                        ;31066          SC_Q,LONG,            ;  USE SC AS POINTER
U 0F30, 0001,213C,0180,F800,0082,095C   ;31067          Z?                    ;  R15 ?
                                        ;31068          ;----------------------;
                                        ;31069  =0      ;BRANCH ON ALU Z-BIT
                                        ;31070          ;----------------------;
                                        ;31071          D_R(SC),LONG,         ;  READ REGISTER POINTED TO BY SC
U 095C, 0800,003C,0180,F868,0000,095D   ;31072          J/EXAM1               ;
                                        ;31073          ;----------------------;
                                        ;31074  EXAM1:  ID[TXDB]_D,           ;  TRANSMIT DATA TO CONSOLE
U 095D, 0000,003C,1D80,3C00,0000,0439   ;31075          J/CONS.HALT           ;  TRANSMIT DATA
                                        ;31076          ;----------------------;
                                        ;31077  123:
                                        ;31078  DEPOGR:
                                        ;31079          ;----------------------;  ***DEPOSIT GENERAL REGISTER***
U 0123, 0000,003C,15F0,2C00,0000,0F31   ;31080          Q_ID[RXDB],LONG       ;  GET ADDRESS-POINTER
                                        ;31081          ;----------------------;
                                        ;31082          ALU_Q.XOR.K[.F],BYTE, ;  CHECK FOR R15
U 0F31, 0019,A020,6180,F800,0010,0F32   ;31083          CLK.UBCC              ;
                                        ;31084          ;----------------------;
                                        ;31085          SC_Q,Q_ID[T2],LONG,   ;  LOAD POINTER, GET DATA
U 0F32, 0001,213C,C9F0,2C00,0082,0960   ;31086          Z?                    ;  R15?
                                        ;31087          ;----------------------;
                                        ;31088  =0      ;BRANCH ON ALU Z-BIT
                                        ;31089          ;----------------------;
                                        ;31090          R(SC)_Q,LONG,
U 0960, 0001,203C,0180,F8E8,0000,0439   ;31091          J/CONS.HALT           ;  DEPOSIT DATA
                                        ;31092          ;----------------------;
                                        ;31093          FLUSH.IB,             ;  PC CHANGES, SO FLUSH IB
                                        ;31094          PC&VA_Q,              ;  LOAD PC RATHER THAN R15
U 0961, 2001,203C,0180,F801,4200,0439   ;31095          J/CONS.HALT
                                        ;31096          ;----------------------;
```

```
                                    ;31097  .TOC    ""      Console interface     : EXAMINE/DEPOSIT IPR'S""
                                    ;31098
                                    ;31099  124:
                                    ;31100  EXAM.PROC.REG:
                                    ;31101          ;ROUTINE TO EXECUTE A MFPR FROM CONSOLE
                                    ;31102          ;-------------------------------;
U 0124, 0000,003C,15F0,2C00,0000,05C0 ;31103          Q_ID[RXDB]                      ;  GET REGISTER #
                                    ;31104          ;-------------------------------;
U 05C0, 0C00,003D,0180,F800,0000,0500 ;31105  =00***  D_Q,CALL,J/RPR                  ;  CALL MFPR-ROUTINE
                                    ;31106          ;-------------------------------;
                                    ;31107  =10***  ALU_0+K[.4]+1,D_ALU.LEFT,       ;  GENERATE CONSTANT .B
U 05D0, 083B,0010,1380,F800,0000,039F ;31108          SI/MUL-,J/HALT.ERR              ;  BAD REG #
                                    ;31109          ;-------------------------------;
U 05D8, 0C00,003C,0180,F800,0000,095D ;31110  =11***  D_Q,J/EXAM1
                                    ;31111  =;END   ;-------------------------------;
                                    ;31112  125:
                                    ;31113  DEP.PRO.REG:
                                    ;31114          ;ROUTINE TO EXECUTE A MTPR FROM CONSOLE
                                    ;31115          ;-------------------------------;
U 0125, 0000,003C,15F0,2C00,0000,0421 ;31116          Q_ID[RXDB]                      ;  GET REGISTER #
                                    ;31117          ;-------------------------------;
                                    ;31118  =00**0
U 0421, 0C00,003D,C9F0,2C00,0000,0540 ;31119  =00**1  D_Q,Q_ID[T2],CALL,J/WPR        ;  GET DATA, CALL MTPR-ROUTINE
                                    ;31120          ;-------------------------------;
                                    ;31121  =10**1  ALU_0+K[.4]+1,D_ALU.LEFT,       ;  GENERATE CONSTANT .B
U 0431, 083B,0010,1380,F800,0000,039F ;31122          SI/MUL-,J/HALT.ERR              ;  BAD REG #
                                    ;31123          ;-------------------------------;
                                    ;31124  =11**0  ;BRANCH ON ALU Z-BIT
                                    ;31125          ;-------------------------------;  PART OF UNJAM-ROUTINE
                                    ;31126  UNJH2:  D_D-K[.1],                      ;  DECREMENT COUNT
                                    ;31127          C[K.UBCC,BYTE,                  ;  CLOCK IT
                                    ;31128          MCT/SBI.HOLD,                   ;  HOLD SBI
U 0438, 0819,8100,0580,8000,0010,0438 ;31129          Z?,J/UNJH2                      ;  TEST COUNT
                                    ;31130          ;-------------------------------;
                                    ;31131  =11**1
U 0439, 0F00,003C,0180,F800,0000,039F ;31132  CONS.HALT:      D_0,J/HALT.ERR        ;  CLEAR ERROR CODE
                                    ;31133          ;-------------------------------;
```

```
                                        ;31134  .TOC     ''      Console interface     : QUAD CLEAR AND UNJAM''
                                        ;31135
                                        ;31136  129:
                                        ;31137  QUADCLR:
                                        ;31138          ;-----------------------;   ***QUAD CLEAR***
U 0129, 0000,003C,15F0,2C00,0000,0F33   ;31139          Q_ID[RXDB],LONG         ;   GET QUAD ADDRESS
                                        ;31140          ;-----------------------;
                                        ;31141          ALU_Q.ANDNOT.K[.7],     ;   STRIP LOW BITS, JUST IN CASE
                                        ;31142          VA_ALU,LONG,            ;   LOAD ADDRESS
U 0F33, 0F19,2024,5D80,F800,0200,0F35   ;31143          D_0                     ;   CLEAR DATA
                                        ;31144          ;-----------------------;
                                        ;31145          MCT/EXTWRITE.P,LONG,
U 0F35, 0000,003C,0180,A000,0000,0439   ;31146          J/CONS.HALT             ;   QUAD WRITE 0
                                        ;31147          ;-----------------------;
                                        ;31148
                                        ;31149  12A:
                                        ;31150  UNJAM:  ;-----------------------;   ***UNJAM THE SBI***
                                        ;31151          D_K[.10],CLK.UBCC,BYTE, ;   INITIALIZE COUNT
U 012A, 0818,8038,6580,F800,0010,0964   ;31152          J/UNJH1
                                        ;31153          ;-----------------------;
                                        ;31154  =0      ;BRANCH ON ALU Z-BIT
                                        ;31155  UNJH1:  ;0----------------------;
                                        ;31156          D_D-K[.1],              ;   DECREMENT COUNT
                                        ;31157          C[K.UBCC,DT/BYTE,       ;   CLOCK COUNT
                                        ;31158          MCT/SBI.HOLD,           ;   ASSERT HOLD ON SBI
                                        ;31159          Z?,                     ;   TEST COUNT
U 0964, 0819,8100,0580,8000,0010,0964   ;31160          J/UNJH1                 ;   KEEP LOOPING
                                        ;31161          ;1----------------------;
                                        ;31162          D_K[.10],CLK.UBCC,BYTE, ;   INITIALIZE COUNT
U 0965, 0818,8038,6580,8000,0010,0968   ;31163          MCT/SBI.HOLD            ;   KEEP ON HOLDING
                                        ;31164          ;-----------------------;
                                        ;31165  =0      ;BRANCH ON ALU Z-BIT
                                        ;31166  UNJU:   ;-----------------------;
                                        ;31167          D_D-K[.1],              ;   DECREMENT COUNT
                                        ;31168          C[K.UBCC,BYTE,          ;   CLOCK IT
                                        ;31169          MCT/SBI.HOLD+UNJAM,     ;   WHATEVER THAT MEANS
                                        ;31170          Z?,                     ;   TEST COUNT
U 0968, 0819,8100,0580,8800,0010,0968   ;31171          J/UNJU                  ;   KEEP LOOPING
                                        ;31172          ;-----------------------;
                                        ;31173          D_K[.10],CLK.UBCC,BYTE, ;   INITIALIZE COUNT
                                        ;31174          MCT/SBI.HOLD+UNJAM,     ;   DON'T LET UP
U 0969, 0818,8038,6580,8800,0010,0438   ;31175          J/UNJH2                 ;
```

```
                                        ;31176  .TOC    ""       Console interface     : ECO functionality test""
                                        ;31177
                                        ;31178
                                        ;31179  .LIST            ;Re-enable full listing
                                        ;31180
                                        ;31181  ; THE FOLLOWING TWO LOCATIONS ARE USED BY THE MICRO-DIAGNOSTICS TO VERIFY
                                        ;31182  ; THAT THE PLA ECO MECHANISM IS FUNCTIONAL. MICRO-DIAGNOSTICS POINTS THE UPC
                                        ;31183  ; TO 0D55, EXECUTES A SINGLE MICRO STEP, READS THE UPC AND VERIFIES THAT A
                                        ;31184  ; TRAP TO 116A OCCURED. MICRO-DIAGNOSTICS POINTS THE UPC TO 12AA, EXECUTES
                                        ;31185  ; A SINGLE MICRO STEP, READS THE UPC AND VERIFIES THAT A TRAP TO 1155 OCCURED.
                                        ;31186
                                        ;31187  0D55:   ;-------------------------------; THIS PCS ADDRESS IS ALWAYS TRAPPED
U 0D55, 0000,003C,0180,F800,0000,0D55   ;31188          J/<.>                           ; BY THE PLA'S TO 116A IN WCS
                                        ;31189
                                        ;31190  12AA:   ;-------------------------------; THIS PCS ADDRESS IS ALWAYS TRAPPED
U 12AA, 0000,003C,0180,F800,0000,12AA   ;31191          J/<.>                           ; BY THE PLA'S TO 1155 IN WCS
```

31191; This page intentionally left blank.

```
                                        ;31192  .TOC    "CALCRC.MIC"
                                        ;31193  .TOC    "Revision 0.0"
                                        ;31194  ;        P. R. Guilbault
                                        ;31195
;31196  .NOBIN
;31197  .REGION/<WCSR1L>,<WCSR1H>/<WCSR2L>,<WCSR2H>
;31198
;31199  .TOC    ""      Revision History"
;31200
;31201  ; 00    Create this file by merging CALL2 and CRC
;31202  ;               CALL2   02      Remove absolute jumps.
;31203  ;                               Add general WCS region
;31204  ;                       01      Change .REGION to simpler form
;31205  ;                       00      Start of history
;31206  ;               CRC     01      Remove absolute jumps.
;31207  ;                               Change macro that deal with conditions codes.
;31208  ;                       00      Start of history
;31209
                                        ;31210  .BIN
```

```
                                ;31211  .TOC    "       Procedure call & CRC  : CALLG, CALLS"
                                ;31212
                                ;31213  ;CALLG -- CALL INSTRUCTION FOR GENERAL ARGLIST
                                ;31214  ;ENTER WITH ARGLIST ADDRESS IN Q, DST ADDRESS IN VA AND D
                                ;31215
                                ;31216  ;CALLS -- ARGLIST IS ON STACK
                                ;31217  ;ENTER WITH DEST ADDR IN D AND VA, ARG COUNT IN Q
                                ;31218
                                ;31219  34F:
                                ;31220  CALL:   ;-------------------------------;
                                ;31221          RC[T1]_D+K[.2],D[WORD]_CACHE,   ;
U 034F, 0019,4014,0980,4188,0000,1100   ;31222          J/CA.ABS.1100                   ;
                                ;31223
                                ;31224  1100:   ;ASSIGN THIS ADDRESS BECAUSE PCS CALLS IT
                                ;31225  CA.ABS.1100:
                                ;31226          ;-------------------------------;
                                ;31227          RC[T3]_Q,                       ;SAVE ARGLST ADDRESS OR ARG CNT
U 1100, 0001,203C,21F8,F998,0084,74A4   ;31228          SC_K[.14], Q_0                  ;PREPARE TO SHIFT MASK<11:0> TO D<31:20>
                                ;31229
                                ;31230          ;-------------------------------;
                                ;31231          Q_D.OXT[WORD],                  ;CLEAR GARBAGE FROM LH OF MASK
                                ;31232          RC[T2]_ALU, SC_ALU(EXP),        ;SAVE MASK IN T2, SC<7:0>=MASK<14:7>
                                ;31233          FE_SC+K[.7C],                   ;FE=90 - 80 FOR MASK, 10 FOR SHFTCT
U 14A4, 0D03,403C,9DC0,F990,0187,94A5   ;31234          D_DAL.SC                        ;D<31:20>=MASK<11:0> FOR BIT COUNTING
                                ;31235
                                ;31236          ;-------------------------------;
                                ;31237          ID[VECTOR]_D,                   ;SEND BITS TO BE COUNTED
U 14A5, 0000,003C,3580,3C00,0080,54A6   ;31238          SC_SC.ANDNOT.FE                 ;CLEAR SC<7> (=MASK<14>)
                                ;31239
                                ;31240          ;-------------------------------;
                                ;31241          Q_ID[VECTOR],                   ;COUNT OF ONES IS IN Q<23:20>
                                ;31242          D_0,                            ;MAKE SURE ZEROS ARE SHIFTED IN
                                ;31243          SC_FE,                          ;PUT 16 IN SC FOR SHIFT(ACTUALLY 90)
U 14A6, 0F00,143C,35F0,2C00,0081,1382   ;31244          SC?                             ;TEST MASK<13:12> IN SC<6:5>
                                ;31245
                                ;31246  =*10    ;-------------------------------;MASK<13:12>=0 (SC<9:8> KNOWN 0)
                                ;31247          D_DAL.SC,                       ;COUNT OF ONES TO D<3:0>
U 1382, 0D18,0038,D5C0,F800,0010,14A8   ;31248          Q_K[.6], CLK.UBCC, J/CALL.0     ;SETUP SLOW CONST 6 IN CASE NEEDED
                                ;31249
                                ;31250          ;-------------------------------;MASK<13:12> .NE. 0
U 1383, 0000,003C,0180,F800,0000,0106   ;31251          J/RSVOPR                        ;RESERVED OP, MASK <13:12> NOT ZERO
                                ;31252
                                ;31253          ;-------------------------------;
                                ;31254  CALL.0: D_D.AND.K[.F], LC_RC[T2],       ;STRIP BIT COUNT, LATCH MASK FOR CALLG
U 14A8, 0819,1B34,6180,F910,0000,1391   ;31255          IR0?                            ;IS THIS CALLS OR CALLG?
                                ;31256
                                ;31257  =**01   ;-------------------------------; (ALU.N=ALU.Z=0)
                                ;31258          ALU_D+K[.4]+1,                  ;# OF ONES IN MASK, +5 GIVES # OF PUSHES
                                ;31259          D_ALU.LEFT2,                    ;CONVERT TO BYTE COUNT
U 1391, 0879,0010,1180,FA70,0000,14AD   ;31260          LAB_R[SP],J/CALL.5              ;BRING UP SP FOR BASE ADDRESS
                                ;31261
                                ;31262          ;-------------------------------;
                                ;31263          ALU_D+Q,                        ;CALLS DOES 6 PUSHES
                                ;31264          D_ALU.LEFT2,                    ;CONVERT TOTAL PUSHES TO BYTE COUNT
U 1393, 087D,0014,0180,FA70,0000,14AA   ;31265          LAB_R[SP]                       ;PICK UP STACK POINTER
```

```
                                ;31266  ;HERE ON CALLS, WITH D CONTAINING THE NUMBER OF BYTES TO BE PUSHED
                                ;31267
                                ;31268          ;--------------------------------;
                                ;31269          VA_LA-D,                         ;SETUP LAST PUSH ADDR
                                ;31270          RC[T4]_ALU.LEFT,                 ;SAVE SP<1:0>
U 14AA, 003C,2000,0380,F9A0,0200,14AB   ;31271          SI/MUL-                          ; WITH CALLS FLAG
                                ;31272
                                ;31273          ;--------------------------------;
                                ;31274          Q_RC[T3], SC_K[ZERO],            ;GET ARG COUNT, CLR SC FOR CONSTRAINT
U 14AB, 0010,0038,19C0,1118,0084,74AC   ;31275          TEST.TB.WCHK                     ;ASK TB ABOUT ACCESSIBILITY
                                ;31276
                                ;31277          ;--------------------------------;
                                ;31278          R[SP]_LA-K[.4].RLOG,LONG,        ;READY TO PUSH ARG COUNT
                                ;31279          D_ALU,                           ;SAVE IN D FOR ALIGNMENT CHECK
                                ;31280          INHIBIT.IB,                      ;DON'T LET PAGE BOUNDARY CONFUSE THINGS
U 14AC, 0818,1D04,1180,0AF0,0000,1019   ;31281          TB.TEST?                         ;WHAT DOES TB SAY?
                                ;31282
                                ;31283  =011*01 ;--------------------------------;TB HIT AND NOT ACCESS VIOLATION
                                ;31284  CALL.1: VA_D, RC[T3]_D,                  ;GET UPDATED SP AGAIN, SAVE AS ARGLIST ADR
                                ;31285          FE_K[.1F], D_Q, Q_D,             ;GET INIT REG # FOR PUSHM, D=ARGCNT, Q=SP
U 1019, 0C01,0C3C,8DE0,F998,0304,7398   ;31286          MUL?, J/CALL.2                   ;TEST ALIGNMENT AND GO STORE ARG COUNT
                                ;31287
                                ;31288          ;--------------------------------;TB MISS OR ACCESS VIOLATION
                                ;31289          RC[PC.SV]_PC,PC_VA,              ;GO REFILL TB AND VALIDATE ACCESS
                                ;31290          ID[D.SV]_D,D_Q,                  ;START BY SAVING REGISTERS
U 101B, 0C14,0039,B980,3DE1,0000,0D12   ;31291          CALL,J/TBF.W
                                ;31292
                                ;31293  =111*11 ;--------------------------------;RETURN HERE IF ACCESS ALLOWED
                                ;31294          D_Q,Q_ID[Q.SV],                  ;FINISH RESTORING REGISTERS
U 103B, 0C00,003C,BDF0,2C00,0000,1019   ;31295          J7CAL[.1                         ;REJOIN NORMAL FLOWS
                                ;31296
                                ;31297  ;HERE AFTER PROBING LAST ADDRESS TO BE PUSHED
                                ;31298  ;STORE ARG COUNT, ALIGN SP, AND BEGIN REGISTER SAVE
                                ;31299
                                ;31300  =*00    ;--------------------------------; SP ALIGNED (SC=0)
                                ;31301  CALL.2: CACHE_D[LONG],                   ;STORE ARG COUNT
                                ;31302          Q_K[.FFFF].RIGHT2, SI/ZERO,      ;GET MASK FOR REGISTER MASK
U 1398, 0098,0038,C1C0,3270,0081,12CC   ;31303          LAB_R[SP], SC_FE, J/CALL.7       ;SET SC(=1F) AND LA FOR PUSHM
                                ;31304
                                ;31305          ;--------------------------------;SP NOT ALIGNED
U 1399, 0019,2004,0580,FAF0,0000,1398   ;31306          R[SP]_Q-K[.1].RLOG, J/CALL.2     ;ALIGN STACK POINTER
                                ;31307
                                ;31308          ;--------------------------------;SP NOT ALIGNED
U 139A, 0019,2004,0980,FAF0,0000,1398   ;31309          R[SP]_Q-K[.2].RLOG, J/CALL.2     ;ALIGN STACK POINTER
                                ;31310
                                ;31311          ;--------------------------------;SP NOT ALIGNED
U 139B, 0019,2004,0D80,FAF0,0000,1398   ;31312          R[SP]_Q-K[.3].RLOG, J/CALL.2     ;ALIGN STACK POINTER
```

```
                                ;31313  ;HERE IN CALLG, WITH D CONTAINING THE NUMBER OF BYTES TO BE PUSHED
                                ;31314
                                ;31315          ;--------------------------------;
                                ;31316  CALL.5: VA_LA-D, SC_ALU,                  ;ADDR OF LAST ITEM TO BE PUSHED
                                ;31317          FE_K[.FFFC],                    ;GET MASK FOR ISOLATING SP<1:0> IN SC
U 14AD, 003C,2000,F180,F9A0,0386,74AE   ;31318          RC[T4]_ALU.LEFT,SI/ZERO         ;SAVE SP<1:0>
                                ;31319
                                ;31320          ;--------------------------------;
                                ;31321          TEST.TB.WCHK,                   ;CHECK WRITABILITY
                                ;31322          SC_SC.ANDNOT.FE,                ;SC NOW HAS SP<1:0>
                                ;31323          LAB_R[SP],                      ;GET SP FOR ALIGNING
U 14AE, 0098,0038,C1C0,1270,0080,52C4   ;31324          Q_K[.FFFF].RIGHT2, SI/ZERO      ;GET MASK FOR REGISTER MASK
                                ;31325
                                ;31326  =0      ;0--------------------------------;
                                ;31327          R[SP]_LA-K[SC].RLOG,            ;ALIGN SP WHETHER IT NEEDS IT OR NOT
                                ;31328          D_Q, INHIBIT.IB,                ;D = FFF IN CASE WE CALL TBF.W
U 12C4, 0C18,1D05,1D80,0AF0,0000,1059   ;31329          TB.TEST?, CALL, J/CALL.6        ;TEST STACK SPACE, CALL PUSH ROUTINE
                                ;31330
                                ;31331          ;1--------------------------------;RETURN FROM MPUSH
                                ;31332          CACHE_D[LONG],                  ;WRITE PC ONTO STACK
U 12C5, 0000,003C,0180,3270,0000,14B0   ;31333          LAB_R[SP], J/CALL.8             ;JOIN CALLS CODE
                                ;31334
                                ;31335  =011*01 ;--------------------------------;TB HIT AND NO ACCESS VIOLATION
                                ;31336  CALL.6: LAB_R[SP], SC_K[.1F],           ;SET UP FOR PUSHM AND GO TO IT
U 1059, 0811,2034,8D80,FA70,0084,6CD8   ;31337          D_Q.AND.LC, J/MPUSH
                                ;31338
                                ;31339          ;--------------------------------;TB MISS OR ACCESS VIOLATION
                                ;31340          RC[PC_SV]_PC,PC_VA,             ;SAVE REGISTERS
                                ;31341          ID[D.SV]_D,D_Q,                 ; IN RC AND ID TEMPS
U 105B, 0C14,0039,B980,3DE1,0000,0D12   ;31342          CALL,J/TBF.W                    ;WHILE REFILLING TB FOR WRITE
                                ;31343
                                ;31344  =111*11 ;--------------------------------;RETURN HERE AFTER ACCESS VERIFIED
U 107B, 0000,003C,0180,F910,0000,1052   ;31345          LC_[T2], J/CALL.6               ;RESTORE LC AND GO PUSH REGS
```

```
                                        ;31346  ;HERE FOR CALLG OR CALLS, AFTER PROBING THE EXTENT OF THE STACK
                                        ;31347
                                        ;31348  =0      ;-------------------------------;CALL SITE FOR MPUSH
                                        ;31349  CALL.7: D_Q.AND.RC[T2],                 ;STRIP MASK TO BITS 11-0
U 12CC, 0811,2035,0180,F910,0000,0CD8   ;31350          CALL,J/MPUSH                    ;PUSH REGISTERS
                                        ;31351
                                        ;31352          ;-------------------------------;RETURN FROM MPUSH
                                        ;31353          CACHE_D[LONG],                  ;PUSH PC
U 12CD, 0000,003C,0180,3270,0000,14B0   ;31354          LAB_R[SP]                       ; BY SP
                                        ;31355
                                        ;31356          ;-------------------------------;
U 14B0, 0018,0000,0180,FAF0,0200,14B1   ;31357  CALL.8: R[SP]&VA_LA-K[.8]                ;UPDATE SP FOR PUSH OF PC & FP
                                        ;31358
                                        ;31359          ;-------------------------------;
U 14B1, 0800,003C,0180,FA68,0000,12E0   ;31360          D_R[FP]                         ;READY TO PUSH FRAME POINTER
                                        ;31361
                                        ;31362  =0      ;-------------------------------;CALL SITE FOR PSHSP
                                        ;31363          CACHE_D[LONG],                  ;STORE FP,
                                        ;31364          LAB_R[SP],                      ; GET SP AGAIN
                                        ;31365          SC_K[.FFF0],                    ;-16 TO SC
U 12E0, 0000,003D,6D80,3270,0084,6CD9   ;31366          CALL,J/PSHSP
                                        ;31367
                                        ;31368          ;-------------------------------;
                                        ;31369          D_R[AP],                        ;READY TO PUSH AP
U 12E1, 0800,003C,3DF0,2E60,0000,14B2   ;31370          Q_ID[PSL]                       ; AND GET PSW FOR COMBINATION WITH MASK
                                        ;31371
                                        ;31372          ;-------------------------------;
                                        ;31373          CACHE_D[LONG],                  ;STORE OLD AP
                                        ;31374          Q_Q.ANDNOT.K[.1F],              ;CLEAR PSW<T,N,Z,V,C>
U 14B2, 0019,2024,8DC0,3270,0000,14B3   ;31375          LAB_R[SP]                       ;GET SP INTO LATCHES AGAIN
                                        ;31376
                                        ;31377          ;-------------------------------;
U 14B3, 2010,0038,0180,F909,4200,14B4   ;31378          PC&VA_RC[T1], FLUSH.IB          ; LOAD NEW PC AND CLEAR OUT IB
                                        ;31379
                                        ;31380          ;-------------------------------;
                                        ;31381          D_DAL.SC,                       ;PSW TO D<31:16>
                                        ;31382          Q_RC[T2],                       ;RECOVER MASK
                                        ;31383          SC_SC+K[.3],                    ;PUT -13 IN SC
U 14B4, 0D10,0038,0DC0,6114,0084,94B5   ;31384          LOAD.IB, PC_PC+1                ;START FETCHING SUBROUTINE I-STREAM
                                        ;31385
                                        ;31386          ;-------------------------------;
                                        ;31387          D_DAL.SC,                       ;MASK AND PSW IN D<31:03>
                                        ;31388          Q_RC[T4],                       ;GET LOW BITS OF OLD SP TO Q<1:0>
U 14B5, 0D10,0038,F5C0,F920,0084,94B8   ;31389          SC_SC+K[.A]                     ;PUT -3 IN SC
```

```
                                        ;31390  ;HERE TO FINISH CALL
                                        ;31391
                                        ;31392          ;--------------------------------;
                                        ;31393          D_DAL.SC,                       ;MASK/PSW LONGWORD NOW IN D
U 14B8, 0D18,0000,11C0,FAF0,0200,14B9   ;31394          R[SP]&VA_LA-K[.4], Q_ALU        ;PUSH SP, SAVE NEW VALUE IN Q
                                        ;31395
                                        ;31396          ;--------------------------------;
                                        ;31397          CACHE_D[LONG],                  ;PUSH MASK, PSW
                                        ;31398          ALU_RC[T2], SC_ALU(EXP),        ;GET ENTRY MASK, SC<7>=MASK<14>,
                                        ;31399          SS_ALU15,                       ;SS = MASK<15>
U 14B9, 0010,0038,5981,3110,0197,74BA   ;31400          FE_K[.7F], CLK.UBCC             ;GET MASK TO ISOLATE SC<7>, CLR EALU.CC
                                        ;31401
                                        ;31402          ;--------------------------------;
                                        ;31403          R[SP]&VA_Q-K[.4],               ;SETUP TO PUSH THE ON-CONDITION MASK (0)
                                        ;31404          Q_ALU,                          ;THIS ADDRESS IS BASE OF STACK FRAME
U 14BA, 0019,2000,11C0,FAF0,0280,54BB   ;31405          SC_SC.ANDNOT.FE                 ;ISOLATE MASK<14> IN SC<7>
                                        ;31406
                                        ;31407          ;--------------------------------;
                                        ;31408          R[FP]_Q,                        ;SET UP STACK FRAME POINTER
U 14BB, 00C1,203C,3DF0,2EE8,0000,14BC   ;31409          Q_ID[PSL]                       ;GET PSL FOR TRAP ENABLE SETTINGS
                                        ;31410
                                        ;31411          ;--------------------------------;
                                        ;31412          D_Q.ANDNOT.K[.EF],              ;PRESERVE T-BIT AND PSL<31:8>
U 14BC, 0819,3224,3D80,F918,0000,13A8   ;31413          LC_RC[T3], EALU?                ;GET ARGLIST ADDR, TEST MASK<15:14>
                                        ;31414
                                        ;31415  =**00   ;00-------------------------------; MASK<15:14>=00 (EALU.N=EALU.Z=0)
                                        ;31416  CALL.9: R[AP]_LC, ID[PSL]_D,            ;SAVE ARG PTR AND NEW PSL
U 13A8, 0F10,0038,3D80,3EE0,0000,14BE   ;31417          D_0, J/CALL.E                   ;PREPARE TO SAVE 0 AS ON-CONDITION WD
                                        ;31418
                                        ;31419          ;01-------------------------------;MASK<15:14>=10
U 13A9, 0819,0030,4180,F800,0000,13A8   ;31420          D_D.OR.K[.80], J/CALL.9         ;SET DECIMAL OVERFLOW ENABLE
                                        ;31421
                                        ;31422          ;10-------------------------------;MASK<15:14>=01
U 13AA, 0819,0030,7580,F800,0000,13A8   ;31423          D_D.OR.K[.20], J/CALL.9         ;SET INTEGER OVERFLOW ENABLE
                                        ;31424
                                        ;31425          ;11-------------------------------;MASK<15:14>=11
U 13AB, 0819,0030,2580,F800,0000,13A8   ;31426          D_D.OR.K[.A0], J/CALL.9         ;SET BOTH ENABLES
                                        ;31427
                                        ;31428          ;--------------------------------;
U 14BE, 0000,003C,0180,3000,0000,0062   ;31429  CALL.E: CACHE_D[LONG], J/IRD             ;WRITE THE ZERO AND GET OUT
```

```
;31430  .TOC    ""      Procedure call & CRC  : CRC Description"
;31431
;31432  ;       CRC     TBLADDR,INICRC,LENGTH,STREAMADDR
;31433
;31434  ;INPUTS:
;31435  ; D     INITIAL CRC
;31436  ; Q     TABLE ADDRESS
;31437
;31438  ;REGISTER USAGE:
;31439  ; R0    LENGTH; IF FAULTS OR INTERRUPTS OCCUR, R0 CONTAINS
;31440  ;       LENGTH IN <15:0>, PC DELTA IN <23:16>, STATE IN <31:23>
;31441  ; R1    CRC
;31442  ; R2    TABLE ADDR
;31443  ; R3    STREAM ADDRESS
;31444
;31445  ; ID(T0) USED TO HOLD CRC AFTER EACH PASS THROUGH OUTER LOOP
;31446  ; STATE IS NOT USED
;31447  ; FE    IS USED TO DISTINGUISH INTERRUPTS FROM PAGE FAULTS FOR
;31448  ;       SUSPENSION LOGIC (0 = INTERRUPT)
;31449
;31450  ;OUTPUTS:
;31451  ; R0    CRC
;31452  ; R1    0
;31453  ; R2    0
;31454  ; R3    END OF STREAM + 1
```

```
                                        ;31455  .TOC    "       Procedure call & CRC  : CRC Entry"
                                        ;31456
                                        ;31457  ;CRC GET ARGUMENTS ROUTINE
                                        ;31458
                                        ;31459  48B:
                                        ;31460  CRC:    ;-------------------------------;CRC ENTRY
U 048B, 0001,203D,0180,F980,0000,037E   ;31461          RC[TO]_Q,CALL,J/SPEC            ;
                                        ;31462
                                        ;31463  49B:    ;-------------------------------;
                                        ;31464          D_D.OXT[WORD],STATE_K[.80],     ;CLEAN EXTRA BITS OF LENGTH
                                        ;31465          STATE_K[.80],FE_K[.80],         ;
U 049B, 0803,403C,4180,F800,1504,7029   ;31466          J/CR.ABS.1029                   ;
                                        ;31467
                                        ;31468  1029:   ;ASSIGN THIS ADDRESS BECAUSE PCS CALLS IT
                                        ;31469  CR.ABS.1029:
                                        ;31470          ;01*****-----------------------;SAVE LENGTH, GET STREAM ADDR
U 1029, 0C00,003D,C580,3C00,0000,047E   ;31471          ID[T1]_D, D_Q, CALL[ASPC]       ;SAVE INITIAL CRC ACROSS ASPC
                                        ;31472
                                        ;31473  1069:   ;ASSIGN THIS ADDRESS BECAUSE OF CONSTRAINT ON PREVIOUS INSTRUCTION
                                        ;31474          ;11*****-----------------------;
U 1069, 0C01,003C,0180,FA98,0000,14C0   ;31475          R[R3]_D,D_Q                     ;SAVE STREAM ADDR. COPY INICRC
                                        ;31476
                                        ;31477          ;-------------------------------;
U 14C0, 0000,003C,C5F0,2D00,0000,14C4   ;31478          LC_RC[TO], Q_ID[T1]             ;SAVE STREAM, GET TBLADR & LENGTH
                                        ;31479
                                        ;31480          ;-------------------------------;
U 14C4, 0010,0038,0180,FA90,0000,13B8   ;31481          R[R2]_LC                        ;SAVE TBL ADDR
```

```
                                ;31482  .TOC     "       Procedure call & CRC  : CRC Loop setup and loop"
                                ;31483
                                ;31484  =00     ;--------------------------------;CALL CONSTRAINT BLOCK FOR FPDSETUP
                                ;31485  CRC.3:  R[R0]_Q.AND.K[.FFFF], CLK.UBCC, ;SAVE LOOP COUNT, SET FAULT VECTOR
U 13B8, 0019,2035,C180,FA80,0114,6E16 ;31486    FE_K[.FFFF], CALL[SETFPD]       ;SET FAULT/INT FLAG TO FAULT
                                ;31487
                                ;31488          ;--------------------------------;
U 13B9, 0000,003C,C1F0,2C00,0000,14D0 ;31489    Q_ID[TO], J/CRCSUSPND           ;WRITE FAULT - Q=CRC
                                ;31490
                                ;31491          ;--------------------------------;
U 13BA, 0000,003C,C1F0,2C00,C000,14D0 ;31492    Q_ID[TO], J/CRCSUSPND           ;READ FAULT - Q=CRC,
                                ;31493
                                ;31494          ;--------------------------------;
                                ;31495  CRCLOOP:
                                ;31496          LAB_R[R3], VA_LA, ID[TO]_D,     ;GET STREAM ADDR, SAVE CRC
                                ;31497          Q_D,                            ;COPY INICRC
U 13BB, 0000,013C,C1E0,3E18,4200,12EC ;31498    INTRPT.STROBE, Z?               ;TEST FOR STREAM EXHAUSTED
                                ;31499
                                ;31500  =C      ;--------------------------------;BRANCH ON Z (STREAM IS EXHAUSTED)
                                ;31501          D[BYTE]_CACHE, INT?,            ;MORE TO GO - GET A BYTE
                                ;31502          SC_K[.FFFF],                    ;IF THERE'S AN INTERRUPT, WE'LL
                                ;31503                                          ;NEED TO AND FE
U 12EC, 0000,8E3C,C180,4000,0084,73C6 ;31504    J/CRCXORBYTE
                                ;31505
                                ;31506          ;1-------------------------------;
                                ;31507          ALU_0+D,
                                ;31508          R[R0]_ALU,N&Z_ALU.V&C_0,        ;SAVE RESULT, SET CC<N&Z>, CLEAR <C>
U 12ED, 001F,2014,0180,FA80,0050,14CE ;31509    J/CRCEXIT
                                ;31510
                                ;31511  =110    ;--------------------------------;BRANCH ON INTERRUPTS (PENDING)
                                ;31512  CRCXORBYTE:
                                ;31513          D_D.OXT[BYTE].XOR.Q, Q_ALU,     ;.XOR. THE BYTE INTO THE CRC
U 13C6, 081F,8020,C1C0,F800,0084,7250 ;31514    SC_K[.FFFF], J/CRCILOOP         ;SET COUNT AND GO TO INNER LOOP
                                ;31515
                                ;31516          ;111-------------------------------;
                                ;31517          FE_SC.ANDNOT.FE,                ;INTERRUPT PENDING - SET FE=0
U 13C7, 0000,003C,C1F0,2C00,0100,54D0 ;31518    Q_ID[TO], J/CRCSUSPND           ;TO INDICATE INTERRUPT & GO AWAY
```

```
                                ;31519  =*0*    ;-------------------------------;BRANCH ON SC .GT. 0 (INNER LOOP EXHAUSTED)
                                ;31520  CRCILOOP:
                                ;31521          ALU_D.AND.K[.F], D_ALU.LEFT2,   ;FORM LONGWORD INDEX FROM D<3:0>
                                ;31522          Q_Q.RIGHT2, SI/ZERO,            ;SHIFT Q THE FIRST 2 OF 4 PLACES
U 1250, 0879,0034,6190,F890,0000,14C9   ;31523          LA_RA[R2], J/CRCRDTBL
                                ;31524
                                ;31525          ;*1*-------------------- -------;
U 1252, 000F,0010,0180,FA98,0000,14C6   ;31526          R[R3]_0+LB+1                    ;END OF INNER LOOP - UPDATE STREAM PTR
                                ;31527
                                ;31528          ;-------------------------------;
                                ;31529          R[R0]_LA-K[.1], CLK.UBCC,       ;DECREMENT LOOP COUNT
U 14C6, 0018,0000,0580,FA80,0010,1388   ;31530          J/CRCLOOP                       ;AND LOOP
                                ;31531
                                ;31532          ;-------------------------------;
                                ;31533  CRCRDTBL:
U 14C9, 001C,2014,0190,F800,0200,14CA   ;31534          VA_LA+D, Q_Q.RIGHT2, SI/ZERO    ;INDEX INTO TABLE, COMPLETE Q SHIFT.
                                ;31535
                                ;31536          ;-------------------------------;
U 14CA, 0000,003C,0180,4000,0080,D4CB   ;31537          D[LONG]_CACHE, SC_SC+1          ;GET TABLE ENTRY, BUMP INNER LOOPCT
                                ;31538
                                ;31539          ;-------------------------------;
                                ;31540          Q_D.XOR.Q, D_ALU, LA_RA[R0],    ;XOR TABLE ENTRY INTO CRC,
                                ;31541          SC.GT.0?,                       ;TEST INNER LOOP COUNT,
U 14CB, 081D,1420,01C0,F880,0000,1250   ;31542          J/CRCILOOP                      ;AND LOOP TILL IT OVERFLOWS (2 TIMES)
                                ;31543
```

```
                                    ;31544  .TOC    "       Procedure call & CRC  : CRC Termination and FPD"
                                    ;31545
                                    ;31546  ;       CRC LOOP TERMINATION COMES HERE
                                    ;31547
                                    ;31548          ;-------------------------------;
                                    ;31549  CRCEXIT:
                                    ;31550          R[R1] K[ZERO],
                                    ;31551          CLR.SD&SS,                      ;NO STORAGE TO REGISTERS
U 14CE, 0018,0038,1987,FA88,0000,09F1 ;31552        J/R2ZERO
                                    ;31553
                                    ;31554          ;-------------------------------;
                                    ;31555
                                    ;31556
                                    ;31557  ;       CRC FAULT AND INTERRUPT CODE COMES HERE
                                    ;31558
                                    ;31559          ;-------------------------------;
                                    ;31560  CRCSUSPND:
U 14D0, 0001,203C,0180,FA88,0000,06D8 ;31561        R[R1]_Q, J/FPDPACK              ;SAVE CRC, PACK R0 AND GO AWAY
                                    ;31562
                                    ;31563          ;-------------------------------;
                                    ;31564
                                    ;31565  ;       CRC INSTRUCTION RESTART COMES HERE FROM COMMON CODE WITH EDIT&MATCHC
                                    ;31566  ;       WITH D = COUNT, STATE AND PC ALREADY SET UP
                                    ;31567
                                    ;31568          ;-------------------------------;
                                    ;31569  1010:                                  ;THIS ADDRESS IS ASSIGNED IN PCS
                                    ;31570  CRCUNSCRAMBLE:
U 1010, 0800,003C,01E0,FA08,0000,13B8 ;31571        D_R[R1], Q_D, J/CRC.3           ;D = CRC, Q = COUNT, RE-ENTER CODE
                                    ;31572          ;-------------------------------;
                                    ;31573
                                    ;31574
                                    ;31575  ;       END OF CRC
```

31575; This page intentionally left blank.

```
                                        ;31576  .TOC    "FPA.MIC"
                                        ;31577  .TOC    "Revision 1.8"
                                        ;31578  ;        Kevin J. Cassidy, P. R. Guilbault
                                        ;31579
                                        ;31580
;31581  .NOBIN
;31582  .TOC    "       Revision History"
;31583  .LIST
;31584
;31585  ; 01    Fix POLYF/D FPD back up general regs before allowing interrupt
;31586  ;       Install faults in place of traps.
;31587  ;       Change macro names that deal with conditions codes.
;31588  ;       Fix POLYD rounding problem with negative fractions. Rounded by 2 instead of 1.
;31589  ; 00    Add FPA2.MIC code here
;31590  ;       Start of history
;31591
                                        ;31592  .BIN
```

```
;31593  .TOC     "      FPA Interface          : Description"
;31594
;31595          ;*******************************;
;31596          ;                               ;
;31597          ;  FLOATING POINT ACCELERATOR  ;
;31598          ;  -------------------------    ;
;31599          ;     CP INTERFACE FIRMWARE     ;
;31600          ;     ------------------        ;
;31601          ;                               ;
;31602          ;*******************************;
;31603
;31604          ;THESE ROUTINES ARE THE FIRMWARE INTERFACE TO THE
;31605          ;FLOATING POINT ACCELERATOR (FPA).  THEY  DETERMINE THE
;31606          ;ADDRESS FOR THE DST OPERAND IF THREE ADDRESS TYPE, ACCEPT RESULTS
;31607          ;DATA FROM THE FPA, STORE THE RESULTS, AND HANDLE
;31608          ;ANY EXCEPTIONS FLAGGED BY THE FPA.
;31609
;31610          ;THERE ARE FOUR GROUPS OF ROUTINES WHICH ARE
;31611
;31612          ;       1)  GENERALIZED FPA F/W
;31613          ;           USED TO HANDLE ADD(F/D), SUB(F/D), MUL(F/D), DIV(F/D)
;31614          ;           AND EXCEPTION HANDLING ROUTINES
;31615
;31616          ;       2)  POLY(F/D) ROUTINES
;31617
;31618          ;       3)  MULL ROUTINES
;31619
;31620          ;       4)  EMOD(F/D) ROUTINES
;31621
;31622          ;SPECIFICS ON ASSUMPTIONS ON EACH ROUTINE ARE INCLUDED
;31623          ;IN ITS HEADING.
;31624
;31625          ;THESE ROUTINES ARE ALL LOADED INTO WCS WITH UPC<12>=1
;31626          ;FORCED BY THE ACCELERATOR WHEN IT DETECTS AN EXECUTION
;31627          ;OR  OPTIMIZATION EXIT OFF OF A-FORK, B-FORK, OR C-FORK
;31628          ;FOR AN INSTRUCTION THAT IS IMPLEMENTED BY IT.
;31629
;31630          ;INTERFACE SYNCHRONIZATION
;31631          ;-------------------------------------
;31632          ;THE FOLLOWING MACRO'S ARE USED TO CO-ORDINATE THE INTERFACE
;31633          ;BETWEEN THE FPA MICRO-CODE AND THE CP MICRO-CODE:
;31634          ;
;31635          ;       1)  "ID_D.SYNC"
;31636          ;           GATES THE D-REG'S CONTENTS ONTO THE ID-BUSS AND SIGNALS
;31637          ;           THE FPA (USING CP SYNC) THAT THE DATA IS BEING PRESENTED
;31638          ;           DURING THE CURRENT CP CYCLE.
;31639
;31640          ;       2)  "D_ACCEL&SYNC" AND "Q_ACCEL&SYNC"
;31641          ;           GATES DATA BEING PRESENTED BY THE FPA ON THE DFMX BUSS
;31642          ;           ONTO EITHER THE D-REG OR THE Q-REG AS SPECIFIED BY THE
;31643          ;           MACRO, AND SIGNALS THE FPA THAT THE FPA'S DATA HAS BEEN
;31644          ;           ACCEPTED.
```

```
;31645          ;       3)  "ACCEL?"  (BEN/ACC<UB2,UB1,UB0>)
;31646          ;           THIS MACRO IN COMBINATION WITH THE TARGET CONSTRAINT
;31647          ;           BLOCK ALLOWS THE CP MICRO-CODE TO TEST FOR:
;31648          ;
;31649          ;               FPA SYNC
;31650          ;               ========
;31651          ;               ACC<UB0> = 1, RESULT DATA, ON THE DFMX, AND CONDITION
;31652          ;               CODES ARE BEING PRESENTED BY THE FPA.   IF DOUBLE
;31653          ;               PRECISION, CONDITION CODES PASSED WITH FIRST HALF.
;31654
;31655          ;               ERROR SYNC
;31656          ;               ==========
;31657          ;               ACC<UB1> = 1; SIGNALS CP MICRO-CODE THAT AN EXCEPTION
;31658          ;               HAS BEEN DETECTED BY THE FPA
;31659
;31660          ;               NOT MULL
;31661          ;               ========
;31662          ;               ACC<UB2> = 1;  SIGNAL PROVIDED BY FPA TO SEPARATE
;31663          ;               MULL FROM MULF
;31664
;31665          ;               RESERVED OPERAND OR DIVIDE CHECK (ALL EXCEPT POLY)
;31666          ;               ================================
;31667          ;               WHEN RESULT DATA IS EXPECTED BY THE CP MICRO-CODE
;31668          ;               'ERROR SYNC' AND NOT 'FPA SYNC' SIGNALS THE CP MICRO-
;31669          ;               CODE THAT ONE OF THESE HAVE BEEN ENCOUNTERED BY THE
;31670          ;               FPA.  THE PSW<V> BIT IS TESTED BY THE CP MICRO-CODE TO
;31671          ;               FIND OUT WHICH ONE OCCURRED (= 1 IF DIVIDE CK)
;31672
;31673          ;               ON POLYF/D, THE ACCELERATOR CONTROL/STATUS REG MUST
;31674          ;               BE EXAMINED TO DETERMINE IF A RESERVED OPERAND IS
;31675          ;               BEING REPORTED (INSTEAD OF UNDERFLOW)
;31676
;31677          ;       4)  "POLY.DONE"
;31678          ;           USED BY CP MICRO-CODE TO SIGNAL THE FPA MICRO-CODE
;31679          ;           THAT THE LAST COEFFICIENT FOR POLYF/D IS BEING PRESENTED.
;31680          ;           (ON POLYD, USED ON BOTH HALVES OF LAST COEFFICIENT).
;31681
;31682          ;       5)  "TRAP.ACC[1]"
;31683          ;           USED BY CP MICRO-CODE TO ABORT FPA MICRO-CODE TO THE
;31684          ;           IRD STATE.
;31685
;31686          ;       6)  "MSC/LOAD.ACC.CC"
;31687          ;           USED TO LOAD PSW<N,Z,V,C> WITH FPA GENERATED CONDITION
;31688          ;           CODES FROM LATCHES IN CP LOADED IN PREVIOUS CYCLE
```

```
B  1  Decimal string          : FAULT PARAMETER SAVE-ROUTINES
C  1  RPPR.MIC
D  1  RET, PUSHR, and POPR    : RET
E  1  RET, PUSHR, and POPR    : RET
F  1  RET, PUSHR, and POPR    : RET
G  1  RET, PUSHR, and POPR    : PUSHR
H  1  RET, PUSHR, and POPR    : POPR
I  1  RET, PUSHR, and POPR    : POPR
J  1  RET, PUSHR, and POPR    : POPR
K  1  RET, PUSHR, and POPR    : POPR
L  1  RET, PUSHR, and POPR    : POPR
M  1  REI.MIC
N  1  REI instruction         : REI
B  2  REI instruction         : REI
C  2  REI instruction         : REI
D  2  REI instruction         : REI
E  2  REI instruction         : REI
F  2  QUEUE.MIC
G  2  Queue instructions      : INSQUE
H  2  Queue instructions      : REMQUE
I  2  Queue instructions      : REMQUE
J  2  Queue instructions      : INSQHI, INSQTI
K  2  Queue instructions      : INSQHI, INSQTI
L  2  Queue instructions      : INSQHI, INSQTI
M  2  Queue instructions      : INSQHI, INSQTI
N  2  Queue instructions      : INSQHI, INSQTI
B  3  Queue instructions      : INSQHI, INSQTI
C  3  Queue instructions      : INSQHI, INSQTI
D  3  Queue instructions      : REMQHI, REMQTI
E  3  Queue instructions      : REMQHI, REMQTI
F  3  Queue instructions      : REMQHI, REMQTI
G  3  Queue instructions      : REMQHI, REMQTI
H  3  Queue instructions      : REMQHI, REMQTI
I  3  PROBE.MIC
J  3  Probe instruction       : PROBE
K  3  Probe instruction       : PROBE
L  3  Probe instruction       : PROBE
M  3  Probe instruction       : PROBE
N  3  Probe instruction       : PROBE
B  4  Probe instruction       : PROBE
C  4  Probe instruction       : PROBE
D  4  MOVPR.MIC
E  4  SVP/LDPCTX, MF/MTPR     : SVPCTX - Save Process Context
F  4  SVP/LDPCTX, MF/MTPR     : SVPCTX - Save Process Context
G  4  SVP/LDPCTX, MF/MTPR     : SVPCTX - Save Process Context
H  4  SVP/LDPCTX, MF/MTPR     : SVPCTX - Save Process Context
I  4  SVP/LDPCTX, MF/MTPR     : LDPCTX - Load Process Context
J  4  SVP/LDPCTX, MF/MTPR     : LDPCTX - Load Process Context
K  4  SVP/LDPCTX, MF/MTPR     : LDPCTX - Load Process Context
L  4  SVP/LDPCTX, MF/MTPR     : LDPCTX - Load Process Context
M  4  SVP/LDPCTX, MF/MTPR     : LDPCTX - Load Process Context
N  4  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
B  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
C  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
D  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
E  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
F  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
G  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
H  5  SVP/LDPCTX, MF/MTPR     : MFPR - Move From Processor Register
I  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
J  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
K  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
L  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
M  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
N  5  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
B  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
C  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
D  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
E  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
F  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
G  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
H  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
I  6  SVP/LDPCTX, MF/MTPR     : MTPR - Move To Processor Register
J  6  11MODE.MIC
K  6  Compatibility mode      : Description
L  6  Compatibility mode      : DP0 -  BASIC
M  6  Compatibility mode      : DP0 -  BASIC
N  6  Compatibility mode      : DP0 -  BASIC
B  7  Compatibility mode      : DP0 -  BASIC
C  7  Compatibility mode      : DP0 -  [SMO]*DMO
D  7  Compatibility mode      : DP0 -  [SMO]*DMO
E  7  Compatibility mode      : DP0 -  [SMO]*DMO
F  7  Compatibility mode      : DP0 -  [SMO]*DMO
G  7  Compatibility mode      : DP0 -  [SMO]*DMO
H  7  Compatibility mode      : DP0 -  DMO*(ASH+ASHC+MUL+DIV)
I  7  Compatibility mode      : DP0 -  SRC MODE DECODING FOR BIN INS
J  7  Compatibility mode      : DP0 -  SRC MODE DECODING FOR BIN INS
K  7  Compatibility mode      : DP0 -  SRC MODE DECODING FOR BIN INS
L  7  Compatibility mode      : DP0(DP1 FOR BIN INSTRUCTIONS) -  DES
M  7  Compatibility mode      : DP0(DP1 FOR BIN INSTRUCTIONS) -  DES
N  7  Compatibility mode      : DP0(DP1 FOR BIN INSTRUCTIONS) -  DES
B  8  Compatibility mode      : DP0(DP1 FOR BIN INSTRUCTIONS) -  DES
C  8  Compatibility mode      : DP0(DP1 FOR BIN INSTRUCTIONS) -  DES
D  8  Compatibility mode      : DP1 -  MEMORY TO R/PC FOR BIN INSTRU
E  8  Compatibility mode      : DP1 (DP2 FOR BIN INSTRUCTIONS) -  ME
F  8  Compatibility mode      : DP1 (DP2 FOR BIN INSTRUCTIONS) -  ME
G  8  Compatibility mode      : DP1 (DP2 FOR BIN INSTRUCTIONS) -  ME
H  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
I  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
J  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
K  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
L  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
M  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
N  8  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
B  9  Compatibility mode      : DP1 -  (-DMO)*(ASH+ASHC+MUL+DIV)
C  9  IE.MIC
D  9  Interrupt & Exception   : Description
E  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
F  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
G  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
H  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
I  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
J  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
K  9  Interrupt & Exception   : INTIO, INTRPT, EXCPTN, EXCPT
L  9  Interrupt & Exception   : ARITHT, TRACE, RSVOP1
M  9  Interrupt & Exception   : SNV, KSNV
N  9  Interrupt & Exception   : Compatibility mode faults
B 10  Interrupt & Exception   : Compatibility mode faults
C 10  Interrupt & Exception   : SETFPD
D 10  Change modes            : CHMK, CHME, CHMS, CHMU
```

```
E 10  Change modes            : CHMK, CHME, CHMS, CHMU
F 10  Change modes            : CHMK, CHME, CHMS, CHMU
G 10  Change modes            : CHMK, CHME, CHMS, CHMU
H 10  MMFW.MIC
I 10  Memory management       : MEMORY MANAGEMENT FIRMWARE DESCRIPTI
J 10  Memory management       : MEMORY MANAGEMENT FIRMWARE DESCRIPTI
K 10  Memory management       : TBUFF MISS MICRO-TRAP ROUTINE
L 10  Memory management       : TBUFF MISS MICRO-TRAP ROUTINE
M 10  Memory management       : SET MBIT MICRO-TRAP ROUTINE
N 10  Memory management       : UNALIGNED DATA MICRO-TRAP ROUTINE
B 11  Memory management       : UNALIGNED DATA MICRO-TRAP ROUTINE
C 11  Memory management       : PAGE BNDRY MICRO-TRAP ROUTINE
D 11  Memory management       : PAGE BNDRY MICRO-TRAP ROUTINE
E 11  Memory management       : PAGE BNDRY MICRO-TRAP ROUTINE
F 11  Memory management       : PROTECTION MICRO-TRAP ROUTINE
G 11  Memory management       : IBUF PTE FETCH ROUTINE
H 11  Memory management       : IBUF PTE FETCH ROUTINE
I 11  Memory management       : IBUF PTE FETCH ROUTINE
J 11  Memory management       : PROBE VIRTUAL ADDRESS ROUTINE
K 11  Memory management       : PROBE VIRTUAL ADDRESS ROUTINE
L 11  Memory management       : PROBE VIRTUAL ADDRESS ROUTINE
M 11  Memory management       : PTE<M> TEST & SET ROUTINE
N 11  Memory management       : PTE<M> TEST & SET ROUTINE
B 12  Memory management       : PTE<M> TEST & SET ROUTINE
C 12  Memory management       : PTE<M> TEST & SET ROUTINE
D 12  Memory management       : FETCH PTE ROUTINE
E 12  Memory management       : FETCH PTE ROUTINE
F 12  Memory management       : FETCH PTE ROUTINE
G 12  Memory management       : FETCH PTE ROUTINE
H 12  Memory management       : FETCH PTE ROUTINE
I 12  Memory management       : FETCH PTE ROUTINE
J 12  Memory management       : FETCH PTE ROUTINE
K 12  Memory management       : FETCH PTE ROUTINE
L 12  Memory management       : FETCH PTE ROUTINE
M 12  Memory management       : FETCH PTE ROUTINE
N 12  Memory management       : MEMORY MANAGEMENT FAULT HANDLING ROU
B 13  Memory management       : MEMORY MANAGEMENT FAULT HANDLING ROU
C 13  Memory management       : MEMORY MANAGEMENT FAULT HANDLING ROU
D 13  Memory management       : MEMORY MANAGEMENT FAULT HANDLING ROU
E 13  Memory management       : BACK-UP GENERAL REGS AND PC
F 13  Memory management       : BACK-UP GENERAL REGS AND PC
G 13  Memory management       : BACK-UP GENERAL REGS AND PC
H 13  Memory management       : BACK-UP GENERAL REGS AND PC
I 13  ERCODE.MIC
J 13  Error handling          : IB.ERR - IBUF error routine
K 13  Error handling          : IB.ERR - IBUF error routine
L 13  Error handling          : IB.ERR - IBUF error routine
M 13  Error handling          : Machine check errors
N 13  Error handling          : Machine check errors
B 14  Error handling          : Machine check errors
C 14  Error handling          : Machine check errors
D 14  Error handling          : Error handling initialize
E 14  Error handling          : Error handling initialize
F 14  Error handling          : Error snapshot routine
G 14  Error handling          : Error snapshot routine
H 14  Error handling          : Error snapshot routine
I 14  Error handling          : Error snapshot routine
J 14  Error handling          : Push parameters
K 14  Error handling          : Push parameters
L 14  CNSL.MIC
M 14  Console interface       : DESCRIPTION
N 14  Console interface       : HALT AND CONTINUE
B 15  Console interface       : EXAMINE MEMORY
C 15  Console interface       : EXAMINE MEMORY
D 15  Console interface       : DEPOSIT MEMORY
E 15  Console interface       : DEPOSIT MEMORY
F 15  Console interface       : EXAMINE/DEPOSIT GPR'S
G 15  Console interface       : EXAMINE/DEPOSIT IPR'S
H 15  Console interface       : QUAD CLEAR AND UNJAM
I 15  Console interface       : ECO functionality test
J 15  Console interface       : ECO functionality test
K 15  CALCRC.MIC
L 15  Procedure call & CRC    : CALLG, CALLS
M 15  Procedure call & CRC    : CALLG, CALLS
N 15  Procedure call & CRC    : CALLG, CALLS
B 16  Procedure call & CRC    : CALLG, CALLS
C 16  Procedure call & CRC    : CALLG, CALLS
D 16  Procedure call & CRC    : CRC Description
E 16  Procedure call & CRC    : CRC Entry
F 16  Procedure call & CRC    : CRC Loop setup and loop
G 16  Procedure call & CRC    : CRC Loop setup and loop
H 16  Procedure call & CRC    : CRC Termination and FPD
I 16  Procedure call & CRC    : CRC Termination and FPD
J 16  FPA.MIC
K 16  FPA Interface           : Description
L 16  FPA Interface           : Description
```