UNIVERSAL ASSEMBLER VERSION 1.2 JANUARY 4, 1978 (IN-HOUSE)

CONFIDENTIAL PROPRIETARY INFORMATION

THIS ITEM IS THE PROPERTY OF DATAPOINT CORPORATION, SAN ANTONIO, TEXAS, AND CONTAINS CONFIDENTIAL AND TRADE SECRET INFORMATION. THIS ITEM MAY NOT BE TRANSFERRED FROM THE CUSTODY OR CONTROL OF DATAPOINT EXCEPT AS AUTHORIZED BY DATAPOINT AND THEN ONLY BY WAY OF LOAN FOR LIMITED PURPOSES. IT MUST NOT BE REPRODUCED IN WHOLE OR IN PART AND MUST BE RETURNED TO DATAPOINT UPON REQUEST AND IN ALL EVENTS UPON COMPLETION OF THE PURPOSE OF THE LOAN.

NEITHER THIS ITEM NOR THE INFORMATION IT CONTAINS MAY BE USED OR DISCLOSED TO PERSONS NOT HAVING A NEED FOR SUCH USE OR DISCLOSURE CONSISTENT WITH THE PURPOSE OF THE LOAN, WITHOUT THE PRIOR WRITTEN CONSENT OF DATAPOINT.

```
COMMAND LINE WAS: SNAP3 PROC,,,PROC;GRPLX

INCLUSION A: PROCPARM/TXT
INCLUSION B: PMACMIC/TXT
INCLUSION C: GMACROZ/TXT
INCLUSION D: PORTASGN/TXT
INCLUSION E: PROCEQUS/TXT
INCLUSION F: MDEF1800/TXT
INCLUSION G: BDEF1800/TXT
INCLUSION H: PORTEQUS/TXT
INCLUSION I: DDEF1800/TXT
INCLUSION J: HDEF1800/TXT

PROGRAM NAME: PROC

PROGRAM ADDRESS BLOCKS:      010000    /ABSOLUTE/  SIZE=000000  (ABS)
                             167400    /SYSIVR/    SIZE=000400  (ABS)
                             170000    /SYSROM/    SIZE=000047  (ABS)
                             000000    /PROCL/     SIZE=002000  (REL)
                             000000    /PROCP/     SIZE=004000  (REL)

PRIMARY TRANSFER ADDRESS:    000000    /PROCL/

EXTERNAL DEFINITIONS:

POR      000000 /PROCL/   INPUT    000061 /PROCL/   IVIOLS   000317 /PROCL/   PIN      000074 /PROCL/
EXADR    000104 /PROCL/   EXSTAT   000123 /PROCL/   EXDATA   000134 /PROCL/   OUTPUT   000145 /PROCL/
MIN      000200 /PROCL/   MOUT     000230 /PROCL/   BP       000302 /PROCL/   SC       000304 /PROCL/
SCROML   000310 /PROCL/   SCROM    000311 /PROCL/   SCDON    000335 /PROCL/   UDPOP    000314 /PROCL/
UDOP     000321 /PROCL/   SCLST    000325 /PROCL/   SCRAM    000333 /PROCL/   BETA     000400 /PROCL/
ALPHA    000417 /PROCL/   SYSRET1  000512 /PROCL/   REGS     000557 /PROCL/   STL      000652 /PROCL/
RIND     001215 /PROCL/   SYSTAT   000757 /PROCL/   BT       001000 /PROCL/   RIN256   001110 /PROCL/
RINST    001116 /PROCL/   BCP      001126 /PROCL/   RIN16    001207 /PROCL/   BFSB     001222 /PROCL/
BFAC     001226 /PROCL/   BFS      001300 /PROCL/   SLC      001346 /PROCL/   SRC      001355 /PROCL/
SRE      001366 /PROCL/   CCS      001402 /PROCL/   PLR      001407 /PROCL/   PSR      001440 /PROCL/
INCPA    001463 /PROCL/   INCP     001470 /PROCL/   DECPA    001512 /PROCL/   DECP     001517 /PROCL/
DECX     001541 /PROCL/   INCX     001543 /PROCL/   DS       001647 /PROCL/   DLHL     001651 /PROCL/
DL       001655 /PROCL/   BRL      001731 /PROCL/   EI       001745 /PROCL/   DI       001756 /PROCL/

EXTERNAL REFERENCES (UNDEFINED SYMBOLS):

RETS     FETCHW    SRVRPT    PSHST0    PCMOD     MEMPF$    SRVDO     POPST0    SYSRET0   FETCHI    FETCH     EIROJ

UNUSED LABELS:

RINSIO   SCLSTI    SCDECP    MEMPFSC
```

```
1,
2,          . 2.9   HJS   78 JUL  20     FINAL RELEASE OF VERSION 9
3,          . 2.9.K HJS   78 APR  23     SPLIT PROC, MAKE RELOCATABLE, CHANGE APF, ADD AML
4,          . 2.9.J HJS   78 MAR  20     RESTRUCTURE INTERRUPT SEQUENCE & MINOR MODS
5,          . 2.9.I HJS   78 FEB  27     CORRECT 9.H FOR FAULT CLEANUP
6,          . 2.9.H HJS   78 FEB  16     EVERYBODY MEMPF'S, KEYBOARD SCAN, & SIR CHANGE
7,          . 2.9.G HJS   78 FEB   3     CORRECT TIMING, COMMENTS, & ADD POR TIMEOUT
8,          . 2.9.F HJS   78 JAN  11     FIXING MIN/MOUT TIMINGS
9,          . 2.9.E HJS   78 JAN   4     TESTING REPEATED KEYIN CONTROLS
10,         . 2.9.D HJS   77 DEC  21     CORRECT STL INSTRUCTION
11,         . 2.9.C HJS   77 DEC  13     BACK OFF FROM KBD RPT & RE-DO STL FOR TIMING
12,         . 2.9.B HJS   77 NOV  20     INCLUDE TIMINGS AS CALCULATED & FIX MINOR BUGS
13,         . 2.9.A HJS   77 NOV  14     CHANGE KEYBOARD CODE TO AID REPEATED KEY CONTROL
14,         *
15,         . 2.8.B HJS   77 SEP  22     MTI CHANGE SO LENGTH IS 2 BYTE NUMBER
16,         . 2.8.A HJS   77 SEP  19     MTI CHANGE TO ALLOW MFRPT ON ANY INTERRUPT
17,         *
18,         . 2.7.  HJS   77 SEP   7     MINOR BUG-FIX AND OPTIMIZATION FOR RELEASE
19,         *
20,         . 2.H.B HJS   77 AUG  31     MTI SPECIAL VERSION
21,         *
22,         . 2.5.C HJS   77 AUG  16     UPDATE COMMENTS ON THE CODE
23,         . 2.5.B HJS   77 JULY 13     CORRECTED NAMES FOR COM REGISTERS
24,         . 2.5.A HJS   77 JULY 12     UP TO NEXT NEW VERSION NUMBER
25,         *
26,         . 2.4.B HJS   77 JULY 12     FIXED ILLEGAL MAR CHANGE IN REGL RETURN TO FETCH
27,         .                            FIXED FILE TO CONFORM TO VRP FORMAT (A LITTLE)
28,         . 2.4.A HJS   77 JULY 7      INITIAL PRE-RELEASE OF THE MICRO-CODE
29.         *
```

```
30;
31;                     +
32;                     . USEFUL INFORMATION TO BE PASSED ON:
33;                     *
34;                     . ************************************************************************
35;                     . IF SECTOR CONTAINING SEKBS1 IS WRITE PROTECTED THEN CAN NOT DO A RESTART!!!!!
36;                     . ************************************************************************
37;                     . DL$DO CODE, ACC$DO & SYSRET FORCE! PRIVED MODE TO BE SURE CAN WRITE/ACCESS
38;                     . MEMORY THAT PROTECTED, FXIO, COMM MUST DO THIS ALSO!!
39;                     . ************************************************************************
40;                     . ALL SMR'S AND MDW'S MUST HAVE FOLLOWING   MWAIT ,MEMPFX    EVERYWHERE
41;                     . IN THE CODE EVEN IF  MWAIT ,$+1 IS USED (IGNORING MEMORY FAULT)
42;                     . ************************************************************************
43;                     . FOR TESTING MEMPF, CHANGE MWAIT MACRO TO BE JT,MP  $+1  <<<
44;                     . CODE THERE, TESTABLE AS USED BUT NO EFFECT!!
45;                     . ************************************************************************
46;                     . EACH SERVICE ROUTINE THAT USES THE MAR MUST HAVE A MWAIT ,$+1 BEFORE THE
47;                     . MAR CHANGED, IF NOT USED, RETURN THROUGH SRVNXT!
48;                     . ************************************************************************
49;                     . ALL INPUTS FROM A PORT INSTRUCTION SHOULD BE LOGICAL (OUTPUT CAN ONLY BE
50;                     . LOGICAL OR 'IT').
51;                     . ARITHMETICS MAY BE DONE ON INPUTS WITH CAUTION, ONLY IF THE DATA HAS NO SETUP
52;                     . TIME,  . . MDR IS OK, BUT IMPI OR URI ARE NO-NO'S.
53;                     . ************************************************************************
54;                     . DELAY AROUND SECTOR TABLE WRITES IS NOT NECESSARY.
55;                     . ************************************************************************
56;                     . DELAY DURING FONT LOAD IS ONLY AFTER LDCH.  A 400  SEC. DELAY IS NECESSARY
57;                     . FROM AFTER LDCH TO ANY OF  RDLM, SDLM, SKCH, MAROL, OR  LDMAP.
58;                     . ALSO, RDLM AND SDLM SHOULD NOT BE TOO CLOSE TOGETHER.
59;                     . ************************************************************************
60;                     .
61;                             INC     PROCPARM
62;                     .
63; 000000              PROCL   ORG     PROC                    LOGICAL SPACE IN ROM'S
64; 000000              PROCP   ORG     0                       PHYSICAL SPACE DONE AT LINK TIME
65; 000000              PROCL   USE     PROCL                   ENABLE BOTH ADDRESS SPACES
66; 000000                      USE     PROCP                   BUILD CODE IN PHYSICAL SPACE
67; 000000L             PROCP   LOC     PROCL,2                 LOC'ING TO LOGICAL ADDRESSES
```

```
                                        . POWER ON RESTART, INITIALIZE THE WORLD
                                        *
000000L                                 POR:
                                        .
                                        . INITIALIZE ALL CONTROL TABLES THAT CAN NOT BE INIT'D BY POR MACRO CODE
                                        .
000000L 01011001 11111111                       BPGX    $
                                        .       STB     RDLM                    DO NOT CLEAR UP THE DISPLAY
000001L 11000100 11111110                       MWAIT   ,IGNORE                 TIMEOUT SAFETY (IGNORE MEMORY FAULT)
000002L 00110111 01000110                       STB     CMPF                    (INSTEAD OF FAULT JUMP)
                                        .                                       ** T-REG IS ZERO FOR THE FOLLOWING **
000003L 01010001 00000000                       LDPI    PSW0,0                  SET EMULATION PSW
000004L 00110111 10001100
000005L 00110111 00000100                       LDPT    MODW                    PORT CONTROLS APPROPRIATLY
000006L 00110111 00000011                       LDPT    BASW                    BASE REGISTER IS INITIALLY ZERO
000007L 00000111 11110101                       LDRT    MBSTAT                  MICRO-BUS STATE INITIALLY OFF!
000010L 00000111 11110100                       LDRT    MBITS                   MICRO-BUS INITIALLY CLEARED FOR ACTION
                                                IFC     APF
                                                XIF
                                                IFS     APF
000011L 00000111 11111111                       LDRT    ACCTL                   NO AUDIO CHANNEL INITIALLY
                                                XIF
000012L 00110111 00100101                       LDPT    SDLCMD                  COMM OUTPUT IS OFF
000013L 00110111 00100100                       LDPT    ACUOT                   AUTO CALL UNIT IS ALSO OFF
000014L 01101111 11111001                       LDRT    COMMODE                 FLUSH THE COMMODE! ("MURF")
                                        .       LDPT    LIMP                    IMPLICIT REG IS ZERO AT THE START
000015L 00000111 11110001                       LDRT    KBSCNT                  SAVE THE SCAN NUMBER AND
000016L 00110111 01000100                       LDPT    KBSC                    START THE FIRST KEY SCAN
                                        .
000017L 01010001 11111111               .       LDRI    MADR,-1                 NO DEVICE ACCESSED YET ON MICRO-BUS
000020L 00000111 11110011
000021L 01010001 10011100                       LDRI    PDLNP,SEDLBOT,CC        POINT TO THE BOTTOM LINE
000022L 00000111 10110000
000023L 01010001 00100100                       LDPI    PCOL,SRTMOUT            ASSUME THAT A TIMEOUT WILL HAPPEN
000024L 00110111 10001001
000025L 00110001 00110111                       TSTIP   ,STLIMOUT,STEK          CHECK IF IT DID
000026L 01000101 00000010
000027L 11000010 11010010                       BRA     PORDUT,FZ               YES, DON'T INIT STL OR MEMORY
                                        .
                                        . MUST INIT SECTOR TABLE & ZERO SYSESR OR IMMEDIATE RESTART OR MEMPF
                                        .       LDPI    MODW,SWBASD             DISABLE BASING FOR SECTOR TABLE LOAD
000030L 01010001 00001100               .       LDTI    STAE+STWE               ** INIT WHOLE SECTOR TABLE **
000031L 00110111 11100000               PORLST  LDPT    MAROH,STW
000032L 00110111 00000101
000033L 01010010 00010000                       DOTI    ,AC,16                  LOOP TILL ALL ARE ACCESS & WRITE ENABLED
000034L 11000000 11100110                       BRA     PORLST,FC
                                        .
000035L 01010001 11110000                       LDPI    STW,SYSROM>8,AND,0360   SECTOR 017 IS SYSTEM ROM, PROTECT IT
000036L 00110111 00000101
                                        .       DLDPI   SP0,SESTACK             STACK TO EMULATION RAM (CAN ELIMINATE)
000037L 01010001 11101111                       LDPI    MAROH,SYSESR>8          POINT TO THE EMULATION SUPPORT PAGE
000040L 00110111 11100000
000041L 01010001 00000000                       LDPI    MAROL,SYSESR            START OF IT (T-REG IS ZERO)
```

```
                        . POWER ON RESTART, INITIALIZE THE WORLD
        000042L 00110111 11000000
114.    000043L 01101111 11110010          LDRT   TEMPL                  INITIALIZE THE COUNTER
115.    000044L 00110111 00001101          STB    DMAR                   CORRECT MAR SO ALL OF PAGE INITED
116.                                       .
117.    000045L 00110111 00001100  PORZRO   STB    IMAR,MDW               ZERO THE NEXT BYTE (T=REG IS ZERO)
        000046L 00110111 00100001
118.    000047L 01110001 11110010          INCR   TEMPL,TEMPL            COUNT ALL OF THE BYTES IN THE PAGE
        000050L 01101110 01110010
119.    000051L 01010001 00000000          LDTI   0                      DATA TO BE WRITTEN IS ZERO
120.    000052L 11000100 11010101          MWAIT  ,IGNORE                !! IGNORE ANY AND ALL FAULTS !!
121.    000053L 11000000 11011010          BRA    PORZRO,FC              CONTINUE TILL ALL PAGE DONE
122.                                       *
123.    000054L 00110111 10001001  FETPC   LDPT   PCOL                   LOAD THE PC LSB ADDRESS
124.                                       .
125.    000055L 01010001 11110000  POROUT  LDPI   PCOH,SYSROM>8
        000056L 00110111 10001000
126.                                       .
127.                                       . NOTE:  NORMALLY HERE, PCL IS ZERO AND PCH = SYSROM>8
128.                                       .        THIS GENERATES A POINTER TO SRPOWER-UP VECTOR ENTRY
129.                                       .
130.  >000057L 01011001 11111111          BRAX   RETS                   OK, MICRO-POR DONE, DO MACRO POR
      >000060L 11001111 11111111
131.                                       .                                (OR TIMEOUT) (OR SC ERROR)
```

```
                                . INPUT OUTPUT OPERATIONS
                                *
                                INPUT:
                                . 5,80    (    101)  IN                 INPUT FROM 5500 BUS
                                . 7,80    (IMP 101)  INR                INPUT TO OTHER THAN A-REG
                                . ******
                                . NOTE:  INSTRUCTION FETCH TIME (APPROX. 2 MICRO-SEC.) INCLUDED IN COUNT FOR
                                .        TOTAL OF 5 MICRO-SEC. I/O TIMING OF THE INSTRUCTION.
                                . ******
                                .
                                .        IF USER THEN IVIOL
                                .          ELSE
                                .   INPUTX:
                                .          IMP <- INBUS
                                .          SINS                          (ACKNOWLEDGE THE DEVICE)
                                .          DELAY                         (WAIT FOR BUSS TIMING TO COMPLETE)
                                .          FETCHW
                                .
000061L 00110001 11011100       .          TSTIP  ,SWUSER,PSWI
000062L 01000101 00000100
000063L 11000010 00110000       .          BRA    IVIOL$,FZ              CONTINUE ONLY IF IN PRIVED MODE
000064L 00110001 00010100   INPUTX  LDPP   SINS,INBUS,IMPO               GET INPUT DATA AND FAST ACKNOWLEDGE
000065L 00110111 00101011
000066L 00110111 10001111
000067L 00110001 00110001   INPW1    TSTIP  ,STIODR,STATUS
000070L 01000101 00000010
000071L 11000011 11001000       .          BRA    INPW1,TZ               WAIT FOR ACK. TO MAKE ITSELF KNOWN
>000072L 01011001 11111111  FTCHIO   BRAX   FETCHW                       'W' BECAUSE I/O DELAY TIMING NEEDS IT
>000073L 11001111 11111111
                                .                                         AS WELL AS EX ADR AND MIN MEMORY WRITES
                                *
000074L                         PIN:
                                . 6,25    (    103)  PIN                 PARITY CHECKING INPUT
                                . 8,25    (IMP 103)  PINR                INTO OTHER THAN A-REG
                                . ******
                                . NOTE:  INSTRUCTION FETCH TIME (APPROX. 2 MICRO-SEC. INCLUDED IN COUNT FOR
                                .        TOTAL OF 5 MICRO-SEC. I/O TIMING OF THE INSTRUCTION.
                                . ******
                                .
                                .        IF USER THEN IVIOL
                                .          ELSE
                                .          IF NOT STPFIN IN STATUS THEN INPUTX
                                .            ELSE T <= SVINP
                                .                 SCLST
                                .
                                .                                         *** INITIAL DELAY 2 MICRO-SEC. IN FETCH
000074L 00110001 11011100       .          TSTIP  ,SWUSER,PSWI
000075L 01000101 00000100
000076L 11000010 00110000       .          BRA    IVIOL$,FZ              DON'T CONTINUE UNLESS PRIVED
000077L 00110001 00110001       .          TSTIP  ,STPFIN,STATUS
000100L 01000101 00000100
000101L 11000011 11001011       .          BRA    INPUTX,TZ              NO PARITY FAULT, CONTINUE REGULAR INPUT
000102L 01010001 00000110   MINPAR   LDTI   SVINP                        USE PARITY INPUT ERROR VECTOR
```

```
                . INPUT OUTPUT OPERATIONS
179*  000103L 11001111 00101101          BRA    SCLSTW              AND CALL SUPERVISOR ERROR ROUTINE
180*                                     .                          PC CORRECT OR BACK UP IF IMP NON-ZERO
```

```
181,
182,  000104L                          +
                                       EXADR:
183,
184,                                   .
                                       . 11.50  (    121)  EX ADR              SELECT DEVICE (AND PUT IN STATUS MODE)
185,                                   . 13.50  (IMP 121)  EX ADR
186,
187,                                   .
188,                                   .       OTBUS <- IMP
189,                                   .       IF SWUSER THEN IVIOL$
190,                                   .       PSW <- PSW .AND. (-1-SWSTDT)  MARK IT IN STATUS MODE
191,                                   .       (SEXADR) <- IMP               SAVE THE EXADR ADDRESS
192,                                   .       OUTW0
                                       .
193.  000104L 00110001 11011111                LDPP   OTBUS,IMPI,SIOO        OUTPUT DATA AND START FIRST DELAY
      000105L 00110111 00100000
      000106L 00110111 00101100
194.  000107L 00110001 11011100                TSTIP  ,SWUSER,PSWI           SHOULD I HAVE DONE THAT?
      000110L 01000101 00000100
195.  000111L 11000010 00110000                BRA    IVIOL$,FZ
196.  000112L 01010101 11101111                DOPI   PSWO,ND,-1-SWSTDT      MARK IN STATUS MODE
      000113L 00110111 10001100
197.  000114L 01010001 10101010                DLDPI  MARD,SEXADR
      000115L 00110111 11000000
      000116L 01010001 11101111
      000117L 00110111 11100000
198.  000120L 00110001 11011111                LDPP   MDW,IMPI               OUTPUT ADDRESS TO ITS SAVE AREA
      000121L 00110111 00100001
199.  000122L 11001111 10010100                BRA    OUTW0
200.                                   *
201.  000123L                          EXSTAT:
202.                                   . 11.60  (    123)  EX STATUS           PUT IN STATUS MODE
203.                                   . 13.60  (IMP 123)  EX STATUS
204.
205.                                   .
206.                                   .       OTBUS <- IMP
207.                                   .       IF SWUSER THEN IVIOL$
208.                                   .       PSW <- PSW .AND. (-1-SWSTDT)  MARK IT IN STATUS MODE
209.                                   .       OUTW0
                                       .
210.  000123L 00110001 11011111                LDPP   OTBUS,IMPI,SIOD        OUTPUT DATA AND START FIRST DELAY
      000124L 00110111 00100000
      000125L 00110111 00101100
211.  000126L 00110001 11011100                TSTIP  ,SWUSER,PSWI           SHOULD I HAVE DONE THAT?
      000127L 01000101 00000100
212.  000130L 11000010 00110000                BRA    IVIOL$,FZ
213.  000131L 01010101 11101111                DOPI   PSWO,ND,-1-SWSTDT      MARK IN STATUS MODE
      000132L 00110111 10001100
214.  000133L 11001111 10010100                BRA    OUTW0
```

```
                                . INPUT OUTPUT OPERATIONS

215,                            *
216,  000134L                   EXDATA:
217,                            . 11.60  (    125)  EX DATA            PUT IN DATA MODE
218,                            . 13.60  (IMP 125)  EX DATA
219,                            .
220,                            .       OTBUS <- IMP
221,                            .       IF SWUSER THEN IVIOL$
222,                            .       PSW <- PSW .OR. SWSTDT        (MARK IN DATA MODE)
223,                            .       OUTW0
224,                            .
225.  000134L 00110001 11011111         LDPP   OTBUS,IMPI,SIOD        OUTPUT DATA AND START FIRST DELAY
      000135L 00110111 00100000
      000136L 00110111 00101100
226.  000137L 00110001 11011100         TSTIP  ,SWUSER,PSWI           SHOULD I HAVE DONE THAT?
      000140L 01000101 00000100
227.  000141L 11000010 00110000         BRA    IVIOL$,FZ
228.  000142L 01010011 00010000         DOPI   PSWO,OR,SWSTDT         MARK IN DATA MODE
      000143L 00110111 10001100
229.  000144L 11001111 10010100         BRA    OUTW0
```

```
                        . INPUT OUTPUT OPERATIONS
                        +
000145L                 OUTPUT:
                        . 11,60  (1  :  =2,3:   ODD)  EX COM    OUTPUT TO 5500 BUS
                        . 13,60  (IMP 1  )  EX  COM             WITH OTHER THAN THE A-REG
                        .
                        .        (    127)  EX WRITE            WRITE DATA TO THE DEVICE
                        .        (IMP 127)  EX  WRITE           USING OTHER THEN A-REG
                        .
                        .        (    131)  EX COM1             DO CONTROL STROBE 1
                        .        (IMP 131)  EX  COM1            USING OTHER THEN A-REG
                        .
                        .        (    133)  EX COM2             DO CONTROL STROBE 2
                        .        (IMP 133)  EX  COM2            USING OTHER THEN A-REG
                        .
                        .        (    135)  EX COM3             DO CONTROL STROBE 3
                        .        (IMP 135)  EX  COM3            USING OTHER THEN A-REG
                        .
                        .        (    137)  EX COM4             DO CONTROL STROBE 4
                        .        (IMP 137)  EX  COM4            USING OTHER THEN A-REG
                        .
                        .           OTBUS <- IMP
                        .           IF USER THEN IVIOL
                        .       OUTW0:
                        .           DELAY                       DELAY 0 (ON DATA GOING DOWN BUS)
                        .           SOTS                        DO COMMAND OUT STROBE
                        .           DELAY                       DELAY 1 (ON COMMAND DOWN BUS)
                        .       OUTW2:
                        .           DELAY                       DELAY 2 (ON PARITY COMING BACK)
                        .           IF NOT STPFOU IN STATUS THEN FETCHW
                        .             ELSE T <- SVOUTP
                        .                  SCLST
```

```
                                . INPUT OUTPUT OPERATIONS
                                *
000145L 00110001 11011111       LDPP   OTBUS,IMPI,SIOD        OUTPUT DATA AND START ZEROTH DELAY
000146L 00110111 00100000
000147L 00110111 00101100
000150L 00110001 11011100       TSTIP  ,SWUSER,PSWI
000151L 01000101 00000100
000152L 11000010 00110000       BRA    IVIOL$,FZ              ONLY CONTINUE IF PRIVED
000153L 00110001 00110001 OUTW0 TSTIP  ,STIODR,STATUS
000154L 01000101 00000010
000155L 11000011 10010100       BRA    OUTW0,TZ               DELAY 0 WAITING FOR DATA TO REACH DEVICE
000156L 00110111 00101111       STB    SOTS                   GIVE COMMAND
000157L 00110001 00110001 OUTW1 TSTIP  ,STIODR,STATUS
000160L 01000101 00000010
000161L 11000011 10010000       BRA    OUTW1,TZ               DELAY 1 WAITING FOR COMMAND TO GET THERE
000162L 00110111 00101100       STB    SIOD                   EXTEND THE COMMAND DELAY FOR FINAL STEP
000163L 00110001 00110001 OUTW2 TSTIP  ,STIODR,STATUS
000164L 01000101 00000010
000165L 11000011 10001100       BRA    OUTW2,TZ               DELAY 2 WAITING FOR PARITY TO RETURN
000166L 01000101 00001000       TSTIT  ,STPFOU                WAS THERE AN OUTPUT PARITY FAULT?
000167L 11000011 11000101       BRA    FTCHIO,TZ              NO!
000170L 01010001 00001100 MOTPAR LDTI  SVOUTP                 YES, TELL SUPERVISOR THAT THERE WAS
000171L 11001111 00101101       BRA    SCLSTW
                                *
000172L                         MINOWT
                                . COMMON MIN/MOUT FINAL TERMINATION CODE
                                .
                                .       SIOD
                                .       MODW <- PSW <- PSW .AND. (-1-SWRPT)
                                .       DELAY
                                .       SIOD
                                .       OUTW2
                                .
000172L 00110111 00101100       STB    SIOD                   DOUBLE DELAY AT THE END
000173L 00110001 11011100       DOPIP  PSWO,ND,-1-SWRPT,PSWI  TURN OFF THE REPEATED FLAG
000174L 01010101 11011111
000175L 00110111 10001100
000176L 00110111 00000100       LDPT   MODW                   RESET IT AND EI/DI BITS
000177L 11001111 10010000       BRA    OUTW1                  GO TO FINAL DELAYS AND PARITY CHECK
```

```
                        . INPUT OUTPUT OPERATIONS
                        .
                        +
                        MIN:
                        . NOTE: IF  EI / MIN, THE EI DOESN'T TAKE EFFECT UNTIL THE INSTRUCTION
                        .       AFTER THE MIN, ONE LATER THAN ON 5500
                        .       BUT, IF MIN INTERRUPTED AT MACRO LEVEL FOR ANY REASON,
                        .       WHEN IT IS RETURNED IT WILL BE AS IF THE EI HAD ALREADY TAKEN EFFECT!
                        .
                        .        (111 061)  MIN                 5500 COMPATIBLE (UNNECESSARY MIN)
                        . 9.45 + C * 8.40
                        .        (    061)  MIN                 MULTIPLE INPUT UNSPECIFIED
                        .        (IMP 061)  MIN                 UNSPECIFIED UNNECESSARY MIN
                        .
                        .        IF USER THEN IVIOL
                        .          ELSE MODW <= PSW .AND. -1 - SWINTE
                        .               DELAY
                        .               IF STPFIN IN STATUS THEN MINPAR
                        .                 ELSE (HL) <= INBUS
                        .                      SINS
                        .                      HL <= HL + 1
                        .                      TEMP1 <= CODMIN
                        .        MINOUT:
                        .                      DELAY
                        .                      T <= URC <= URC - 1
                        .                      IF (T .AND. 017) = ZERO THEN MINOWT
                        .                      ELSE IREG <= TEMP1
                        .                           IF SRVREQ THEN SRVRPT
                        .                             ELSE IDCOD (IREG)
                        .
000200L 00110001 11011100               TSTIP   ,SWUSER,PSWI
000201L 01000101 00000100
000202L 11000010 00110000               BRA     IVIOL$,FZ               ONLY CONTINUE IF PRIVED
000203L 01010101 11111110               DOPI    MODW,ND,-1-SWINTE       DISABLE ONE MS. INTERRUPTS
000204L 00110111 00000100
000205L 00110001 11000110               DLDX    HL2MR                   PRELOAD MAR WITH WHERE TO SAVE DATA
000206L 00110001 11100101
000207L 00110111 00101100               STB     SIOD                    START ZEROTH DELAY
000210L 00110001 00110001   MINW0       TSTIP   ,STIODR,STATUS          WAIT ON INITIAL DELAY
000211L 01000101 00000010
000212L 11000011 01110111               BRA     MINW0,TZ
000213L 01000101 00000100               TSTIT   ,STPFIN
000214L 11000010 10111101               BRA     MINPAR,FZ               DO NOT CONTINUE IF PARITY FAULT
000215L 00110001 00010100               LDPP    SINS,INBUS,MDW          GET DATA FAST ACKNOWLEDGE AND SAVE IT
000216L 00110111 00101011
000217L 00110111 00100001
000220L 00110001 10010000               DADDP   URO+UR,MARI             UPDATE HL, SADLY CAN'T BE TOO FAST
000221L 00010110 01110010
000222L 00110111 10000110
000223L 00110001 10110000
000224L 00110110 10000101
000225L 01010001 00110001               LDRI    TEMP1,CODMIN,CC         NEEDED ONLY FOR COMMON SPEEDUP CODE
000226L 01101111 10110001
000227L 11001111 01001101               BRA     MINOUT                  GO TO COMMON MIN/MOUT CODE
```

```
                                . INPUT OUTPUT OPERATIONS
                                .
                                +
                                MOUT:
                                . NOTE: IF  EI / MOUT, THE EI DOESN'T TAKE EFFECT UNTIL THE INSTRUCTION
                                .       AFTER THE MOUT, ONE LATER THAN ON 5500
                                .       BUT, IF MOUT INTERRUPTED AT MACRO LEVEL FOR ANY REASON,
                                .       WHEN IT RETURNS, IT WILL BE AS IF THE EI HAD ALREADY TAKEN EFFECT!
                                .
                                .        (111 071)  MOUT                FOR 5500 COMPATIBILITY (NOT NECESSARY)
                                .        (    071)  MOUT                MULTIPLE OUTPUT UNSPECIFIED
                                . 9.45 + C * 8.75
                                .        (IMP 071)  MOUT                UNSPECIFIED UNNECESSARY IMPLICIT CODE
                                .
                                .        IF USER THEN IVIOL$
                                .          ELSE MODW <= PSW .AND. (-1-SWINTE)
                                .               OTBUS <= (HL)
                                .               IREG <= CODWRITE        SO THAT SOTS WILL BE CORRECT
                                .               DELAY
                                .               IF STPFOU IN STATUS THEN MOTPAR
                                .                 ELSE SOTS
                                .                      HL <= HL + 1
                                .                      DELAY
                                .                      TEMP1 <= CODMOUT  SO THAT REPEATS WILL BE CORRECT
                                .                      MINOUT
                                .
356.  000230L 00110001 11000110          DLDX   HL2MR,,SMR              GET DATA AS SOON AS POSSIBLE
      000231L 00110001 11100101
      000232L 00110111 01000111
357.  000233L 00110001 11011100          TSTIP  ,SWUSER,PSWI            ONLY CONTINUE IF PRIVED
      000234L 01000101 00000100
358.  000235L 11000010 00110000          BRA    IVIOL$,FZ
359.  000236L 01010101 11111110          DOPI   MODW,ND,-1-SWINTE       DISABLE ONE M.SEC. INTERRUPT
      000237L 00110111 00000100
360.                                     .                              ?? CHANGES BASE ENABLE/DISABLE BIT ??
361.  000240L 11000100 01011111          MWAIT  ,MEMPF0
      000241L 11000111 00001100
362.  000242L 00110001 00110110          LDPP   OTBUS,MDR,SIOD          OUTPUT DATA AND START DELAY
      000243L 00110111 00100000
      000244L 00110111 00101100
363.  000245L 01010001 01010111          LDPI   LIREG,CODWRT            FIX IT UP FOR THE SOTS
      000246L 00110111 00000000
364.  000247L 00110001 00110001  MOTW0   TSTIP  ,STIODR,STATUS
      000250L 01000101 00000010
365.  000251L 11000011 01011000          BRA    MOTW0,TZ                DELAY WHILE DATA GETS TO THE DEVICE
366.  000252L 01000101 00001000          TSTIT  ,STPFOU
367.  000253L 11000010 10000111          BRA    MOTPAR,FZ               DO THE PARITY CHECK (FASTER THAN EX COM)
368.  000254L 00110111 00101111          STB    SOTS                    OK, SO DO THE COMMAND NOW!
369.  000255L 00110111 00001100          STB    IMAR
370.  000256L 00110001 10000110          DLDX   MR2HL                   UPDATE HL (THIS IS FASTEST WAY)
      000257L 00110001 10100101
371.  000260L 01010001 00111001          LDRI   TEMP1,CODMOUT,CC        FOR MINOUT CODE
      000261L 01101111 10110001
372.  000262L                    MINOUT
```

```
                                . INPUT OUTPUT OPERATIONS

373.  000262L 00110001 00110001 MINW1   TSTIP  ,STIODR,STATUS          WAIT ON ACK, TO GET TO THE DEVICE
      000263L 01000101 00000010
374.  000264L 11000011 01001101         BRA    MINW1,TZ
375.                            . * * *  7,00 - MOUT
376.                            .        6,65 - MIN
377.  000265L 00110001 11010010         DOPIP  URF0+URC,SB,1,URI+URC  DECREMENT THE COUNT IN C
      000266L 01010100 00000001
      000267L 00110111 01100010
378.  000270L 01000101 00001111         TSTIT  ,017                   BASE 16 END?
379.  000271L 11000011 10000101         BRA    MINOWT,TZ              COMPLETE AS A SIMPLE EX COM
380.  000272L 01110001 11110001         LDPR   LIREG,TEMP1            RESTORE TO MIN OR MOUT
      000273L 00110111 00000000
381.                            *
382.  000274L                   RINSIO
383.                            .       THIS CODE IS SIMILAR TO RINST CODE ON THE REPEATED PAGE
384.                            .
385.  000274L 11000100 01000011         MWAIT  ,MEMPF0                WAIT ON MIN MDW TO FINISH
      000275L 11000111 00001100
386.  000276L 00110001 00110000         TSTPT  FI,SRVREQ              ANY SERVICE REQUESTS? THEN DO THEM
387.  000277L 10100011 00110011         BRP    IDCODL,TZ              ELSE, CONTINUE MIN'ING OR MOUT'ING
388.                                                                  (I KNOW ITS ON THIS PAGE)
389. >000300L 01011001 11111111         BRAX   SRVRPT                 DO SERVICE
     >000301L 11001111 11111111
```

```
392,                                    *
393,                                    . NOTE:  FOR INSTRUCTION ENTRIES THE WAY THE PC SHOULD BE LEFT IS SPECIFIED
394,                                    *
395,  000302L                           BP:                                     ** PC LEFT AFTER THE INSTRUCTION **
396,                                    . 10,30  (    052)  BP                  BREAKPOINT
397,                                    .
398,                                    .       T <= SVBKPNT
399,                                    .       SCBP
400,                                    .
401,  000302L 01010001 00110000                 LDTI   SVBKPNT                  DEBUGGING BREAKPOINT RAM VETCOR
402,  000303L 11001111 00111010                 BRA    SCBP
403,                                    *
404,  000304L                           SC:                                     ** PC LEFT AFTER THE INSTRUCTION **
405,                                    . 10,10  (    067)  SC                  SYSTEM CALL (SUPERVISOR CALL)
406,                                    .
407,                                    .       T <= SVSCAL
408,                                    .       SCBP
409,                                    .
410,  000304L 01010001 00101010                 LDTI   SVSCAL                   SUPERVISOR CALL RAM VECTOR
411,                                    .       BRA    SCBP
412,                                    *
413,  000305L                           SCBP                                    ** UP THE PC TO NEXT INSTRUCTION **
414,                                    . 10,00                                 ** RAM VECTOR FOR INTERRUPT **
415,                                    .
416,                                    .       TEMPL <= T
417,                                    .       MAR <= MAR + 1
418,                                    .       SCPC
419,                                    .
420,  000305L 01101111 11110010                 LDRT   TEMPL                    RAM VECTOR SAVED
421,  000306L 00110111 00001100                 STB    IMAR                     CORRECT THE PC
422,  000307L 11001111 00100110                 BRA    SCPC
423,                                    *
424,  000310L                           SCROML:                                 ** ROM VECTOR ENTRY, PC ALREADY OK? **
425,                                    .                                       ** FAULT INSIDE MEMORY FAULT ROUTINE **
426,                                    .                                       ** MEMORY OR SECTOR TABLE PARITY ERROR *
427,                                    .                                       ** RESTART, DEC PC ALREADY DONE AS NEEDS
428,                                    .       TEMPL <= T
429,                                    .SCROM:
430,                                    .       TEMPH <= SYSROM>8
431,                                    .       SCDON
432,                                    .
433,  000310L 01101111 11110010                 LDRT   TEMPL
434,  000311L 01010001 11110000         SCROM:  LDRI   TEMPH,SYSROM>8
      000312L 01101111 11110001
435,  000313L 11001111 00100010                 BRA    SCDON
```

```
                                        *
000314L                 UDPOP:                          ** PC LEFT AT NEXT INSTRUCTION **
                        . 10,10  (    000)  HALT        UNDEFINED PRIVILEDGED OP
                        . 10,10  (    001)  HALT        UNDEFINED PRIVILEDGED OP
                        . 10,10  (    377)  HALT        UNDEFINED PRIVILEDGED OP
                        . 10,40 IF IN USER MODE
                        .
                        .       IF NOT USER THEN UDOP
                        .       IVIOL$
                        .
000314L 00110001 11011100               TSTIP  ,SWUSER,PSWI
000315L 01000101 00000100
000316L 11000011 00101110               BRA    UDOP,TZ          UNDEFINED IF IN USER MODE
                        *
000317L                 IVIOL$:                         ** PC LEFT AT LAST INSTRUCTION **
                        . 10,15  (11 IF IMP NON-ZERO)
                        .       T <= SVIVIOL
                        .       SCLSTI
000317L 01010001 00011110               LDTI   SVIVIOL          TRIES PRIV'D INSTRUCTION WHILE USER MODE
000320L 11001111 00101101               BRA    SCLSTW
                        *
000321L                 UDOP:                           ** PC LEFT AT LAST INSTRUCTION **
                        . 9,75   (    147)              UNDEFINED OP-CODE
                        . 9,75   (    171)              UNDEFINED OP-CODE
                        . 9,75   (    173)              UNDEFINED OP-CODE
                        . 9,75   (    175)              UNDEFINED OP-CODE
                        . 9,75   (    177)              UNDEFINED OP-CODE
                        . *2,85 IF IMP SPECIFIED
                        .
                        .       T <= SVUAINS
                        .       SCLST
                        .
000321L 01010001 00110110               LDTI   SVUAINS          UNDEFINED OP-CODE VECTOR SYSCALL
                        *
000322L                 SCLSTW                          ** RAM VECTOR INTERRUPT **
                        .                               ** 5500 I/O BUS PARITY ERROR **
                        .                               ** BACKUP PC TO START OF **
                        .                               ** INTERRUPTING INSTRUCTION **
                        .SCLSTI:
                        .       TEMPL <= T
                        .SCLST:
                        .       IF IMP ZERO THEN SCDON
                        .SCDECP:
                        .       MAR <= PC - 1
                        .       SCPC
                        .
                        .
000322L 11000100 00101101               MWAIT  ,MEMPF0
000323L 11000111 00001100
                        .
000324L 01101111 11110010       SCLSTI  LDRT   TEMPL            SAVE RAM VECTOR ADDRESS
                        .
```

```
                                        *
000325L 11001001 00100100   SCLST:  BRA    SCRAM,T0,IZ          POINTS TO START OF LAST INSTRUCTION.
                                                                 ** 1 MSEC INTERRUPT WHEN REPEATED **
                                        *
000326L                     SCDECP
                            . 9.90
000326L 00110001 11001001           DLDX   PC2MR,,DMAR          BACK UP TO IMP SPEC CODE
000327L 00110001 11101000
000330L 00110111 00001101
                                        *
000331L                     SCPC                                ** SYSTEM CALL PERFORM SECTION **
                            . 9.75
                            .           PC <= MAR
                            .
000331L 00110001 10001001           DLDX   MR2PC                SAVE P.C. AS SHOULD BE
000332L 00110001 10101000
                                        *
000333L                     SCRAM:                              ** RAM VECTOR DOESN'T WANT PC BACKUP **
                            . 9.45                              ** MEMORY, ACCESS & WRITE VIOLATIONS **
                                                                ** 1 MSEC INTERRUPT WHEN NOT REPEATED **
                            .
                            .           TEMPH <= SYSIVR>8
                            .           SCDON
                            .
000333L 01010001 11101111           LDRI   TEMPH,SYSIVR>8
000334L 01101111 11110001
                                        *
000335L                     SCDON:
                            .           STW (SYSROM>8) <= SYSROM>8  (CORRECT LAST SECTOR TABLE ENTRY)
                            .           (SP(PSW)) <= PSW            (SAVE THE PSW IN SYS SAVE AREA)
                            .           PSW <= PSW .AND. (-1-SWSCF)
                            .           IMP <= PCL
                            .           PSHST0 (PCMOD)              PUSH PC ON STACK AND GO TO TEMP
                            .
                            . 8.95      INCL.  IFETCH
                            .
000335L 01010001 11110000           LDPI   MAROH,SYSROM>8.AND.0360   CORRECT ENTRY 017 TO BE SURE
000336L 00110111 11100000
000337L 00110111 00000101           LDPT   STW                       IN THE SECTOR TABLE
000340L 01010001 11000000           LDTI   0300
000341L 00110101 11001011           DOP    SP2MRL,ND            POINT TO THE PSW SAVE BYTE
000342L 00110001 11101010           LDX    SP2MRH,DMAR
000343L 00110111 00001101
000344L 00110001 11011100           DOPIP  MODW,ND,-1-SWSCF,PSWI  CLEAR NECESSARY SYSCALL FLAGS
000345L 01010101 11011010
000346L 00110111 00000100
000347L 00110001 11011100           LDPP   MDW,PSWI              SAVE ORIGINAL PSW IN ITS SAVE AREA
000350L 00110111 00100001
000351L 01010101 11011010           DOPI   PSWO,ND,-1-SWSCF      SET NEW SYSCALL FLAGS
000352L 00110111 10001100
000353L 01010001 00001001           LDPI   LIMP,PCL              GET SET FOR PUSH OF THE P.C.
000354L 00110111 00000001
000355L 11000100 00010010           MWAIT  ,IGNORE               WAIT ON WRITE & IF ERROR, TROUBLE
```

```
                        . SYSTEM CALLS AND OP-CODES FOR IT

527:  >000356L 01011001 11111111          BRCX   PSHST0,F0,MP,PCMOD     IF NO ERROR, GO TO IT (SAVED 1 WORD)
      >000357L 01010001 11111111
      >000360L 11000110 11111111
528:                                      *
529:   000361L                           MEMPFSC
530:                                      .
531:                                      . WOW! SYSTEM SAVE AREA HAD SECTOR TABLE OR WRITE PROTECT FAULT
532:                                      .      USE THE ROM VECTOR POINT TO MARK A DOUBLE BAD INTERRUPT
533:                                      .
534:                                      .      TEMP <= SRSYSMF
535:                                      .      FETPC
536:                                      .
537:   000361L 01010001 00000110          LDTI   SRSYSMF                DO SPECIAL INTERRUPT
538:   000362L 11001111 11010011          BRA    FETPC                  FORGET PUSH, WILL PROBABLY NOT WORK
539:                                      *
540:  >000363L 01011001 11111111          MEMPF0 BRAX   MEMPFS
      >000364L 11001111 11111111
541:                                      .
542:   000365L 11111111 11111111          TABPAGE PROCL
       000366L 11111111 11111111
       000367L 11111111 11111111
       000370L 11111111 11111111
       000371L 11111111 11111111
       000372L 11111111 11111111
       000373L 11111111 11111111
       000374L 11111111 11111111
       000375L 11111111 11111111
       000376L 11111111 11111111
       000377L 11111111 11111111
```

```
545,                                    *
546,  000400L                           BETA:
547,                                    . 30,70  (    020)  BETA              SAVE ALPHA REGS AND LOAD BETA SET
548,                                    . (13,55, 17,45) IF SPLIT
549,                                    .
550,                                    . 19,55  (111 020)  BETAL             LOAD REGS FROM BETA SAVE AREA
551,                                    .        (IOD 020)  BETAL             UNSPECIFIED
552,                                    .
553,                                    . 17,65  (062 020)  SYSSAV            SAVE SYSTEM REGS
554,                                    . - ,10 IF BETA SAVED
555,                                    .        (IEV 020)  SYSSAV            UNSPECIFIED
556,                                    .
557,                                    . NOTE: CAN NOT 1MS. INTERRUPT BETWEEN BETA & FOLLOWING INSTRUCTION.
558,                                    .
559,                                    .        IF USER THEN IVIOL$
560,                                    .          ELSE IF IMP ODD OR SWRPT IN PSW THEN BETAL
561,                                    .                 ELSE IF IMP NOT ZERO THEN SYSSAV
562,                                    .                        ELSE IF SWALBT IN PSW THEN FETCHI
563,                                    .                               PSW <- PSW .OR. SWRPT
564,                                    .                 ALPHAS:
565,                                    .                               T <- SESAVAF
566,                                    .                               MODSAV
567,                                    .
568,  000400L 00110001 11011100                  TSTIP  ,SWUSER,PSWI           ONLY IF PRIV'D
      000401L 01000101 00000100
569,  000402L 11000010 00110000                  BRA    IVIOL$,FZ
570,  000403L 01000101 00100000                  TSTIT  ,SWRPT
571,  000404L 11010010 11000110                  BRA    BETAL,FZ               IF REPEATED DO SECOND HALF
572,  000405L 11011011 11000110                  BRA    BETAL,T@,IO            IF ODD, ONLY LOAD BETA REGS
573,  000406L 01000101 10000000                  TSTIT  ,SWALBT                TEST HERE, SYSSAV NEEDS IT
574,  000407L 11011000 11110010                  BRA    SYSSAV,F@,IZ           SYSTEM SAVE IF NON-ZERO
575,  000410L 11010010 00000101                  BRA    FTCHIAB,FZ             SIMPLE BETA, FORGET IT, ALREADY IN BETA
576,  000411L 01010011 00100000                  DOPI   PSW0,OR,SWRPT          TWO PARTS, SO REPEATED
      000412L 00110111 10001100
577,  000413L 01010001 00111110  ALPHAS  LDTI   SESAVAF.AND.0177               POINT TO CORRECT SAVE AREA (7 BITS!)
578,  000414L 11011111 11100100                  BRA    MODSAV
579,                                    *
580,  000415L                           SYSSAV
581,                                    . 17,65  (062 020)  SYSSAV            SYSTEM SAVE WHICHEVER REGISTERS & FLAGS
582,                                    . -0,10  IF SAVING BETA REGISTERS
583,                                    .
584,                                    .        IF SWALBT IN PSW
585,                                    .          THEN BETAS
586,                                    .          ELSE ALPHAS
587,                                    .
588,  000415L 11010011 11110100                  BRA    ALPHAS,TZ              SAVE ALL REGISTERS TO THE ALPHA SET
589,  000416L 11011111 11100101                  BRA    BETAS                  NO, TO BETA SET
```

```
590;                                    *
591;  000417L                           ALPHA:
592;                                    . 30,70  (    030)  ALPHA              SAVE BETA REGS AND LOAD ALPHA SET
593;                                    . (13.35, 17.55) IF SPLIT
594;                                    .
595;                                    . 19,65  (111 030)  ALPHAL             LOAD REGS FROM ALPHA SAVE AREA
596;                                    .        (IOD 030)  ALPHAL             UNSPECIFIED
597;                                    .
598;                                    . 27,10  (062 030)  SYSRET             RESTORE REGS AND RETURN
599;                                    .        (IEV 030)  SYSRET             UNSPECIFIED
600;                                    .
601;                                    . NOTE: CAN NOT 1MS. INTERRUPT BETWEEN ALPHA & FOLLOWING INSTRUCTION,
602;                                    . NOTE: ALPHA UNPROTECTED IN 5500 BY DEFN. OF OLD PROCESSOR MANUAL
603;                                    .
604;                                    .       IF USER THEN IVIOL$
605;                                    .         ELSE IF IMP ODD OR SWRPT IN PSW THEN ALPHAL
606;                                    .                ELSE IF IMP NOT ZERO THEN SYSRET
607;                                    .                       ELSE IF SWALBT NOT IN PSW THEN FETCHI
608;                                    .                              PSW <= PSW .OR. SWRPT
609;                                    .                BETAS:
610;                                    .                              T <= SESAVBF
611;                                    .                              MODSAV
612;                                    .
613;  000417L 00110001 11011100                 TSTIP  ,SWUSER,PSWI          ONLY IF PRIVED
      000420L 01000101 00000100
614;  000421L 11000010 00110000                BRA    IVIOL$,FZ
615;  000422L 01000101 00100000                TSTIT  ,SWRPT
616;  000423L 11010010 10100011                BRA    ALPHAL,FZ             IF REPEATED JUST DO SECOND HALF
617;  000424L 11011011 10100011                BRA    ALPHAL,T@,IO          NO SAVE, JUST DO ALPHA LOAD
618;  000425L 11011000 10111010                BRA    SYSRET,F@,IZ          RELOAD REGS & RETURN
619;  000426L 01000101 10000000                TSTIT  ,SWALBT               IS A SIMPLE ALPHA INSTRUCTION
620;  000427L 11010011 00000101                BRA    FTCHIAB,TZ            FORGET IT, ALREADY IN ALPHA MODE
621;  000430L 01010011 00100000                DOPI   PSWO,OR,SWRPT         MARK REPEATED BECAUSE 2 PARTS!
      000431L 00110111 10001100
622;  000432L 01010001 00110101     BETAS      LDTI   SESAVBF.AND.0177      (7 BITS) TO POINT TO THE SAVE AREA
623;                                .          BRA    MODSAV
```

```
624:
625:  000433L                          MODSAV
626:                                   .                                       SAVE FLAGS AND REGS IN SAVE AREA
627:                                   .
628:                                   .       MAR <= (SPH, SPIL.AND.0200.OR.T)
629:                                   .       (MAR) <= UCFLG                  (IN SYSTEM SAVE AREA SAVE FLAGS)
630:                                   .       IMP <= 8
631:                                   .       REPEAT IMP <= IMP - 1           (AND ALL REGISTERS)
632:                                   .              MAR <= MAR - 1
633:                                   .              (MAR) <= IMP
634:                                   .       UNTIL  IMP ZERO
635:                                   .       IF SWRPT NOT IN PSW THEN FETCHW   IF NO REPEATED BIT, WAS SYSS
636:                                   .         ELSE MODW <= PSW .AND. -1-SWINTE
637:                                   .              IF SRVREQ NOT ZERO THEN SRVDO
638:                                   .                ELSE IF SWALBT IN PSW
639:                                   .                       THEN ALPHAL
640:                                   .                       ELSE BETAL
641:                                   .
642:  000433L 01101111 11110010                LDRT   TEMP2                   SAVE AWAY FOR THE MOMENT
643:  000434L 00110001 11011011                DOTIP  ,ND,0200,SPIL           GET 128 BYTE BOUNDARY BIT (1 MSBIT)
      000435L 01010101 10000000
644:  000436L 01110011 11110010                DOPR   MAROL,OR,TEMP2          COMBINE WITH THE FLAG ADDRESS
      000437L 00110111 11000000
645:  000440L 00110001 11101010                LDX    SP2MRH                  MSB STACK POINTER IS SAVE AREA ALSO
646:  000441L 00110001 00110101                DOPIP  MDW,ND,0303,UCFLG       SAVE THE FLAGS, ONLY 4 BITS OF INTEREST
      000442L 01010101 11000011
      000443L 00110111 00100001
647:  000444L 01010001 00010000                LDPI   LIMP,IMP8               INIT COUNT
      000445L 00110111 00000001
648:  000446L 00110111 00001001     MODSVLP    STB    DIMP
649:  000447L 00110001 11011111                LDTP   IMPI                    SPEEDUP, GET DATA NOW
650:  000450L 11010100 11010111                MWAIT  ,MEMPF0
      000451L 11000111 00001100
651:  000452L 00110111 00001101                STB    DMAR                    SAVE THE REGISTER
652:  000453L 00110111 00100001                LDPT   MDW
653:  000454L 11011000 11011001                BRA    MODSVLP,F0,IZ           LOOP TILL ALL 8 REGS. ARE SAVED
654:  000455L 00110001 11011100                TSTIP  ,SWRPT,PSWI             WELL, ONE OR TWO PARTS?
      000456L 01000101 00100000
655:                                   . 11,15
656:  000457L 11010011 01100110                BRA    FTCHABW,TZ              ONLY SYSSAV, NO-LOAD NEW REGS,
657:  000460L 01010101 11111110                DOPI   MODW,ND,-1-SWINTE       TEMPORARILY DISABLE INTERRUPTS FOR IT
      000461L 00110111 00000100
658:                                   .
659:                                   . INLINE RINST CODE!!!
660:                                   .
661:  000462L 00110001 00110000                TSTPT  FI,SRVREQ               ANY REQUESTS PENDING?
662:  000463L 11010010 10111100                BRA    MODSRV,FZ               YES, DO THEM AND I HOPE NO TROUBLE
663:  000464L 00110001 11011100                TSTIP  ,SWALBT,PSWI            WHAT WAS IT?
      000465L 01000101 10000000
664:  000466L 11010100 11001001                MWAIT  ,MEMPF0                 NEEDED AFTER MDW AT END OF LOOP ABOVE
      000467L 11000111 00001100
665:  000470L 11010010 10100011                BRA    ALPHAL,FZ               WAS BETA, DO ALPHAL NOW
```

```
666.            .       BRA     BETAL                   WAS ALPHA DO BETAL NOW
```

```
667,
668,  000471L                          +
669,                                   BETAL
670,                                   . 19,55  (111 020)  BETAL              LOAD REGS FROM BETA SAVE AREA
671,                                   .        (IOD 020)  BETAL              UNSPECIFIED
672,                                   .
673,                                   . 27,10  (062 030)  SYSRET             ONLY IF TO BETA MODE
674,                                   .
675,                                   .        MAR <= SPH, (SPL .AND. 300) .OR. SESAVBF
676,                                   .        T <= PSW .OR. SWALBT .AND. -1-SWRPT
677,                                   .        MODLOD
678,                                   . 14,60  TO FETCH (NO FETCH)
679,  000471L 00110001 11011011         .
      000472L 01010101 10000000                DOTIP  ,ND,0200,SPIL
680,  000473L 01010011 00110101                DOPI   MAROL,OR,SESAVBF,AND,0177    POINT TO BETA FLAGS IN SAVE AREA
      000474L 00110111 11000000
681,  000475L 00110001 11101010                LDX    SP2MRH,SMR
      000476L 00110111 01000111
682,  000477L 00110001 11011100                DOTIP  ,OR,SWALBT,PSWI         SET BETA MODE
      000500L 01010011 10000000
683,  000501L 01010101 11011111                DOTI   ,ND,-1-SWRPT            RESET REPEAT FLAG (IF SET)
684,  000502L 11011111 10011011                BRA    MODLOD
685,                                   *
686, >000503L 01011001 11111111       MODSRV   BRAX   SRVDO                  GO TO SERVICE AS NEEDED (MODSAV USES IT)
     >000504L 11001111 11111111
687,                                   *
688,  000505L                          SYSRET
689,                                   .*      NOTE: THIS INSTRUCTION IS VERY LONG IN EXECUTION
690,                                   .*            IF DMA SLOWS IT EVEN FARTHER WE MAY HAVE PROBLEMS
691,                                   .
692,                                   .* WARNING:   IF AS SOLUTION TO LENGTH IT IS DONE AS REPEATED INSTRUCTON
693,                                   .*    THEN:   IF INTERUPTED (I.E. BY RESTART-DEBUG INTERRUPT)
694,                                   .*            THE PC WILL REFLECT THE ADDRESS OF THE INSTRUCTION TO RETURN
695,                                   .*            TO, WITH REPEATED FLAG ON AND THE INCORRECT REGISTER SET IN THE
696,                                   .*            SUPPORT REGISTERS (THE ALPHAL OR BETAL NOT COMPLETED!)
697,                                   .*            THIS IS VERY BAD!  BUT POSSIBILITY IS LOW AND NO SOLUTION SEEN
698,                                   .
699,                                   .* WARNING:   IF INSIDE SYSRET (REGISTER LOAD PART) THERE IS A MEMORY FAULT,
700,                                   .*            THE PC WILL BE INCORRECT BECAUSE OF A SPECIAL PRE-DECREMENT.
701,                                   .*            IF THE IMP WAS NON-ZERO, IT WILL BE EVEN FARTHER OFF
702,                                   .
703,                                   . 27,?0  (062 030)  SYSRET             SYSTEM RETURN (FROM SC, BP, ETC)
704,                                   .        (IMP 030)                     UNSPECIFIED
705,                                   .
706,                                   .        IMP <= PCL                    POP THE RETURN ADDRESS INTO THE PC
707,                                   .        POPST0 (SYSRET)
708,                                   . SYSRET:
709,                                   .        PC <= PC -1
710,                                   .        PSW <= (SP (PSW))
711,                                   .        IF SWALBT THEN BETAL
712,                                   .          ESLE ALPHAL
713,                                   .
```

```
714.  000505L 01010001 00001001          LDPI   LIMP,PCL              GET THE RETURN ADDRESS INTO THE P.C.
      000506L 00110111 00000001
715. >000507L 01011001 11111111          BRCX   POPST0,,SYSRET0        OFF PAGE WITH SPECIAL RETURN
     >000510L 01010001 11111111
     >000511L 11001111 11111111
716.
717.  000512L 00110001 11001001  .
      000513L 00110001 11101000  SYSRET1: DLDX   PC2MR,,DMAR           CORRECT FOR IMAR IN FETCH CODE
      000514L 00110111 00001101
718.  000515L 00110001 10001001          DLDX   MR2PC
      000516L 00110001 10101000
719.  000517L 00110001 11101010          LDX    SP2MRH                 GET THE PSW
720.  000520L 01010001 11000000          LDTI   0300
721.  000521L 00110101 11001011          DOP    SP2MRL,ND              TO BE LOADED
722.  000522L 00110111 00001101          STB    DMAR,SMR               FROM THE SYSTEM SAVE AREA
      000523L 00110111 01000111
723.  000524L 11001110 11111111          MWAIT  NOOP,MEMPF0
      000525L 11010100 10101010
      000526L 11000111 00001100
724.  000527L 00110001 00110110          LDPP   PSW0,MDR,MODW           GET NEW PSW & SET IT EVEN IF USER MODE!
      000530L 00110111 10001100
      000531L 00110111 00000100
725.                                     .                              MODW MAY CHANGE BASE EN/DIS BIT
726.                                     .                              BUT DO IT NOW FOR BASED STACK AREA??
727.                                     .                              NO PROBLEM, IF PROTECTED FIRST POP/PUSH
728.                                     .                              WOULD FAIL ANYWAY
729.  000532L 01000101 10000000          .      TSTIT  ,SWALBT          WHICH WAY TO GO?
730.  000533L 11010010 11000110          .      BRA    BETAL,FZ         TO BETA
731.                                     .      BRA    ALPHAL           TO ALPHA
732.                                     *
733.  000534L                           ALPHAL
734.                                     . 19,65  (111 030)  ALPHAL      LOAD REGISTERS FROM ALPHA SAVE AREA
735.                                     .        (IOD 030)  ALPHAL      UNSPECIFIED
736.                                     .
737.                                     . 27,?0  (062 030)  SYSRET      ONLY IF IN ALPHA MODE
738.                                     .
739.                                     .        MAR <= SPH, (SPIL .AND. 300) .OR. SESAVAF
740.                                     .        T <= PSW .AND. =1 - SWRPT - SWALBT
741.                                     .        MODLOD
742.                                     . 14,70  TO FETCH (NO FETCH)
743.                                     .
744.  000534L 00110001 11011011          .      DOTIP  ,ND,0200,SPIL    GET IT ON 128 BYTE BOUNDARY
      000535L 01010101 10000000
745.  000536L 01010011 00111110          .      DOPI   MAR0L,OR,SESAVAF.AND.0177  ONLY 7 BITS OF INTEREST
      000537L 00110111 11000000
746.  000540L 00110001 11101010          .      LDX    SP2MRH,SMR
      000541L 00110111 01000111
747.  000542L 00110001 11011100          .      DOTIP  ,ND,-1-SWRPT-SWALBT,PSWI  RESET REPEAT FLAG AND SET ALPHA
      000543L 01010101 01011111
748.                                     .      BRA    MODLOD
```

```
749:
750:  000544L                          *
                                       MODLOD
751:                                   .
                                                                LOAD USERS FLAGS AND REGISTERS
752:                                   . NOTE: IF MACRO LEVEL PLAYS WITH FLAG BYTE, IT BETTER DO IT CORRECTLY
753:                                   .
754:                                   .       PSW <= T
755:                                   .       LUF <= (MDR) + (MDR)
756:                                   .       MAR <= MAR - 1
757:                                   .       MODLOP
758:                                   .
759:  000544L 00110111 10001100               LDPT   PSWO                    LOAD STATUS WORD
760:  000545L 00010111 10110010               CCLR
761:  000546L 11010100 10011001               MWAIT  ,MEMPF0
      000547L 11000111 00001100
762:  000550L 00110111 00000100               LDPT   MODW
763:  000551L 00110001 00110110               DOTPP  ,AC,MDR,MDR             SET USER FLAGS
      000552L 00110010 00110110
764:  000553L 00110111 00001101               STB    DMAR,SMR                POSITION CORRECTLY
      000554L 00110111 01000111
765:  000555L 00110111 00000110               STB    LUF
766:  000556L 11011111 01100001               BRA    MODLOP                  AND DO THE REGISTER LOAD
```

```
                        . REGISTER SAVE AND RESTORE
                        *
                        REGS:
                        . 19.40  (    055)  REGS              REGISTER SAVE
                        .
                        . 16.95  (111 055)  REGL              REGISTER LOAD
                        .        (IMP 055)  REGL              REGISTER LOAD UNSPECIFIED
                        .
                        .        IF IMP NOT ZERO THEN REGL
                        .          ELSE IMP <= 8
                        .               MAR <= (SP, SP+1) + 1
                        .               REPEAT IMP <= IMP - 1
                        .                      MAR <= MAR - 1
                        .                      (MAR) <= IMP
                        .               UNTIL IMP ZERO
                        .               (SP, SP+1) <= MAR - 1
                        .               FETCHW
                        .
000557L 11011000 01100100       BRA     REGL,F0,IZ              DO REGISTER LOAD NOT SAVE!
000560L 00110001 11001011       DLDX    SP2MR,,SMR              GET TOP ENTRY ON THE STACK
000561L 00110001 11101010
000562L 00110111 01000111
000563L 01010001 00010000       LDPI    LIMP,IMP8               INITIALIZE COUNTER
000564L 00110111 00000001
000565L 11010100 10001010       MWAIT   ,MEMPF0
000566L 11000111 00001100
000567L 00110001 00110110       LDTP    MDR                     GET LSB
000570L 00110111 00001100       STB     IMAR,SMR
000571L 00110111 01000111
000572L 01101111 11110010       LDRT    TEMPL                   HOLD IT
000573L 11010100 10000100       MWAIT   ,MEMPF0
000574L 11000111 00001100
000575L 00110001 00110110       LDPP    MAROH,MDR               GET MSB
000576L 00110111 11100000
000577L 01110001 11110010       LDPR    MAROL,TEMPL             PUT BOTH IN THE MAR
000600L 00110111 11000000
000601L 00110111 00001100       STB     IMAR                    (EXTRA CODE MAKES IT FASTER)
000602L 00110111 00001001 REGSLP STB    DIMP                    COUNT DOWN
000603L 00110001 11011111       LDTP    IMPI                    GET NEXT REGISTER
000604L 11010100 01111011       MWAIT   ,MEMPF0
000605L 11000111 00001100
000606L 00110111 00001101       STB     DMAR                    POINT TO THE NEXT SAVE LOCN,
000607L 00110111 00100001       LDPT    MDW                     AND OUTPUT THE REGISTER
000610L 11011000 01111101       BRA     REGSLP,F0,IZ            CONTINUE THROUGH LOOP TILL DONE
000611L 11010100 01110110       MWAIT   ,MEMPF0
000612L 11000111 00001100
000613L 00110111 00001101       STB     DMAR                    (EXTRA CODE MAKES IT FASTER)
000614L 00110001 10010000       DLDRP   TEMP,MARI               MOVE THE MAR OUT OF THE WAY
000615L 01101111 11110010
000616L 00110001 10110000
000617L 01101111 11110001
000620L 00110001 11001011       DLDX    SP2MR                   PUT STACK POINTER THERE
000621L 00110001 11101010
```

```
                . REGISTER SAVE AND RESTORE

807:  000622L 01110001 11110010          LDPR   MDW,TEMPL            SAVE NEW TOP OF STACK ENTRY LSB
      000623L 00110111 00100001
808:  000624L 01110001 11110001          LDTR   TEMPH                GET SET FOR MSB
809:  000625L 11010100 01101010          MWAIT  ,MEMPF0
      000626L 11000111 00001100
810:  000627L 00110111 00001100          STB    IMAR                 POINT TO THE MSB
811:  000630L 00110111 00100001          LDPT   MDW                  SAVE IT
812: >000631L 01011001 11111111 FTCHABW  BRAX   FETCHW
     >000632L 11001111 11111111
813:                             .
```

```
                                . REGISTER SAVE AND RESTORE
814.                            +
815.  000633L                   REGL
816.                            . 16,95  (111 055)  REGL              REGISTER LOAD
817.                            .        (IMP 055)  REGL              REGISTER LOAD UNSPECIFIED
818.                            .
819.                            .        MAR <= HL
820.                            . MODLOD:
821.                            .        IMP <= 8
822.                            .        REPEAT IMP <= IMP - 1
823.                            .               IMP <= (MAR)
824.                            .               MAR <= MAR - 1
825.                            .        UNTIL IMP ZERO
826.                            .        FETCHW
827.                            .
828.  000633L 00110001 11000110         DLDX   HL2MR,,SMR            INIT HL TO MAR AND GET FIRST REG (X)
      000634L 00110001 11100101
      000635L 00110111 01000111
829.  000636L 01010001 00010000 MODLOP  LDPI   LIMP,IMP8              INIT COUNTER
      000637L 00110111 00000001
830.  000640L 00110111 00001001 REGLLP  STB    DIMP                  COUNT DOWN
831.  000641L 11010100 01011110         MWAIT  ,MEMPF0
      000642L 11000111 00001100
832.  000643L 00110001 00110110         LDTP   MDR                   GET DATA AND START NEXT READ FAST!
833.  000644L 00110111 00001101         STB    DMAR,SMR
      000645L 00110111 01000111
834.  000646L 00110111 10001111         LDPT   IMP0                  SAVE THE DATA IN ITS REGISTER
835.  000647L 11011000 01011111         BRA    REGLLP,F0,IZ          LOOP FOR THE FULL COUNT
836. >000650L 01011001 11111111         BRAX   FETCHW
     >000651L 11001111 11111111
837.                            .
```

```
                                . SECTOR TABLE LOAD
840,                            *
841,  000652L                   STL:
842,                            .       (    077)  STL                  LOAD THE SECTOR TABLE
843,                            . 7.90 + C * 2.5 + 0.55 IF C > 8  (OR 4.45 IF C = 0)
844,                            .      + 5.05 IF SERVICE
845,                            .
846,                            .       (022 077)  STLOA                LOAD SECTOR TABLE STARTING FROM A
847,                            .       (111 077)  STLOB                LOAD SECTOR TABLE STARTING FROM B
848,                            .       (062 077)  STLOC                LOAD SECTOR TABLE STARTING FROM C
849,                            .       (113 077)  STLOD                LOAD SECTOR TABLE STARTING FROM D
850,                            .       (174 077)  STLOE                LOAD SECTOR TABLE STARTING FROM E
851,                            .                                       (IN ABOVE, HIGH 4 BITS SELECT!)
852,                            . 10.25 + C * 2.5 + 0.55 IF C > 8  (OR 6.45 IF C = 0)
853,                            .      + 5.05 IF SERVICE
854,                            .
855,                            .       (IMP 077)                       UNSPECIFIED
856,                            .
857,                            .       NOTE; DOES NOT CHANGE HL OR C-REGISTERS
858,                            .       NOTE: IN STLO'C', LOW 4 BITS COUNTER & HIGH 4 BITS OFFSET!
859,                            .
860,                            .
861.  000652L 00110001 11011100         TSTIP  ,SWUSER,PSWI             ONLY IF PRIVED
      000653L 01000101 00000100
862,  000654L 11000010 00110000         BRA    IVIOLS,FZ
863,  000655L 01010101 00100000         TSTIT  ND,SWRPT,,,TW            ASSUME WILL BE ZERO, NOT REPEATED
864,  000656L 11010011 01000111         BRA    STLNRPT,TZ               WAS NOT REPEATED
865,                            .
866.  000657L 00110001 11010010         DORIP  TEMP2,ND,07,URI+URC      REPEATED, GET THE HL (MAR) BIAS AND
      000660L 01010101 00000111
      000661L 01101111 11110010
867,                                    RPT    4
868,  000662L 00010111 10100010         SHIFT  SL                       THE POINTER BIAS AND COUNT FOR THE
868,  000663L 00010111 10100010         SHIFT  SL                       THE POINTER BIAS AND COUNT FOR THE
868,  000664L 00010111 10100010         SHIFT  SL                       THE POINTER BIAS AND COUNT FOR THE
868,  000665L 00010111 10100010         SHIFT  SL                       THE POINTER BIAS AND COUNT FOR THE
869,  000666L 01010011 00001000         DOTI   ,OR,010                  FINAL EIGHT STL LOADS
870,  000667L 11011111 01000011         BRA    STLRPTD
871,                            .
872,  000670L 01101111 11110010  STLNRPT LDRT   TEMP2                    REGULAR HL (MAR) NEEDS NO BIAS
873.  000671L 00110001 11010010         TSTIP  ,017,URI+URC,TW          GET THE COUNT OF THE STEPS TO DO
      000672L 01010101 00001111
874,  000673L 11010011 00000101         BRA    FTCHIAB,TZ               IF ZERO, WAS NOTHING TO DO. SO, THE END!
875,                            .
876,  000674L 11011001 00111110  STLRPTD BRA    STLZRO,T0,IZ             DO STANDARD STL
877,  000675L 01101111 11110001         LDRT   TEMP1                    SAVE AWAY
878.  000676L 00110001 11011111         DOTIP  ,ND,0360,IMPI            IMP STL STARTS STL WITH OFFSET
      000677L 01010101 11110000
879,  000700L 01110010 00110001         DOTR   ,AC,TEMP1,,C0            COMBINE GIVEN OFFSET WITH BIAS
880,                            .
881,  000701L 01101111 11110001  STLZRO  LDRT   TEMP1                    SAVE THE POINTER/COUNTER AS GENERATED
882.  000702L 00110001 11010110         DOPRP  MAROL,AC,TEMP2,URI+URL,C0  BIAS TABLE ADDRESS (IF NEEDED)
      000703L 01110010 00110010
```

```
                000704L 00110111 11000000
   883:         000705L 00110001 11010101          LDRP    TEMP2,URI+URH           SAVE AWAY URH SO CAN BE RESTORED
                000706L 01101111 11110010
   884:         000707L 00110110 11100000          DOP     MAR0H,IT
   885:         000710L 00110111 01000111          STB     SMR                     GET NEXT (FIRST) TABLE ENTRY
   886:         000711L 00110111 10000101          LDPT    URO+URH                 IN CASE TABLE CROSSES PAGE BOUNDARY
   887:                                             .
   888:         000712L 01110001 11110001  STLAGN   LDTR    TEMP1                   GET THE INITIAL VIRTUAL ADDRESS
   889:                                             .
   890:         000713L 11010100 00110100  STLOOP   MWAIT   ,STLDONE               RESTORE REGS & SET VECTOR FIRST
                000714L 11010111 00011101
   891:         000715L 00110111 11100000          LDPT    MAR0H                   LOAD VIRTUAL ADDRESS FROM TEMP
   892:         000716L 01010001 00000010          LDPI    MODW,SWBASD             DISABLE BASING (AND 1 MS INTERRUPTS)
                000717L 00110111 00000100
   893:         000720L 00110001 00110110          DOPIP   STW,ND,0375,MDR         SET PHYSICAL SECTOR # IN VIRTUAL LOCN.
                000721L 01010101 11111101
                000722L 00110111 00000101
   894:         000723L 00110001 11011100          LDPP    MODW,PSWI               RESTORE BASING (1 MS INT. 1 INST. DELAY)
                000724L 00110111 00000100
   895:         000725L 00110001 11100101          LDX     HL2MRH                  RESTORE MARH FROM H SAVE AREA
   896:         000726L 00110111 00001100          STB     IMAR,SMR                GET NEXT TABLE ENTRY
                000727L 00110111 01000111
   897:         000730L 00110001 10100101          LDX     MR2HLH                  SAVE MARH IN H
   898:         000731L 01110001 10110001          DORIR   TEMP1,AC,017,TEMP1,CC   INC VIRTUAL SECTOR # & DECR COUNT
                000732L 01010010 00001111
                000733L 01101111 11110001
   899:         000734L 01000101 00000111          TSTIT   ,07                     AT SPLIT POINT?
   900:         000735L 11010010 00110100          BRA     STLOOP,FZ               NO, CONTINUE
   901:                                             .                               LOOPING JUMP IS DURING MEMORY READ
   902:                                             .                               THIS MAKES EXEC FASTER.
   903:         000736L 01000101 00001111          TSTIT   ,017                    REALLY FINISHED?
   904:         000737L 11010011 00011101          BRA     STLDONE,TZ              YES, CLEANUP AND END
   905:         000740L 00110001 00110000          TSTPT   FI,SRVREQ               NO, ANY SERVICE TO DO?
   906:         000741L 11010011 00110101          BRA     STLAGN,TZ               NO, THEN DO SECOND HALF NOW
   907:                                             .                               REPEATED! WILL DO SERVICE BEFORE SECOND
   908:                                             .                               PART BUT DO CLEANUP FIRST
   909:         000742L 11010100 00011101  STLDONE  MWAIT   ,$+1                   RESTORE BEFORE MEMPF
                000743L 11010111 00011011
   910:                                             .                               BUT NOTE! MAR MSB WILL BECOME INCORRECT
   911:         000744L 01110001 11110010          LDPR    URO+URH,TEMP2           RESTORE H-REG
                000745L 00110111 10000101
   912:         000746L 01010001 11110000          LDPI    MAR0H,SYSROM>8.AND.0360  ** FORCE LAST ENTRY TO
                000747L 00110111 11100000
   913:         000750L 00110111 00000101          LDPT    STW                     ACCESS PROTECT AND WRITE PROTECT
   914:         000751L 01011001 11111101          BPGX    RIND
   915:         000752L 01110001 11110001          TSTIR   ,017,TEMP1              REALLY FINISHED?
                000753L 01000101 00001111
   916:         000754L 11000011 01110010          BRA     RIND,TZ                 YES, RESET REPEAT FLAG
   917:        >000755L 01011001 11111111          BRAX    SRVRPT                  NO, DO IT AS REPEATED INST.
               >000756L 11001111 11111111
   918:                                             .                               (BOTH RIND & SRVRPT INDIRECT TO MEMPF)
   919:                                             .
```

```
                        . SYSTAT INSTRUCTION   (157 OP-CODE)
 922,                   *
 923,  000757L          SYSTAT:
 924,                   .
 925,                   . 10,25   (    157)  UDOP              UNDEFINED OP-CODE
 926,                   . 5,35    (111 157)  SYSTAT1           PROCESSOR STATUS WORD
 927,                   . 5,75    (062 157)  SYSTAT2           SELECTABLE NODE I.D.
 928,                   . 5,95    (113 157)  SYSTAT3           SYSTEM STATUS (UNUSED - 0)
 929,                   . 5,95    (174 157)  SYSTAT4           SYSTEM STATUS (UNUSED - 0)
 930,                   . 5,95    (115 157)  SYSTAT5           SYSTEM STATUS (UNUSED - 0)
 931,                   . 5,95    (176 157)  SYSTAT6           SYSTEM STATUS (UNUSED - 0)
 932,                   . 5,95    (117 157)  SYSTAT7           SYSTEM STATUS (UNUSED - 0)
 933,                   . 5,95    (022 157)  SYSTAT8           SYSTEM STATUS (UNUSED - 0)
 934,                   .
 935,                   .         CASE IMP OF
 936,                   .         0:   UDOP;
 937,                   .         1:   B <= PSW;
 938,                   .         2:   B <= SNID;
 939,                   .         3..8:B <= 0;
 940,                   .         END CASE
 941,                   .         C <= 0
 942,                   .         FETCHI
 943,                   .
 944,  000757L 11001001 00101110          BRA    UDOP,T0,IZ          157 ALONE IS UNDEFINED
 945,  000760L 00110111 00001001          STB    DIMP                CHECK 111
 946,  000761L 01010001 00000000          LDPI   URO+URC,0           WHATEVER ELSE, C=0
       000762L 00110111 10000010
 947,  000763L 00110001 11011100          LDTP   PSWI                (ASSUME 1)
 948,  000764L 11011001 00000110          BRA    SYSTND,T0,IZ
 949,  000765L 00110111 00001001          STB    DIMP                CHECK 062
 950,  000766L 00110001 01010001          LDTP   SNID
 951,  000767L 11011001 00000110          BRA    SYSTND,T0,IZ
 952,  000770L 01010001 00000000          TCLR                       WILL BE 113 - 022
 953,  000771L 00110111 10000001  SYSTND  LDPT   URO+URB
 954,  000772L                    FTCHIAB                           (FINISHED ALPHA/BETA THAT DID NOTHING)
 955,>000772L 01011001 11111111          BRAX   FETCHI
      >000773L 11001111 11111111
 956,  000774L 11111111 11111111          TABPAGE PROCL
       000775L 11111111 11111111
       000776L 11111111 11111111
       000777L 11111111 11111111
```

```
                        . BLOCK TRANSFER AND CONVERT
                        *
                        BT:
                        .       (    021)  BT                  BLOCK TRANSFER
                        . 2.25 + N * 5.55 - N * 0.15 / ODDSTEP - 0.5 IF MATCH
                        .
                        .       (111 021)  BTR                 BLOCK TRANSFER REVERSE
                        . 4.25 + N * 5.75 - N * 0.15 / ODDSTEP - 0.5 IF MATCH
                        .
                        .       (062 021)  BCV                 BLOCK CONVERT
                        . 4.25 + N * 7.35 - N * 0.15 / ODDSTEP - 0.5 IF MATCH
                        .
                        .       (IOD 021)  BTR                 BTR UNSPECIFIED
                        .       (IEV 021)  BCV                 BCV UNSPECIFIED
                        .
                        .       NOTE:  NONE OF THESE CHANGE THE CONDITION FLAGS
                        .
                        .       IF IMP ZERO THEN BTX
                        .         ELSE IF IMP ODD THEN BTR
                        .                ELSE MARL <- A <- L + (DE)     SET CARRY ON OVERFLOW
                        .                     MARH <- H + C
                        .                     TEMP1 <- 0
                        .                     BTCVT
                        .
                        . BTX:  MAR <- HL
                        .       HL <- HL + 1
                        .       TEMP1 <- A
                        . BTCVT: TEMP1 <- (DE) <- TEMP1 + (MAR)
                        .       DE <- DE + 1
                        .       BTEND
                        .
                        . BTR:  MAR <- HL
                        .       HL <- HL - 1
                        .       TEMP1 <- A
                        .       TEMP1 <- (DE) <- TEMP1 + (MAR)
                        .       DE <- DE - 1
                        .
                        . BTEND: IF (TEMP1 + B) ZERO AND CARRY THEN RIN256
                        .         ELSE RIN256
                        .
001000L 11001001 11101110       BRA    BTX,T0,IZ               BT (021)
001001L 11001011 11010110       BRA    BTR,T0,IO               BTR (111 021)
001002L 00110001 11000100       DLDX   DE2MR,,SMR              BCV (062 021)
001003L 00110001 11100011
001004L 00110111 01000111
001005L 00010111 10110010       CCLR
001006L 00110001 11010110       LDTP   URI+URL                 ADD DATA AT DE TO TABLE POINTER IN HL
001007L 11000100 11111000       MWAIT  ,MEMPF1
001010L 11010111 00001001
001011L 00110010 00110110       DOPP   MAROL,AC,MDR              A <- MAR <- L + (DE)
001012L 00110111 11000000
001013L 00110111 10000000       LDPT   URO+URA                 STORE IN A SO CAN FIND STOP POINT
001014L 00110001 11010101       LDTP   URI+URH                   ADD IN CARRY ALSO
```

```
1007'  001015L 00110110 11100000          DOP    MAR0H,IT
1008'  001016L 00110111 01000111          STB    SMR                    INDIRECT THROUGH THIS FOR TABLE ENTRY
1009'  001017L 01010001 00000000          TCLR                          NO A-REG OFFSETS
1010'  001020L 11001111 11100101          BRA    BTCVT
1011'                                     *
1012'  001021L 00110001 11000110  BTX     DLDX   HL2MR,,SMR             GET DATA AT HL TO BE MOVED
       001022L 00110001 11100101
       001023L 00110111 01000111
1013'  001024L 00110001 10010000          DADDP  UR0+UR,MARI            INCREMENT HL FOR BT
       001025L 00010110 01110010
       001026L 00110111 10000110
       001027L 00110001 10110000
       001030L 00110110 10000101
1014'  001031L 00110001 11010000          LDTP   URI+URA
1015'  001032L 01101111 11110001  BTCVT   LDRT   TEMP1                 SAVE THE A-REG OFFSET (ZERO FOR BCV)
1016'  001033L 11000100 11100100          MWAIT  ,MEMPF1
       001034L 11010111 00001001
1017'  001035L 00110001 11000100          DLDX   DE2MR                  POINT TO WHERE TO STORE RESULT
       001036L 00110001 11100011
1018'  001037L 00110001 00110110          DOPRP  MDW,AC,TEMP1,MDR,C0    STORE DATA IN (DE) AND IN TEMP1
       001040L 01110010 00110001
       001041L 00110111 00100001
1019'  001042L 01101111 11110001          LDRT   TEMP1
1020'  001043L 00110001 10010000          DADDP  UR0+UDE,MARI           INCREMENT DE FOR BT & BCV
       001044L 00010110 01110010
       001045L 00110111 10000100
       001046L 00110001 10110000
       001047L 00110110 10000011
1021'  001050L 11001111 10111100          BRA    BTEND
1022'                                     *
1023'  001051L 00110001 11000110  BTR     DLDX   HL2MR,,SMR             GET DATA AT HL TO BE MOVED
       001052L 00110001 11100101
       001053L 00110111 01000111
1024'  001054L 00010111 10110010          CCLR
1025'  001055L 00110001 10010000          DDECP  UR0+UR,MARI            DECREMENT HL FOR BTR
       001056L 01010100 00000001
       001057L 00110111 10000110
       001060L 00110001 10110000
       001061L 01010100 00000000
       001062L 00110111 10000101
1026'  001063L 00110001 11010000          LDRP   TEMP1,URI+URA          LOAD UP THE A-REG OFFSET
       001064L 01101111 11110001
1027'  001065L 11000100 11001010          MWAIT  ,MEMPF1
       001066L 11010111 00001001
1028'  001067L 00110001 11000100          DLDX   DE2MR
       001070L 00110001 11100011
1029'  001071L 00110001 00110110          DOPRP  MDW,AC,TEMP1,MDR,C0    GET RESULTANT DATA
       001072L 01110010 00110001
       001073L 00110111 00100001
1030'  001074L 01101111 10110001          LDRT   TEMP1,,CC              STORE DATA IN (DE)
1031'  001075L 00110001 10010000          DDECP  UR0+UDE,MARI           BACKUP DE FOR BTR
       001076L 01010100 00000001
```

```
                001077L 00110111 10000100
                001100L 00110001 10110000
                001101L 01010100 00000000
                001102L 00110111 10000011
  1032.                         *
  1033.                         . BT - 3.50, BTR - 3.70, BCV - 5.30
  1034.                         .
  1035.  001103L 00110001 11010001  BTEND   TSTRP  AC,TEMP1,URI+URB,C0   AT THE ENDPOINT?
         001104L 01110010 00000001
  1036.  001105L 11000000 10110111          BRA    RIN256,FC            NO, WANT CARRY TRUE & ZERO TRUE
  1037.  001106L 11000011 01110010          BRA    RIND,TZ              YES, END NOW!
  1038.                         .          BRA    RIN256               NO!
  1039.                         .                                      NO, REPEAT 256 TIMES (OR LESS)
  1040.  001107L 00010111 10110010          CCLR                        ** ONLY CARRY SET RIN256 CASE **
```

```
                                . REPEATED INSTRUCTIONS, REPEAT CONTROL SECTION
1043,                           *
1044,  001110L                  RIN256:
1045,                           . ODD - 1.45, RPT - 1.60, SRV - 1.35, END - 3.85
1046,                           .
1047,                           .       WARNING, CARRY MUST BE CLEAR ON ENTRY
1048,                           .       BLOCK REPEATED INSTRUCTIONS (THOSE THAT COUNT C THROUGH 256 MAX STEPS)
1049,                           .       DOES NOT CHANGE FLAGS
1050,                           .
1051,                           .       URC <- URC - 1
1052,                           .       IF ODD THEN RINDO               EVERY OTHER STEP HAS NO SERVICE
1053,                           .       IF ZERO THEN RIND
1054,                           .         ELSE RINST
1055,                           .
1056.  001110L 00110001 11010010        DOPIP  URO+URC,SB,1,URI+URC     IF ZERO, FINISHED
       001111L 01010100 00000001
       001112L 00110111 10000010
1057,  001113L 11000011 01110010        BRA    RIND,TZ
1058,  001114L 01000101 00000001        TSTIT  ,1                       DO THEM IN PAIRS SO CAN'T INTERRUPT
1059,  001115L 11000010 10101111        BRA    RINDO,FZ                 DURING DOUBLE BYTE MOVE
1060,                           .                                       !CAN USE BT IN DISPLAY POINTERS MOVES!
1061,                           *
1062,  001116L                  RINST:
1063,                           . SRV - 0.65, RPT - 0.90
1064,                           .
1065,                           .       LOOP THROUGH REPEATED INSTRUCTIONS, SET THE REPEAT MODE AND DO
1066,                           .       SERVICE REQUEST IF NEEDED ELSE RE-EXECUTE THE INSTRUCTION.
1067,                           .
1068,                           .       IF SRVREQ NOT ZERO THEN SRVRPT
1069,                           . RINDO:
1070,                           .         ELSE IDCOD (IREG)
1071,                           .
1072,  001116L 00110001 00110000        TSTPT  FI,SRVREQ                DO SERVICE IF NEEDED
1073,  001117L 11000010 10101011        BRA    RINRPT,FZ
1074.  001120L 11000100 10101111 RINDO  MWAIT  ,MEMPF1
       001121L 11010111 00001001
1075.  001122L 00111001 00110100        BRPX   IDCOD                    GO DO OPCODE AGAIN
       001123L 10101111 00110011
1076,                           .
1077.  >001124L 01011001 11111111 RINRPT BRAX   SRVRPT                  YES, DO IT!
       >001125L 11001111 11111111
```

```
                                . BLOCK COMPARE
1080,                           *
1081,  001126L                  BCP:
1082,                           .       (    041)  BCP                  BLOCK COMPARE
1083,                           . 2.15 + N * 5.35 - 0.15 * N / ODDSTEP - 0.5 IF MATCH
1084,                           .
1085,                           .       (111 041)  DFAC                 DECIMAL FIELD ADD
1086,                           . 4.25 + C * 6.95
1087,                           .       (IOD 041)  DFAC                 DFAC UNSPECIFIED
1088,                           . 4.25 + C * 6.85
1089,                           .
1090,                           .       (062 041)  DFSB                 DECIMAL FIELD SUBTRACT
1091,                           .       (IEV 041)  DFSB                 DFSB UNSPECIFIED
1092,                           .
1093,                           .       IF IMP NOT ZERO THEN DFOP
1094,                           .         ELSE LUF <= T <= (DE) - (HL)
1095,                           .              DE <= DE + 1
1096,                           .              HL <= HL + 1
1097,                           .              IF T ZERO THEN RIN256
1098,                           .                ELSE RIND
1099,                           .
1100,  001126L 11001000 10001101        BRA    DFOP,F0,IZ               GO TO DECIMAL STUFF
1101.  001127L 00110001 11000100        DLDX   DE2MR,,SMR
       001130L 00110001 11100011
       001131L 00110111 01000111
1102.  001132L 00110001 10010000        DADDP  URO+UDE,MARI             GET DATA AND UPDATE DE
       001133L 00010110 01110010
       001134L 00110111 10000100
       001135L 00110001 10110000
       001136L 00110110 10000011
1103.  001137L 11000100 10100000        MWAIT  ,MEMPF1
       001140L 11010111 00001001
1104.  001141L 00110001 00110110        LDRP   TEMP1,MDR
       001142L 01101111 11110001
1105.  001143L 00110001 11000110        DLDX   HL2MR,,SMR               GET DATA AND UPDATE HL
       001144L 00110001 11100101
       001145L 00110111 01000111
1106.  001146L 00110001 10010000        DADDP  URO+UR,MARI
       001147L 00010110 01110010
       001150L 00110111 10000110
       001151L 00110001 10110000
       001152L 00110110 10000101
1107,  001153L 01110001 10110001        LDTR   TEMP1,CC                 GET DE DATA
1108.  001154L 11000100 10010011        MWAIT  ,MEMPF1
       001155L 11010111 00001001
1109.  001156L 00110100 00110110        DOPP   LUF,SB,MDR               SUBTRACT HL DATA AND SET COMPARE FLAGS
       001157L 00110111 00000110
1110,  001160L 11000011 10110111        BRA    RIN256,TZ                CONTINUE IF A MATCH (CARRY ZERO TOO!)
1111,  001161L 11001111 01110010        BRA    RIND                     END IF DATA DIFFERS
1112.
```

```
                        . DECIMAL FIELD OPERATIONS
1115,
1116,  001162L          .
1117,                   DFOP
1118,                   .
1119,                   .       (111 041)  DFAC             DECIMAL FIELD ADD
1120,                   . 4.25 + C * 7.45
1121,                   .       (IOD 041)  DFAC             DFAC UNSPECIFIED
1122,                   .
1123,                   .       (062 041)  DFSB             DECIMAL FIELD SUBTRACT
1124,                   . 4.25 + C * 7.85
1125,                   .       (IEV 041)  DFSB             DFSB UNSPECIFIED
1126,                   .
1127,                   .       TEMP1 <= (HL) .AND. 017     SELECT 4 BITS ONLY
1128,                   .       HL <= HL -1
1129,                   .       MAR <= DE
1130,                   .       DE <= DE - 1
1131,                   .       C     <= UCFLG
1132,                   .       T <= (MAR) .AND. 017
1133,                   .       IF IMP ODD THEN T <= TEMP1 <= T + TEMP1    (DFAC)
1134,                   .              LUF <= T + (-10)     (SET CARRY ON 10 OR GREATER)
1135,                   .                                     & CORRECT IF >= 10
1136,                   .              IF NOT C     THEN T <= TEMP1  <10, USE ORIGINAL VALUE
1137,                   .              (MAR) <= T .IOR. URB
1138,                   .              RIN16
1139,                   .         ELSE LUF <= T <= T - TEMP1              (DFSB)
1140,                   .                                  (NEGATIVE IF UNDEFLOW)
1141,                   .              IF C     THEN T <= T + 9   (+ C     FOR + 10, TO CORRECT)
1142,                   .              (MAR) <= T .IOR. URB
1143,                   .              RIN16
1144.  001162L 01010001 00001111          LDRI   TEMP2,017           SELECT 4 BITS, NOT ALL OF THEM
       001163L 01101111 11110010
1145,  001164L 01010001 10000000          BAL    ,DFADD              (ASSUME THIS)
1146,  001165L 11001011 01100110          BRA    BFOP,T0,IO          WAS ODD THEREFORE WAS DECIMAL ADD
1147,  001166L 01010001 10000111          BAL    ,DFSUB
1148,  001167L 11001111 01100110          BRA    BFOP                WAS EVEN SO IT WAS DECIMAL SUBTRACT
1149,                   .
1150,  001170L          DFSUB
1151.  001170L 01110100 11110001          DOPR   LUF,SB,TEMP1        SET FLAGS ON SUBTRACT
       001171L 00110111 00000110
1152,  001172L 11000000 10000011          BRA    DFSEND,FC           NO CARRY, NO PROBLEMS
1153,  001173L 01010010 00001001          DOTI   ,AC,10-1            CARRY, CORRECT THE VALUE (ADDS 10!)
1154.  001174L 00110011 11010001  DFSEND  DOPP   MDW,OR,URI+URB      SET THE ZONE BITS
       001175L 00110111 00100001
1155,  001176L 11001111 01111000          BRA    RIN16
1156,                   .
1157,  001177L          DFADD
1158.  001177L 01110010 11110001          DORR   TEMP1,AC,TEMP1      ADD TOGETHER (CARRY FALSE ON RESULT)
       001200L 01101111 11110001
1159.  001201L 01010010 11110110          DOPI   LUF,AC,-10          SET FLAGS & CORRECT IF OVERFLOW
       001202L 00110111 00000110
1160,  001203L 11000001 01111010          BRA    DFAEND,TC           IF OVERFLOWED, CORRECTED NOW
1161.  001204L 01110001 11110001          LDTR   TEMP1               NO CARRY, GET CORRECT VALUE
```

```
1162.  001205L 00110011 11010001    DFAEND   DOPP   MDW,OR,URI+URB          SET THE ZONE BITS
       001206L 00110111 00100001
1163.                               .        BRA    RIN16
1164.                               .
```

```
                                . REPEATED INSTRUCTIONS, REPEAT CONTROL SECTION
1167,                           *
1168,  001207L                  RIN16:
1169,                           . RPT - 1.70, SRV - 1.45, END - 3.95
1170,                           .
1171,                           .       FIELD REPEATED INSTRUCTIONS ARE THOSE THAT COUNT C THROUGH 16 MAX STEPS
1172,                           .
1173,                           .        URC <- URC - 1
1174,                           .        IF (T .AND. 017) NOT ZERO THEN RINST
1175,                           .          ELSE RIND
1176,                           .
1177,  001207L 00010111 10110010         CCLR
1178,  001210L 00110001 11010010         DOPIP  URO+URC,SB,1,URI+URC   COUNT DOWN
       001211L 01010100 00000001
       001212L 00110111 10000010
1179,  001213L 01000101 00001111         TSTIT  ,017                   AT THE END (BASE 16)?
1180,  001214L 11000010 10110001         BRA    RINST,FZ               NOT YET
1181,                           *
1182,  001215L                  RIND:
1183,                           . 3.25 (NON-IMP)
1184,                           .
1185,                           .        REPEATED INSTRUCTION END.
1186,                           .        CLEAR THE REPEATED FLAG
1187,                           .
1188,                           .        PSW <- PSW .AND. -1 - SWRPT
1189,                           .        FETCHW                        (BECAUSE SOME END IN MDW WITH NO MWAIT)
1190,                           .
1191,  001215L 00110001 11011100         DOPIP  PSWO,ND,-1-SWRPT,PSWI
       001216L 01010101 11011111
       001217L 00110111 10001100
1192,  >001220L 01011001 11111111         BRAX   FETCHW
       >001221L 11001111 11111111
1193,                           .
```

```
1196,                                   *
1197,  001222L                          BFSB:
1198,                                   .       (    031)  BFSB              BINARY FIELD SUBTRACT
1199,                                   . 2.25 + C * 6.90
1200,                                   .
1201,                                   .       (111 031)                    INTEGER DIVIDE
1202,                                   .       (062 031)                    INTEGER DIVIDE
1203,                                   .
1204,                                   .       T <= BFSUB
1205,                                   .       BFOP
1206,                                   .
1207,  001222L 01010001 11111111                LDRI   TEMP2,0377            SELECT ALL THE BITS
       001223L 01101111 11110010
1208,  001224L 01010001 01000011                BRC    BFOP,,BFSUB
       001225L 11001111 01100110
1209,                                   *
1210,  001226L                          BFAC:
1211,                                   .       (    011)  BFAC                  BINARY FIELD ADD
1212,                                   . 2.25 + C * 6.70
1213,                                   .
1214,                                   .       (111 011)                        INTEGER MULTIPLY
1215,                                   .       (IMP 011)                        2'S COMPLEMENT
1216,                                   .
1217,                                   .       T <= BFADD
1218,                                   .       BFOP
1219,                                   .
1220,  001226L 01010001 11111111                LDRI   TEMP2,0377            SELECT ALL THE BITS
       001227L 01101111 11110010
1221,  001230L 01010001 01000111                BAL    ,BFADD
1222,                                   .
1223,  001231L                          BFOP
1224,                                   . 4.1
1225,                                   .       BINARY FIELD OPERATIONS
1226,                                   .
1227,                                   .       LINK <= T
1228,                                   .       TEMP1 <= (HL)
1229,                                   .       HL <= HL - 1
1230,                                   .       MAR <= (DE)
1231,                                   .       DE <= DE - 1
1232,                                   .       CARRY <= UCFLG
1233,                                   .       T <= (MAR)
1234,                                   .       IF LINK = BFADD THEN LUF <= (MAR) <= T + TEMP1
1235,                                   .                            RIN16
1236,                                   .         ELSE LUF <= (MAR) <= T - TEMP1
1237,                                   .                            RIN16
1238,                                   .
1239,  001231L 01101111 10110000                BAS    LINK,CC
1240,  001232L 00110001 11000110                DLDX   HL2MR,,SMR
       001233L 00110001 11100101
       001234L 00110111 01000111
1241,  001235L 00110001 10010000                DDECP  URO+UR,MARI           GET (HL) DATA AND UPDATE HL
       001236L 01010100 00000001
```

```
                001237L 00110111 10000110
                001240L 00110001 10110000
                001241L 01010100 00000000
                001242L 00110111 10000101
1242.           001243L 01110001 11110010              LDTR    TEMP2
1243.           001244L 11000100 01011011              MWAIT   ,MEMPF1
                001245L 11010111 00001001
1244.           001246L 00110101 00110110              DORP    TEMP1,ND,MDR,,CC        PUT IT IN TEMP1
                001247L 01101111 10110001
1245.           001250L 00110001 11000100              DLDX    DE2MR,,SMR
                001251L 00110001 11100011
                001252L 00110111 01000111
1246.           001253L 00110001 10010000              DDECP   URO+UDE,MARI            GET (DE) DATA AND UPDATE DE
                001254L 01010100 00000001
                001255L 00110111 10000100
                001256L 00110001 10110000
                001257L 01010100 00000000
                001260L 00110111 10000011
1247.
1248.           001261L 00110001 00110101      .       DOTPP   ,AC,UCFLG,UCFLG         CARRY IGNORED, BITS 2&6 NOT BOTH HIGH
                001262L 00110010 00110101                                             SET THE CARRY
1249.           001263L 01110001 11110010              LDTR    TEMP2
1250.           001264L 11000100 01001011              MWAIT   ,MEMPF1
                001265L 11010111 00001001
1251.           001266L 00110101 00110110              DOTP    ,ND,MDR
1252.           001267L 11101111 00000000              BRR     LINK                    SELECT ADD & SUBTRACT, BINARY & DECIMAL
1253.
1254.           001270L 01110010 11110001      BFADD   DOPR    MDW,AC,TEMP1            (DE) <- (DE) + (HL)
                001271L 00110111 00100001
1255.           001272L 00110111 00000110              LDPT    LUF
1256.           001273L 11001111 01111000              BRA     RIN16
1257.
1258.           001274L 01110100 11110001      BFSUB   DOPR    MDW,SB,TEMP1            (DE) <- (DE) - (HL)
                001275L 00110111 00100001
1259.           001276L 00110111 00000110              LDPT    LUF
1260.           001277L 11001111 01111000              BRA     RIN16
1261.
```

```
                                . BINARY FIELD SHIFTS LEFT AND RIGHT

1264,                           *
1265,  001300L                  BFS:
1266,                           .       (    075)  BFSL              BINARY FIELD SHIFT LEFT
1267,                           . 2.25 + C * 4.55
1268,                           .
1269,                           .       (111 075)  BFSR              BINARY FIELD SHIFT RIGHT
1270,                           . 4.25 + C * 4.55
1271,                           .       (IMP 075)  BFSR              BFSR UNSPECIFIED
1272,                           .
1273,                           .       NOTE: ONLY CHANGES THE CARRY FLAG
1274,                           .
1275,                           .       MAR <= HL
1276,                           .       IF IMP ZERO THEN HL <= HL - 1     (BFSL)
1277,                           .              C     <= UCFLG
1278,                           .              LUCF <= (MAR) <= (MAR) + (MAR)     ARITHMETIC SHIFT LEFT
1279,                           .              RIN16
1280,                           .         ELSE HL <= HL + 1                (BFSR)
1281,                           .              L    <= UCFLG
1282,                           .              (MAR) <= (MAR) SHFT 1              ARITHMETIC SHIFT RIGHT
1283,                           .              LUF <= L
1284,                           .              RIN16
1285,                           .
1286.  001300L 00110001 11000110        DLDX   HL2MR,,SMR           GET DATA TO SHIFT & SELECT R/L ROUTINE
       001301L 00110001 11100101
       001302L 00110111 01000111
1287.  001303L 11001001 00101001        BRA    BFSL,T0,IZ
1288,                           . BFSR
1289.  001304L 00110001 10010000        DADDP  UR0+UR,MARI          UPDATE HL
       001305L 00010110 01110010
       001306L 00110111 10000110
       001307L 00110001 10110000
       001310L 00110110 10000101
1290.  001311L 00110001 00110101        LDTP   UCFLG
1291,  001312L 00010111 10100010        SHIFT  SL                   PUT THE CARRY IN THE LSB (BIT 0)
1292,  001313L 00010111 10110010        CCLR                        AND INTO THE LINK
1293.  001314L 11000100 00110011        MWAIT  ,MEMPF1
       001315L 11010111 00001001
1294.  001316L 00110001 00110110        LDTP   MDR                  GET THE DATA
1295,  001317L 00010111 10010010        SHIFT  SR                   DO EXTENDED SHIFT
1296.  001320L 00110111 00100001        LDPT   MDW                  OUTPUT SHIFTED DATA
1297,  001321L 00010111 10010010        SHIFT  SR                   PUT LINK IN THE MSB
1298,  001322L 01101111 11110001        LDRT   TEMP1
1299.  001323L 01110010 00110001        DOPR   LUCF,AC,TEMP1,,C0    SET THE CARRY FROM THE MSB (LINK)
       001324L 00110111 00000111
1300,  001325L 11001111 01111000        BRA    RIN16
1301,                           .
1302,  001326L 00010111 10110010  BFSL  CCLR                        UPDATE HL
1303.  001327L 00110001 10010000        DDECP  UR0+UR,MARI
       001330L 01010100 00000001
       001331L 00110111 10000110
       001332L 00110001 10110000
       001333L 01010100 00000000
```

```
                001334L 00110111 10000101
1304*
1305*           001335L 00110001 00110101       .       DOTPP   ,AC,UCFLG,UCFLG          CARRY IN STOPPED IN BIT 2 OR 6 OF UCFLG
                001336L 00110010 00110101                                               SET CARRY FROM USER CARRY
1306*           001337L 11000100 00100000               MWAIT   ,MEMPF1
                001340L 11010111 00001001
1307*           001341L 00110001 00110110               DOPPP   MDW,AC,MDR,MDR          SHIFT LEFT!
                001342L 00110010 00110110
                001343L 00110111 00100001
1308*           001344L 00110111 00000111               LDPT    LUCF                   SET CARRY ON RESULT
1309*           001345L 11001111 01111000               BRA     RIN16
```

```
1312,                                   *
1313,  001346L                          SLC:
1314,                                   . 3,00   (    002)  SLC              SHIFT LEFT CIRCULAR
1315,                                   . 5.00   (IMP 002)  SLCR             USING OTHER THAN A-REG
1316,                                   .
1317,                                   .       LUCF <- T <- IMP + IMP
1318,                                   .       IMP <- T + CARRY
1319,                                   .
1320.  001346L 00110001 11011111                LDRP   TEMP1,IMPI
       001347L 01101111 11110001
1321.  001350L 01110010 00110001                DOPR   LUCF,AC,TEMP1,,C0      ADD TOGETHER AND SET USER CARRY
       001351L 00110111 00000111
1322,  001352L 00110110 10001111                DOP    IMPO,IT                PUT CARRY IN THE LSB
1323. >001353L 01011001 11111111                BRAX   FETCHI                 DON'T USE IMPFO, WILL GET BAD CARRY
      >001354L 11001111 11111111
1324,                                   *
1325,  001355L                          SRC:
1326,                                   . 3,20   (    012)  SRC              SHIFT RIGHT CIRCULAR
1327,                                   . 5,20   (IMP 012)  SRCR             USING OTHER THAN A-REG
1328,                                   .
1329,                                   .       IMP <- T <- IMP ROT 1
1330,                                   .       LUCF <- T + T
1331,                                   .
1332,  001355L 00110001 11011111                LDTP   IMPI
1333,  001356L 00010111 10110010                CCLR                          SET THE LINK BIT ON LSB
1334,  001357L 00010111 10010010                SHIFT  SR                     RIGHT CIRCULAR
1335,  001360L 00110111 10001111                LDPT   IMPO                   STORE RESULT
1336,  001361L 01101111 11110001                LDRT   TEMP1
1337.  001362L 01110010 00110001                DOPR   LUCF,AC,TEMP1,,C0      SET CARRY ON THE MSBIT
       001363L 00110111 00000111
1338. >001364L 01011001 11111111                BRAX   FETCHI
      >001365L 11001111 11111111
1339,                                   *
1340,  001366L                          SRE:
1341,                                   . 3,55   (    032)  SRE              SHIFT RIGHT EXTENDED
1342,                                   . 5,55   (IMP 032)  SRER             USING OTHER THAN THE A-REG
1343,                                   .
1344,                                   .       LINK <- UCFLG (CARRY)
1345,                                   .       LINK, IMP <- (LINK, IMP) ROT 1
1346,                                   .       T <- (LINK, IMP) ROT 1        MOVES LINK INTO MSBIT
1347,                                   .       LUCF <- T + T
1348,                                   .
1349,  001366L 00110001 00110101                LDTP   UCFLG                  GET CARRY
1350,  001367L 00010111 10100010                SHIFT  SL                     IN LSBIT
1351,  001370L 00010111 10110010                CCLR                          AND INTO LINK
1352,  001371L 00110001 11011111                LDTP   IMPI
1353,  001372L 00010111 10010010                SHIFT  SR                     DO RIGHT EXTEND
1354,  001373L 00110111 10001111                LDPT   IMPO
1355,  001374L 00010111 10010010                SHIFT  SR                     PUT LINK IN MSBIT
1356,  001375L 01101111 11110001                LDRT   TEMP1
1357.  001376L 01110010 00110001                DOPR   LUCF,AC,TEMP1,,C0      SET CARRY ON LINK
       001377L 00110111 00000111
```

. SHIFT OPERATIONS

```
1358.  >001400L 01011001 11111111          BRAX   FETCHI
       >001401L 11001111 11111111
1359.                                  *
1360.                                  .       TABPAGE PROOL
1361.                                  *
1362.   001402L                        CCS:
1363.                                  . 2,75   (    042)  CCS              CONDITION CODE SAVE
1364.                                  . 4,75   (IMP 042)  CCS              USING OTHER THAN A-REG
1365.                                  .
1366.                                  .       IMP <- UCFLG .AND. 0303     ONLY SEE THE IMPORTANT BITS
1367.                                  .
1368.   001402L 00110001 00110101              DOPIP  IMPO,ND,0303,UCFLG
        001403L 01010101 11000011
        001404L 00110111 10001111
1369.  >001405L 01011001 11111111          BRAX   FETCHI
       >001406L 11001111 11111111
1370.                                  *
```

```
1373;                                   *
1374;  001407L                          PLR:
1375;                                   . 5.85   (     1R4)  PLR,IDX          LOAD REGISTER FROM PAGED LOCATION
1376;                                   .        (     105)  PLR,IDX          A-REG LOAD (104 IS NOT A PLR)
1377;                                   .        (     114)  PLR,IDX          LOAD B-REG FROM PAGED LOCATION
1378;                                   .        (     124)  PLR,IDX          LOAD C-REG FROM PAGED LOCATION
1379;                                   .        (     134)  PLR,IDX          LOAD D-REG FROM PAGED LOCATION
1380;                                   .        (     144)  PLR,IDX          LOAD E-REG FROM PAGED LOCATION
1381;                                   .        (     154)  PLR,IDX          LOAD H-REG FROM PAGED LOCATION
1382;                                   .        (     164)  PLR,IDX          LOAD L-REG FROM PAGED LOCATION
1383;                                   .
1384;                                   . 9.25   (111 124)  DPL,IDX          BC REG PAIR LOADED FROM PAGED INDEX
1385;                                   .        (062 114)  DPLR,IDX          BC REG PAIR REVERSED
1386;                                   .        (113 144)  DPL,IDX          DE REG PAIR LOADED FROM PAGED INDEX
1387;                                   .        (174 134)  DPLR,IDX         DE REG PAIR REVERSED
1388;                                   .        (115 164)  DPL,IDX          HL REG PAIR LOADED FROM PAGED INDEX
1389;                                   .        (176 154)  DPLR,IDX         HL REG PAIR REVERSED
1390;                                   .
1391;                                   .        (117 105)  PLA,IDX          XA REG PAIR UNSPECIFIED
1392;                                   .        (IMP 1R4)  PLR,IDX          UNSPECIFIED MIXED UP REG PAIR
1393;                                   .                                      USE 105 NOT 104 FOR 1R4 WITH R=0
1394;                                   .
1395;                                   .        PC <- MAR <- MAR + 1
1396;                                   .        MAR <- X, (MAR)
1397;                                   .        I35 <- (MAR)
1398;                                   .        IF IMP NOT ZERO THEN IMP <- (MAR+1)
1399;                                   .
1400;  001407L 00110111 00001100                STB    IMAR,SMR               GET THE INDEX
       001410L 00110111 01000111
1401;  001411L 00110001 10001001                DLDX   MR2PC                  AND SAVE AWAY NEW PC
       001412L 00110001 10101000
1402;  001413L 11010100 11110100                MWAIT  ,MEMPF1
       001414L 11010111 00001001
1403;  001415L 00110001 00110110                LDPP   MAROL,MDR              LSB ADDRESS
       001416L 00110111 11000000
1404;  001417L 00110001 11100111                LDX    XX2MRH+URX,SMR         MSB ADDRESS AND GET DATA
       001420L 00110111 01000111
1405;  001421L 11001110 11111111                MWAIT  NOOP,MEMPF1
       001422L 11010100 11101101
       001423L 11010111 00001001
1406;  001424L 00110001 00110110                LDPP   I350,MDR               GOT THE DATA
       001425L 00110111 10001101
1407;  001426L 11011001 11100001                BRA    FETPL,T0,IZ            IF IMP NOT ZERO GET ANOTHER BYTE
1408;  001427L 00110111 00001100                STB    IMAR,SMR
       001430L 00110111 01000111
1409;  001431L 11001110 11111111                MWAIT  NOOP,MEMPF1
       001432L 11010100 11100101
       001433L 11010111 00001001
1410;  001434L 00110001 00110110                LDPP   IMP0,MDR               GET THE SECOND DATA BYTE
       001435L 00110111 10001111
1411;  >001436L 01011001 11111111     FETPL     BRAX   FETCH
       >001437L 11001111 11111111
```

```
                        . SINGLE AND DOUBLE PAGED LOAD AND STORE
1412.                   *
1413.  001440L           PSR:
1414.                   . 5.20    (     1R6)  PSR,IDX           STORE REG INTO PAGED LOCATION
1415.                   .         (     107)  PSR,IDX           A REG SAVE (106 IS NOT A PSR)
1416.                   .         (     116)  PSR,IDX           STORE REG-B INTO PAGED LOCATION
1417.                   .         (     126)  PSR,IDX           STORE REG-C INTO PAGED LOCATION
1418.                   .         (     136)  PSR,IDX           STORE REG-D INTO PAGED LOCATION
1419.                   .         (     146)  PSR,IDX           STORE REG-E INTO PAGED LOCATION
1420.                   .         (     156)  PSR,IDX           STORE REG-H INTO PAGED LOCATION
1421.                   .         (     166)  PSR,IDX           STORE REG-L INTO PAGED LOCATION
1422.                   .
1423.                   . 8.55    (111 116)  DPSR,IDX           BC REGISTER PAIR SAVED
1424.                   .         (062 116)  DPSR,IDX           BC REGISTER PAIR REVERSE SAVED
1425.                   .         (113 146)  DPS,IDX            DE REGISTER PAIR SAVED
1426.                   .         (174 136)  DPSR,IDX           DE REGISTER PAIR REVERSE SAVED
1427.                   .         (115 166)  DPS,IDX            HL REGISTER PAIR SAVED
1428.                   .         (176 156)  DPSR,IDX           HL REGISTER PAIR REVERSE SAVED
1429.                   .
1430.                   .         (117 107)  PS ,IDX            XA REGISTER PAIR UNSPECIFIED
1431.                   .         (IMP IR6)  PS ,IDX            UNSPECIFIED MIXED REGISTERS SAVED
1432.                   .                                        (USE 107 NOT 1R6 FOR R=0)
1433.                   .
1434.                   .         PC <= MAR <= MAR + 1
1435.                   .         MAR <= X, (MAR)
1436.                   .         (MAR) <= I35
1437.                   .         IF IMP NOT ZERO THEN (MAR+1) <= IMP
1438.                   .
1439.  001440L 00110111 00001100          STB    IMAR,SMR              GET THE INDEX
       001441L 00110111 01000111
1440.  001442L 00110001 10001001          DLDX   MR2PC                 UPDATE THE PC SO IT CORRECT
       001443L 00110001 10101000
1441.  001444L 11010100 11011011          MWAIT  ,MEMPF1
       001445L 11010111 00001001
1442.  001446L 00110001 00110110          LDPP   MAROL,MDR             LSB ADDRESS
       001447L 00110111 11000000
1443.  001450L 00110001 11100111          LDX    XX2MRH+URX            AND MSB ADDRESS FROM X-REG
1444.  001451L 00110001 11011101          LDPP   MDW,I35I              SAVE THE REGISTER IN MEMORY
       001452L 00110111 00100001
1445.  001453L 11011001 11001110          BRA    FETPS,T0,IZ           DONE IF IMP ZERO
1446.  001454L 00110001 11011111          LDTP   IMPI                  GET OTHER REG TO SAVE
1447.  001455L 11010100 11010010          MWAIT  ,MEMPF1
       001456L 11010111 00001001
1448.  001457L 00110111 00001100          STB    IMAR                  POINT TO ITS SAVE AREA
1449.  001460L 00110111 00100001          LDPT   MDW                   SAVE IT
1450. >001461L 01011001 11111111  FETPS   BRAX   FETCHW
      >001462L 11001111 11111111
1451.                   .
```

```
1454,                                   *
1455,  001463L                          INCPA:
1456,                                   . 4,20   (    017)  INCP    HL,A        INCREMENT REG PAIR USING A AS DISP.
1457,                                   . 6,05   (062 017)  INCP                BC,A
1458,                                   .        (174 017)  INCP                DE,A
1459,                                   .        (022 017)  INCP                XA,A
1460,                                   . -.65 IF NO CARRY
1461,                                   .
1462,                                   .        (176 017)  INCP                HL,A UNSPEC'D
1463,                                   .        (IOD 017)  INCP                UNSPECIFIED MISMATCHED REG PAIRS
1464,                                   .
1465,                                   .        IF IMP ZERO THEN IMP <= URL
1466,                                   .        T <= URA
1467,                                   .        INCPIT
1468,                                   .
1469,  001463L 11011000 11001001                BRA    INCPAS,F0,IZ
1470,  001464L 01010001 00000110                LDPI   LIMP,URL                USE HL IF NO IMP SPECIFICATION
       001465L 00110111 00000001
1471,  001466L 00110001 11010000        INCPAS  LDTP   URI+URA
1472,  001467L 11011111 11000000                BRA    INCPIT                  T LOADED WITH VALUE TO ADD, GO DO IT
1473,                                   *
1474,  001470L                          INCP:
1475,                                   . 4,15   (    015)  INCP                HL  INCREMENTED BY 1
1476,                                   . 6,15   (111 015)  INCP                XA,2
1477,                                   . 6,00   (062 015)  INCP                BC  INCREMENTED BY 1
1478,                                   . 6,15   (113 015)  INCP                BC,2
1479,                                   . 6,00   (174 015)  INCP                DE  INCREMENTED BY 1
1480,                                   . 6,15   (115 015)  INCP                DE,2
1481,                                   . 6,15   (117 015)  INCP                HL,2
1482,                                   . 6,00   (022 015)  INCP                XA  INCREMENTED BY 1
1483,                                   . -.65 IF NO CARRY
1484,                                   .
1485,                                   .        (176 015)  INCP                HL  BY 1 UNSPECIFIED
1486,                                   .
1487,                                   .        IF IMP ZERO THEN IMP <= URL
1488,                                   .        T <= 1
1489,                                   .        IF IMP ODD THEN T <= 2
1490,                                   .                        IMP <= IMP - 1
1491,                                   .        LUF <= IMP, IMP-1 <= IMP, IMP-1 + T
1492,                                   .
1493,  001470L 11011000 11000100                BRA    INCPS,F0,IZ
1494,  001471L 01010001 00000110                LDPI   LIMP,URL                DEFAULT TO HL PAIR
       001472L 00110111 00000001
1495,  001473L 01010001 00000001        INCPS   LDTI   1                       ASSUME ITS BY 1 (ODD)
1496,  001474L 11011010 11000000                BRA    INCPIT,F0,IO
1497,  001475L 01010001 00000010                LDTI   2                       NO, BY 2 CAUSE WAS EVEN
1498,  001476L 00110111 00001001                STB    DIMP                    CORRECT IT TO BE ODD
1499,  001477L 01101111 11110001        INCPIT  LDRT   TEMP1
1500,  001500L 00110001 11011111                DOPRP  IMPF0,AC,TEMP1,IMPI,C0 INC THE LSB (T=SAVE NECESSARY)
       001501L 01110010 00110001
       001502L 00110111 01101111
1501, >001503L 01011001 11111111                BRAX   FETCHI,FC               IF NO CARRY FORGET THE MSB
```

. REGISTER PAIR INCREMENT AND DECREMENT

```
      >001504L 11000000 11111111
1502*  001505L 00110111 00001001          STB     DIMP
1503*  001506L 00110001 11011111          DOPIP   IMPFO,AC,0,IMPI         INC THE MSB (COULD USE IT)
       001507L 01010010 00000000
       001510L 00110111 01101111
1504*                                                                      CAN'T USE 'IT IMPFO' BECAUSE DELAYED
1505*                                      .                               ALU CARRY THROUGH PROCESSOR CARRY
1506*  >001511L 11001111 11111111          .       BRA     FETCHI
```

```
1507,                                   *
1508,  001512L                          DECPA:
1509,                                   . 4,20   (    037)  DECP     HL,A      DECREMENT REG, PAIR USING A AS DISP,
1510,                                   .
1511,                                   . 6,05   (062 037)  DECP               BC,A
1512,                                   .        (174 037)  DECP               DE,A
1513,                                   .        (022 037)  DECP               XA,A
1514,                                   . -,65 IF NO CARRY
1515,                                   .
1516,                                   .        (176 037)  DECP               HL,A UNSPEC'D
1517,                                   .        (IOD 037)  DECP               UNSPECIFIED MIXED REGS
1518,                                   .
1519,                                   .        IF IMP ZERO THEN IMP <= URL
1520,                                   .        T <= URA
1521,                                   .        DECPIT
1522,                                   .
1523,  001512L 11011000 10110010                BRA    DECPAS,F0,IZ
1524,  001513L 01010001 00000110                LDPI   LIMP,URL                  SELECT HL AS DEFAULT REG PAIR
       001514L 00110111 00000001
1525,  001515L 00110001 11010000        DECPAS LDTP   URI+URA
1526,  001516L 11011111 10101001                BRA    DECPIT                    GO DECR, USING A-REG AS DISP,
1527,                                   *
1528,  001517L                          DECP:
1529,                                   . 4,15   (    035)  DECP               HL   DECREMENT BY 1
1530,                                   . 6,15   (111 035)  DECP               XA,2
1531,                                   . 6,00   (062 035)  DECP               BC   DECREMENT BY 1
1532,                                   . 6,15   (113 035)  DECP               BC,2
1533,                                   . 6,00   (174 035)  DECP               DE   DECREMENT BY 1
1534,                                   . 6,15   (115 035)  DECP               DE,2
1535,                                   . 6,15   (117 035)  DECP               HL,2
1536,                                   . 6,00   (022 035)  DECP               XA   DECREMENT BY 1
1537,                                   . -,65 IF NO CARRY
1538,                                   .
1539,                                   .        (176 035)  DECP               HL   DECREMENT BY 1 UNSPEC'D
1540,                                   .
1541,                                   .        IF IMP ZERO THEN IMP <= URL
1542,                                   .        T <= 1
1543,                                   .        IF IMP ODD THEN T <= 2
1544,                                   .                        IMP <= IMP - 1
1545,                                   .        IMP, IMP-1 <= IMP, IMP-1 - T
1546,                                   .
1547,  001517L 11011000 10101101                BRA    DECPS,F0,IZ
1548,  001520L 01010001 00000110                LDPI   LIMP,URL
       001521L 00110111 00000001
1549,  001522L 01010001 00000001        DECPS  LDTI   1                          EXACTLY LIKE ABOVE BUT USING SB'S NOT AC
1550,  001523L 11011010 10101001                BRA    DECPIT,F0,IO
1551,  001524L 01010001 00000010                LDTI   2
1552,  001525L 00110111 00001001                STB    DIMP
1553,  001526L 01101111 11110001        DECPIT LDRT   TEMP1
1554,  001527L 00110001 11011111                DOPRP  IMPFO,SB,TEMP1,IMPI,C0
       001530L 01110100 00110001
       001531L 00110111 01101111
```

. REGISTER PAIR INCREMENT AND DECREMENT

```
1555.  >001532L 01011001 11111111          BRAX   FETCHI,FC              IF NO CARRY FORGET THE MSB
       >001533L 11000000 11111111
1556.   001534L 00110111 00001001          STB    DIMP
1557.   001535L 00110001 11011111          DOPIP  IMPFO,SB,0,IMPI
        001536L 01010100 00000000
        001537L 00110111 01101111
1558.  >001540L 11001111 11111111          BRA    FETCHI
1559.
```

```
1562,                                   *
1563,  001541L                          DECX:
1564,                                   . 11,40  (    025)  DECI,LSP,IDX       DECREMENT INDEX BY DISPLACEMENT
1565,                                   . 14,90  (111 025)  DECI,LSP,MSB,IDX      WITH DOUBLE BYTE DISPLACEMENT
1566,                                   .
1567,                                   . 11,90  (062 025)  LFID,LSP,IDX       BC LOADED WITH DECREMENTED INDEX
1568,                                   . 13,40  (113 025)  LFID,LSP,MSP,IDX      WITH DOUBLE BYTE DECREMENT
1569,                                   . 11,90  (174 025)  LFID,LSP,IDX       DE LOADED WITH DECREMENTED INDEX
1570,                                   . 13,40  (115 025)  LFID,LSP,MSP,IDX      WITH DOUBLE BYTE DECREMENT
1571,                                   . 11,90  (176 025)  LFID,LSP,IDX       HL LOADED WITH DECREMENTED INDEX
1572,                                   . 13,40  (117 025)  LFID,LSP,MSP,IDX      WITH DOUBLE BYTE DECREMENT
1573,                                   .
1574,                                   .        (022 025)  LFID,LSP,IDX       XA LOADED UNSPECIFIED
1575,                                   .
1576,                                   .        T <= DECIT
1577,                                   .        INDX                          (AND SELECT INC. OR DEC. CODE)
1578,                                   .
1579,  001541L 01010001 01101011        .        BRC    INDX,,DECIT            GO TO INDEX SPECIFYING DECIT SET
       001542L 11011111 10011011
1580,                                   *
1581,  001543L                          INCX:
1582,                                   . 11,10  (    005)  INCI,LSP,IDX       INCREMENT INDEX BY DISPLACEMENT
1583,                                   . 14,60  (111 005)  INCI,LSP,MSP,IDX      WITH DOUBLE BYTE DISPLACEMENT
1584,                                   .
1585,                                   . 11,80  (062 005)  LFII,LSP,IDX       BC LOADED WITH INCREMENTED INDEX
1586,                                   . 13,30  (113 005)  LFII,LSP,MSP,IDX      WITH DOUBLE BYTE INCREMENT
1587,                                   . 11,80  (174 005)  LFII,LSP,IDX       DE LOADED WITH INCREMENTED INDEX
1588,                                   . 13,30  (115 005)  LFII,LSP,MSP,IDX      WITH DOUBLE BYTE INCREMENT
1589,                                   . 11,80  (176 005)  LFII,LSP,IDX       HL LOADED WITH INCREMENTED INDEX
1590,                                   . 13,30  (117 005)  LFII,LSP,MSP,IDX      WITH DOUBLE BYTE INCREMENT
1591,                                   .
1592,                                   .        (022 005)  LFII,LSP,IDX       XA LOADED UNSPECIFIED
1593,                                   .
1594,                                   .        T <= INCIT
1595,                                   .        INDX
1596,                                   .
1597,  001543L 01010001 10000000        .        BAL    ,INCIT
```

```
                        . INDEXED LOCATIONS,  INCREMENT AND DECREMENT IN PLACE OR TO REG PAIR

1598,                   +
1599,  001544L          INDX
1600,                   .
1601,                   . INDEXED OPERATIONS TAKES 2 BYTES ON PAGE X-REG ADDS DISPLACEMENT TO IT
1602,                   .     AND STORES THE RESULT IN A REG, PAIR OR BACK ON TOP OF ITSELF
1603,                   .
1604,                   .        MAR <- MAR + 1                 GET DISPL, & INDEX BYTES
1605,                   .        TEMPL <- (MAR)
1606,                   .        MAR <- MAR + 1
1607,                   .        TEMPH <- IF IMP ODD
1608,                   .                 THEN IMP <- IMP - 1    (MSB IN MEMORY)
1609,                   .                      (MAR)             (RESULT GIVEN TO TEMPH)
1610,                   .                      MAR <- MAR + 1
1611,                   .                 ELSE 0                 (NO MSB BYTE, USE ZERO)
1612,                   .        PC <- MAR                      PC SET TO INDEX BYTE
1613,                   .        MAR <- X, (MAR)
1614,                   .        IF LINK = DECIT                DO THE INCX OR DECX
1615,                   .          THEN T <- (MAR, MAR+1) - T
1616,                   .          ELSE T <- (MAR, MAR+1) + T
1617,                   .        IF IMP ZERO THEN (MAR, MAR+1) <- T
1618,                   .           ELSE (IMP, IMP-1) <- T
1619,                   .
1620.  001544L 00110111 00001100          STB    IMAR,SMR               GET DISPLACEMENT LSP
       001545L 00110111 01000111
1621.  001546L 01101111 11110000          BAS    LINK                   FREE REGISTER FOR INC, DEC SELECT
1622.  001547L 11010100 10011000          MWAIT  ,MEMPF1
       001550L 11010111 00001001
1623.  001551L 00110001 00110110          LDTP   MDR
1624.  001552L 00110111 00001100          STB    IMAR,SMR               GET MSP OR INDEX
       001553L 00110111 01000111
1625.  001554L 01101111 11110010          LDRT   TEMPL                  SAVE LSP (LSP DISPLACEMENT)
1626.  001555L 01010001 00000000          TCLR                          ASSUME MSP IS ZERO
1627.  001556L 11011010 10001010          BRA    INDGET,F0,IO
1628.  001557L 11010100 10010000          MWAIT  ,MEMPF1                NO, GET MSP OF DISPLACEMENT
       001560L 11010111 00001001
1629.  001561L 00110001 00110110          LDTP   MDR                    GOT MSB
1630.  001562L 00110111 00001100          STB    IMAR,SMR               POINT AND GET INDEX
       001563L 00110111 01000111
1631.  001564L 00110111 00001001          STB    DIMP                   CORRECT IMP REG
1632.  001565L 01101111 11110001  INDGET  LDRT   TEMPH                  SAVE MSP
1633.  001566L 00110001 10001001          DLDX   MR2PC                  PC SAVED AS CORRECT VALUE NOW
       001567L 00110001 10101000
1634.  001570L 11010100 10000111          MWAIT  ,MEMPF1                GOT INDEX
       001571L 11010111 00001001
1635.  001572L 00110001 00110110          LDPP   MAROL,MDR              INDEX TO THE MAR LSB
       001573L 00110111 11000000
1636.  001574L 00110001 11100111          LDX    XX2MRH+URX,SMR         AND MAR MSB FROM THE X-REG
       001575L 00110111 01000111
1637,  001576L 11101111 00000000          BRR    LINK                   SELECT DECIT OR INCIT
1638,                   .
1639.  001577L 11010100 10000000  INCIT   MWAIT  ,MEMPF1
       001600L 11010111 00001001
```

```
1640.  001601L 00110001 00110110          LDTP   MDR
1641.  001602L 00110111 00001100          STB    IMAR,SMR              START TO READ FOR MSB FAST
       001603L 00110111 01000111
1642.  001604L 01110010 00110010          DORR   TEMPL,AC,TEMPL,,C0    ADD DATA AND DISPLACEMENT LSB
       001605L 01101111 11110010
1643.  001606L 11010100 01111001          MWAIT  ,MEMPF1
       001607L 11010111 00001001
1644.  001610L 00110001 00110110          DOTRP  ,AC,TEMPH,MDR         ADD MSB PART, LEAVE RESULT IN T
       001611L 01110010 11110001
1645.  001612L 11011000 01011111          BRA    INDXL,F0,IZ           SELECT LOAD REG PAIR OR RELOAD MEMORY
1646.
1647.  001613L 00110111 00100001  INDXS   LDPT   MDW                  PUT MSB RESULT BACK IN MEMORY
1648.  001614L 00110111 00000110          LDPT   LUF                   SET CONDITION CODES
1649.  001615L 01110001 11110010          LDTR   TEMPL
1650.  001616L 11010100 01110001          MWAIT  ,MEMPF1               PUT LSB DATA BACK ALSO
       001617L 11010111 00001001
1651.  001620L 00110111 00001101          STB    DMAR                  IN CORRECT SPOT
1652.  001621L 00110111 00100001          LDPT   MDW
1653. >001622L 01011001 11111111          BRAX   FETCHW
      >001623L 11001111 11111111
1654.
1655.  001624L 11010100 01101011  DECIT   MWAIT  ,MEMPF1
       001625L 11010111 00001001
1656.  001626L 00110001 00110110          LDTP   MDR                   GET LSB DATA
1657.  001627L 00110111 00001100          STB    IMAR,SMR              START TO READ FOR MSB FAST
       001630L 00110111 01000111
1658.  001631L 01110100 00110010          DORR   TEMPL,SB,TEMPL,,C0    SUBTRACT DISPLACEMENT FROM THE DATA
       001632L 01101111 11110010
1659.  001633L 11010100 01100100          MWAIT  ,MEMPF1
       001634L 11010111 00001001
1660.  001635L 00110001 00110110          DOTRP  ,SB,TEMPH,MDR         SIMILARLY FOR THE MSB PART
       001636L 01110100 11110001
1661.  001637L 11011001 01110100          BRA    INDXS,T0,IZ           SELECT LOAD REG PAIR OR RELOAD MEMORY
1662.
1663.  001640L 00110111 00001001  INDXL   STB    DIMP                  CORRECT POINTER TO MSB
1664.  001641L 00110111 01101111          LDPT   IMPFO,IIMP            LOAD THE INDEX INTO A REGISTER PAIR
       001642L 00110111 00001000
1665.  001643L 01110001 11110010          LDPR   IMPO,TEMPL            THE LSB ALSO
       001644L 00110111 10001111
1666. >001645L 01011001 11111111          BRAX   FETCH
      >001646L 11001111 11111111
1667.
```

. DOUBLE LOADS AND DOUBLE STORES

```
1670,                                   *
1671,  001647L                          DS:
1672,                                   . 6,10   (     027)  DS                  DE,HL  DE WHERE HL POINTS
1673,                                   . 8,80   (111  027)  DS                  BC,HL  BC WHERE HL POINTS
1674,                                   . 8,10   (113  027)  DS                  BC,DE  BC WHERE DE POINTS
1675,                                   . 7,95   (174  027)  DS                  DE,BC  DE WHERE BC POINTS
1676,                                   . 7,95   (176  027)  DS                  HL,BC  HL WHERE BC POINTS
1677,                                   . 8,10   (117  027)  DS                  HL,DE  HL WHERE DE POINTS
1678,                                   .
1679,                                   .        (062  027)  DS                  BC,BC  UNSPECIFIED
1680,                                   .        (115  027)  DS                  DE,DE  UNSPECIFIED
1681,                                   .        (022  027)  DS                  XA,BC  UNSPECIFIED
1682,                                   .
1683,                                   .        T <= DSTORE
1684,                                   .        DLS
1685,                                   .
1686,  001647L 01010001 00110000                 BRC    DLS,,DSTORE              SELECT REG PAIRS & DO SAVE
       001650L 11011111 01010001
1687,                                   *
1688,  001651L                          DLHL:
1689,                                   . 6,50   (     057)  DL                  HL,HL
1690,                                   .
1691,                                   .        LINK <= DSTORE
1692,                                   .        T <= URL
1693,                                   .        DLSLMP
1694,                                   .
1695,  001651L 01010001 00111111                 BAL    LINK,DLOAD               LOAD REG PAIR
       001652L 01101111 11110000
1696,  001653L 01010001 00000110                 LDTI   URL                      HL
1697,  001654L 11011111 01000011                 BRA    DLSLMP                   FROM (HL)
1698,                                   *
1699,  001655L                          DL:
1700,                                   . 6,50   (     047)  DL                  DE,HL  LOAD DE FROM (HL)
1701,                                   . 9,20   (111  047)  DL                  BC,HL  LOAD BC FROM (HL)
1702,                                   . 8,35   (062  047)  DL                  BC,BC  LOAD BC FROM (BC)
1703,                                   . 8,50   (113  047)  DL                  BC,DE  LOAD BC FROM (DE)
1704,                                   . 8,35   (174  047)  DL                  DE,BC  LOAD DE FROM (BC)
1705,                                   . 8,50   (115  047)  DL                  DE,DE  LOAD DE FROM (DE)
1706,                                   . 8,35   (176  047)  DL                  HL,BC  LOAD HL FROM (BC)
1707,                                   . 8,50   (117  047)  DL                  HL,DE  LOAD HL FROM (DE)
1708,                                   .
1709,                                   .        (022  047)  DL                  XA,BC  UNSPECIFIED
1710,                                   .
1711,                                   .
1712,                                   .        T <= DSTORE
1713,                                   .        DLS
1714,                                   .
1715,  001655L 01010001 00111111                 BAL    ,DLOAD                   LOAD SELECT
```

```
                                . DOUBLE LOADS AND DOUBLE STORES
1716,                           *
1717,  001656L                  DLS
1718,                           .                                       DECODE ADDRESS & REG SPEC, OPERATORS
1719,                           .
1720,                           .       LINK <= T
1721,                           .       IF IMP ZERO
1722,                           .         THEN IMP <= URE
1723,                           .       DLSMP:                          (DL HL,HL ENTRY)
1724,                           .              MAR <= HL
1725,                           .         ELSE IF IMP EVEN
1726,                           .                THEN MAR <= BC
1727,                           .                ELSE IMP <= IMP - 1
1728,                           .                     IF IMP ZERO       (111 CASE)
1729,                           .                       THEN IMP <= URC
1730,                           .                            MAR <= HL
1731,                           .                       ELSE MAR <= DE
1732,                           .       IF LINK = DLOAD
1733,                           .         THEN DLOAD
1734,                           .         ELSE DSTORE
1735,                           .
1736,  001656L 01101111 11110000            BAS    LINK
1737,  001657L 11011001 01000100            BRA    DLSZRO,T0,IZ             DE,HL = 000
1738,  001660L 11011010 01001001            BRA    DLSBC,F0,IO              XX,BC = 062, 114...
1739,  001661L 00110111 00001001            STB    DIMP
1740,  001662L 00110001 11000100            LDX    DE2MRL                   DELAY ONE FROM THE DIMP BEFORE BRA
1741,  001663L 11011001 01000110            BRA    DLSONE,T0,IZ             BC,HL = 111
1742,  001664L 00110001 11100011            LDX    DE2MRH
1743,  001665L 11101111 00000000            BRR    LINK                     XX,DE = 113, 115...
1744,  001666L 00110001 11000010  DLSBC     LDX    XX2MRL+URC
1745,  001667L 00110001 11100001            LDX    XX2MRH+URB               XX,BC = 062, 114...
1746,  001670L 11101111 00000000            BRR    LINK
1747,  001671L 01010001 00000010  DLSONE    LDTI   URC                      BC,HL = 111
1748,  001672L 11011111 01000011            BRA    DLSLMP
1749,  001673L 01010001 00000100  DLSZRO    LDTI   URE                      DE,HL = 000
1750,  001674L 00110111 00000001  DLSLMP    LDPT   LIMP
1751,  001675L 00110001 11000110            DLDX   HL2MR                    XX,HL = 000, 111, HL,HL
       001676L 00110001 11100101
1752,  001677L 11101111 00000000            BRR    LINK
1753,                           *
1754,  001700L                  DLOAD
1755,                           .                                       LOAD REGISTER PAIR FROM MEMORY
1756,                           .
1757,                           .       IMP, IMP=1 <= (MAR, MAR+1)
1758,                           .
1759,  001700L 00110111 01000111            STB    SMR
1760,  001701L 11001110 11111111            MWAIT  NOOP,MEMPF1
       001702L 11010100 00111101
       001703L 11010111 00001001
1761,  001704L 00110001 00110110            LDTP   MDR                      GET THE LSB DATA
1762,  001705L 00110111 00001100            STB    IMAR,SMR
       001706L 00110111 01000111
1763,  001707L 00110111 10001111            LDPT   IMP0,DIMP                AND MORE OF THE SAME
```

```
                001710L 00110111 00001001
1764.           001711L 11010100 00110110          MWAIT  ,MEMPF1
                001712L 11010111 00001001
1765.           001713L 00110001 00110110          LDPP   IMPO,MDR              GET THE MSB
                001714L 00110111 10001111
1766.          >001715L 01011001 11111111          BRAX   FETCH
               >001716L 11001111 11111111
1767.
1768.           001717L                            DSTORE
1769.                                              .                            SAVE REGISTER PAIR IN MEMORY
1770.                                              .
1771.                                              .       (MAR, MAR+1) <- IMP, IMP-1
1772.                                              .
1773.           001717L 00110001 11011111          LDPP   MDW,IMPI,DIMP         OUTPUT LSB AND POINT TO MSB
                001720L 00110111 00100001
                001721L 00110111 00001001
1774.           001722L 00110001 11011111          LDTP   IMPI
1775.           001723L 11010100 00101100          MWAIT  ,MEMPF1
                001724L 11010111 00001001
1776.           001725L 00110111 00001100          STB    IMAR
1777.           001726L 00110111 00100001          LDPT   MDW                   OUTPUT MSB
1778.          >001727L 01011001 11111111          BRAX   FETCHW
               >001730L 11001111 11111111
1779.                                              .
```

```
1782,                                   *
1783,  001731L                          BRL:
1784,                                   . 4,55   (    072)  BRL                 BASE REGISTER LOAD
1785,                                   . 6,55   (IMP 072)  BRL                 BASE REGISTER LOAD FROM OTHER THAN A-REG
1786,                                   .
1787,                                   .       IF USER THEN IVIOL$
1788,                                   .         ELSE (SEBRLS) <- BASW <- IMP
1789,                                   .         FETCHW
1790,                                   .
1791.  001731L 00110001 11011100                TSTIP  ,SWUSER,PSWI
       001732L 01000101 00000100
1792,  001733L 11010010 00000111                BRA    IVIOL1,FZ
1793.  001734L 01010001 10101011                DLDPI  MARO,SEBRLS              SAVE BASE REG VALUE AWAY
       001735L 00110111 11000000
       001736L 01010001 11101111
       001737L 00110111 11100000
1794.  001740L 00110001 11011111                LDPP   BASW,IMPI                LOAD BASE REG BEFORE WRITE!
       001741L 00110111 00000011
1795.  001742L 00110111 00100001                LDPT   MDW
1796. >001743L 01011001 11111111                BRAX   FETCHW
      >001744L 11001111 11111111
1797,                                   *
1798,  001745L                          EI:
1799,                                   . 3,10   (    050)  EI                  ENABLE INTERRUPTS
1800,                                   .
1801,                                   . 8,55   (111 050)  EJMP,LSB,MSB        ENABLE INTERRUPTS AND JUMP TO ADDRESS
1802,                                   .        (IDD 050)  EJMP,LSB,MSB          UNSPECIFIED
1803,                                   .
1804,                                   . 10,20  (062 050)  EUR                 ENABLE INTERRUPTS AND RETURN
1805,                                   .        (IEV 050)  EUR                   UNSPECIFIED
1806,                                   .
1807,                                   .       IF USER THEN IVIOL$
1808,                                   .         ELSE MODW <- PSW <- PSW .OR. SWINTE
1809,                                   .              IF IMP ZERO THEN FETCHI
1810,                                   .                 ELSE EIROJ
1811,                                   .
1812.  001745L 00110001 11011100                TSTIP  ,SWUSER,PSWI             NOT ALLOWED IN USER MODE
       001746L 01000101 00000100
1813,  001747L 11010010 00000111                BRA    IVIOL1,FZ
1814.  001750L 01010011 00000001                DOPI   PSWO,OR,SWINTE           SET THE INTERRUPTS ON BIT
       001751L 00110111 10001100
1815.  001752L 00110111 00000100                LDPT   MODW                     IN EMULATION SUPPORT ALSO
1816. >001753L 01011001 11111111                BRAX   FETCHI,T0,IZ             WAS SIMPLE EI
      >001754L 11001001 11111111
1817. >001755L 11001111 11111111                BRA    EIROJ                    WAS EJMP OR EUR
```

```
1818*                                   *
1819*  001756L                         DI:
1820*                                  . 3.10   (    040)  DI                  DISABLE INTERRUPTS IF ALLOWED
1821*                                  .
1822*                                  .        IF USER THEN IVIOL$
1823*                                  .             ELSE MODW <= PSW <= PSW ,AND. (-1-SWINTE)
1824*                                  .             FETCHI
1825*                                  .
1826*  001756L 00110001 11011100               TSTIP  ,SWUSER,PSWI           NOT ALLOWED IN USER MODE
       001757L 01000101 00000100
1827*  001760L 11010010 00000111               BRA    IVIOL1,FZ
1828*  001761L 01010101 11111110               DOPI   PSWO,ND,-1-SWINTE      RESET THE BIT, DISABLING INTERRUPTS
       001762L 00110111 10001100
1829*  001763L 00110111 00000100               LDPT   MODW                   IN EMULATION SUPPORT ALSO
1830*  >001764L 01011001 11111111              BRAX   FETCHI
       >001765L 11001111 11111111
1831*                                  *
1832*  >001766L 01011001 11111111     MEMPF1   BRAX   MEMPF$
       >001767L 11001111 11111111
1833*                                  *
1834*  001770L 01011001 11111111      IVIOL1   BRAX   IVIOL$
       001771L 11001111 00110000
1835*                                  .
1836*  001772L 11111111 11111111               TABPAGE PROCL
       001773L 11111111 11111111
       001774L 11111111 11111111
       001775L 11111111 11111111
       001776L 11111111 11111111
       001777L 11111111 11111111
1837*  002000                         PROCLEN  EQU    $-PROCP
1838*  000000                                  USE    PROCL
1839*  000000                                  SKIP   PROCLEN
1840*  000000                                  END    POR
```

```
        AC              107     763     879     882     898    1004    1018    1029    1035    1153    1158    1159
                       1248    1254    1299    1305    1307    1321    1337    1357    1500    1503    1642    1644
010017  ACCTL           *77#A    88
010015  ACD             *74#A
010017  ACPH            *76#A
010016  ACPL            *75#A
        ACUOT            91
000417  ALPHA           *591
000534  ALPHAL          *733     616     617     665
000413  ALPHAS          *577     588
000001  APF              *9#A   120#A    84      87
010014  APFRK           *68#A
010013  APFRP           *67#A
010016  APFTK           *70#A
010015  APFTP           *69#A
        B6              123#A
        BASW             81     1794
001126  BCP            *1081
000400  BETA            *546
000471  BETAL           *668     571     572     730
000432  BETAS           *622     589
001226  BFAC           *1210
001270  BFADD          *1254    1221
001231  BFOP           *1223    1146    1148    1208
001300  BFS            *1265
001222  BFSB           *1197
001326  BFSL           *1302    1287
001274  BFSUB          *1258    1208
000302  BP              *395
020006  BR               *42#A
001731  BRL            *1783
001000  BT              *960
001032  BTCVT          *1015    1010
001103  BTEND          *1035    1021
001051  BTR            *1023     999
001021  BTX            *1012     998
        C0               879     882    1018    1029    1035    1299    1321    1337    1357    1500    1554    1642
                        1658
        C1               118     329    1013    1020    1102    1106    1289
000100  CAP55IO         *123#A  126#A
000315  CAPABILI        *126#A
000010  CAPAPF          *120#A  126#A
000004  CAPBLUE         *119#A  126#A
000200  CAPCOM          *124#A  126#A
000000  CAPCYNT         *121#A  126#A
000000  CAPGUPY         *118#A  126#A
000001  CAPMICR         *117#A  126#A
000000  CAPRIM          *122#A  126#A
        CC               98     330     371     760     868     898    1001    1024    1030    1040    1107    1177
                        1239    1244    1291    1292    1295    1297    1302    1333    1334    1350    1351    1353
                        1355
001402  CCS            *1362
```

```
007000  CDOR          *134:A
006000  CDOX          *133:A
        CF             82     83     88     92     94     97    114    118    380    420    433    434
                      484    506    642    644    792    795    805    807    808    866    872    877
                      881    883    888    898    911    915   1015   1019   1026   1104   1144   1151
                     1158   1161   1207   1220   1242   1249   1254   1258   1298   1320   1336   1356
                     1499   1553   1621   1625   1632   1642   1644   1649   1658   1660   1665   1695
                     1736
        CMPF           77
        CODMIN        330
        CODMOUT       371
        CODWRT        363
030011  COMMODE       *96:A    92
        DE2MRH        1000   1017   1028   1101   1245   1742
        DE2MRL        1000   1017   1028   1101   1245   1740
001624  DECIT         *1655  1579
001517  DECP          *1528
001512  DECPA         *1508
001515  DECPAS        *1525  1523
001526  DECPIT        *1553  1526   1550
001522  DECPS         *1549  1547
001541  DECX          *1563
001177  DFADD         *1157  1145
001205  DFAEND        *1162  1160
001162  DFOP          *1116  1100
001174  DFSEND        *1154  1152
001170  DFSUB         *1150  1147
001756  DI            *1819
        DIMP           648    797    830    945    949   1498   1502   1552   1556   1631   1663   1739
                      1763   1773
001655  DL            *1699
001651  DLHL          *1688
001700  DLOAD         *1754  1695   1715
001656  DLS           *1717  1686
001666  DLSBC         *1744  1738
001674  DLSLMP        *1750  1697   1748
001671  DLSONE        *1747  1741
001673  DLSZRO        *1749  1737
        DMAR           115    491    521    651    717    722    764    800    804    833   1651
001647  DS            *1671
001717  DSTORE        *1768  1686
001745  EI            *1798
        EIROJ         1817
000104  EXADR         *182
000134  EXDATA        *216
000123  EXSTAT        *201
        FETCH         1411   1666   1766
        FETCHI         955   1323   1338   1358   1369   1501   1506   1555   1558   1816   1830
        FETCHW         156    812    836   1192   1450   1653   1778   1796
000054  FETPC         *123    538
001436  FETPL         *1411  1407
001461  FETPS         *1450  1445
```

```
004000  FLEX          *1321A
000631  FTCHABW       *812    656
000772  FTCHIAB       *954    575    620    874
000072  FTCHIO        *156    274
        HL2MRH         322    356    828    895   1012   1023   1105   1240   1286   1751
        HL2MRL         322    356    828   1012   1023   1105   1240   1286   1751
        I35I          1444
        I350          1406
        IDCODH        1075
        IDCODL         387   1075
        IIMP          1664
        IMAR           117    369    421    791    796    810    896   1400   1408   1439   1448   1620
                      1624   1630   1641   1657   1762   1776
        IMP8           647    788    829
        IMPFO         1500   1503   1554   1557   1664
        IMPI           193    198    210    225    262    649    798    878   1320   1332   1352   1446
                      1500   1503   1554   1557   1773   1774   1794
        IMPO           153    834   1322   1335   1354   1368   1410   1665   1763   1765
        INBUS          153    328
001577  INCIT         *1639   1597
001470  INCP          *1474
001463  INCPA         *1455
001466  INCPAS        *1471   1469
001477  INCPIT        *1499   1472   1496
001473  INCPS         *1495   1493
001543  INCX          *1581
001565  INDGET        *1632   1627
001544  INDX          *1599   1579
001640  INDXL         *1663   1645
001613  INDXS         *1647   1661
000061  INPUT          *135
000064  INPUTX         *153    177
000067  INPW1          *154    155
020005  IO              *41:A  572    617    999   1146   1496   1550   1627   1738
        IT             118    329    884   1007   1013   1020   1102   1106   1289   1322
        ITW           1035
000317  IVIOL$         *449    152    175    195    212    227    264    320    358    569    614    862
                      1834
001770  IVIOL1        *1834   1792   1813   1827
020004  IZ              *40:A  486    574    618    653    786    802    835    876    944    948    951
                       998   1100   1287   1407   1445   1469   1493   1523   1547   1645   1661   1737
                      1741   1816
        KBSC            95
010001  KBSCNT          *49:A  94
        LIMP           525    647    714    788    829   1470   1494   1524   1548   1750
030000  LINK            *82:A 1239   1252   1621   1637   1695   1736   1743   1746   1752
        LIREG          363    380
        LUCF          1299   1308   1321   1337   1357
        LUF            765   1109   1151   1159   1255   1259   1648
010003  MADR            *54:A  97
        MARIH          329    805   1013   1020   1025   1031   1102   1106   1241   1246   1289   1303
        MARIL          329    805   1013   1020   1025   1031   1102   1106   1241   1246   1289   1303
```

```
         MAROH          106     112     197     517     794     884     891     912    1007    1793
         MAROL          113     197     644     680     745     795     882    1004    1403    1442    1635    1793
010004   MBITS          *55:A    83
010005   MBSTAT         *56:A    82
010006   MCRCH          *57:A
010007   MCRCL          *58:A
         MDR            362     724     763     790     794     832     893    1004    1018    1029    1104    1109
                       1244    1251    1294    1307    1403    1406    1410    1442    1623    1629    1635    1640
                       1644    1656    1660    1761    1765
010010   MDSKS          *59:A
010011   MDSKT          *60:A
         MDW            117     198     328     523     646     652     801     807     811    1018    1029    1154
                       1162    1254    1258    1296    1307    1444    1449    1647    1652    1773    1777    1795
         MEMPFS         540    1832
000363   MEMPF0         *540     361     385     482     650     664     723     761     789     793     799     803
                        809     831
001766   MEMPF1        *1832    1003    1016    1027    1074    1103    1108    1243    1250    1293    1306    1402
                       1405    1409    1441    1447    1622    1628    1634    1639    1643    1650    1655    1659
                       1760    1764    1775
000361   MEMPFSC        *529
000200   MIN            *292
000262   MINOUT         *372     331
000172   MINOWT         *278     379
000102   MINPAR         *178     327
000210   MINW0          *324     325
000262   MINW1          *373     374
020002   MO             *38:A    76     120     361     385     482     526     650     664     723     761     789
                        793     799     803     809     831     890     909    1003    1016    1027    1074    1103
                       1108    1243    1250    1293    1306    1402    1405    1409    1441    1447    1622    1628
                       1634    1639    1643    1650    1655    1659    1760    1764    1775
000544   MODLOD         *750     684
000636   MODLOP         *829     766
000433   MODSAV         *625     578
000503   MODSRV         *686     662
000446   MODSVLP        *648     653
         MODW            80     289     321     359     522     657     724     762     892     894    1815    1829
000170   MOTPAR         *275     367
000247   MOTW0          *364     365
000230   MOUT           *333
020003   MP             *39:A    361     385     482     527     650     664     723     761     789     793     799
                        803     809     831     890     909    1003    1016    1027    1074    1103    1108    1243
                       1250    1293    1306    1402    1405    1409    1441    1447    1622    1628    1634    1639
                       1643    1650    1655    1659    1760    1764    1775
         MR2HLH         370     897
         MR2HLL         370
         MR2PCH         497     718    1401    1440    1633
         MR2PCL         497     718    1401    1440    1633
010013   MSECT          *62:A
010012   MTRAK          *61:A
         OTBUS          193     210     225     262     362
000145   OUTPUT         *231
000153   OUTW0          *265     199     214     229     266
```

```
000157  OUTW1           *268     269     290
000163  OUTW2           *271     272
        PC2MRH           491     717
        PC2MRL           491     717
        PCL              525     714
        PCMOD            527
        PCOH             125
        PCOL              99     123
010000  PDLNP            *48:A   98
000074  PIN              *159
001407  PLR              *1374
        POPST0           715
000000  POR              *70     1840
000031  PORLST           *106    108
000055  POROUT           *125    101
000045  PORZRO           *117    121
000040  PRE               *3:A
000000  PROC             *130:A  62
000000  PROCL            *64     66      542     956     1836
002000  PROCLEN          *1837   1839
000000  PROCP            *66     1837
002000  PROD             *131:A
        PSHST0           527
001440  PSR              *1413
        PSWI             151     174     194     211     226     263     288     319     357     446     522     523
                         568     613     654     663     682     747     861     894     947     1191    1791    1812
                         1826
        PSWO              79     196     213     228     288     524     576     621     724     759     1191    1814
                         1828
010002  Q                *46:A   329     760     868     1001    1013    1020    1024    1040    1102    1106    1177
                         1289    1291    1292    1295    1297    1302    1333    1334    1350    1351    1353    1355
030006  RCRCH            *93:A
030007  RCRCL            *94:A
030005  RDATA            *92:A
000633  REGL             *815    786
000640  REGLLP           *830    835
000557  REGS             *770
000602  REGSLP           *797    802
        RETS             130
000011  REV               *2:A
001207  RIN16            *1168   1155    1256    1260    1300    1309
001110  RIN256           *1044   1036    1110
001215  RIND             *1182   914     916     1037    1057    1111
001120  RINDO            *1074   1059
001124  RINRPT           *1077   1073
000274  RINSID           *382
001116  RINST            *1062   1180
030004  RPNTR            *91:A
030003  RSTAT            *90:A
        SB               377     1025    1031    1056    1109    1151    1178    1241    1246    1258    1303    1554
                         1557    1658    1660
000304  SC               *404
```

```
         STWE              105
         SVBKPNT           401
         SVINP             178
         SVIVIOL           453
         SVOUTP            275
         SVSCAL            410
         SVUAINS           467
         SWALBT            573     619     663     682     729     747
         SWBASD            892
         SWINTE            321     359     657    1814    1828
         SWRPT             288     570     576     615     621     654     683     747     863    1191
         SWSCF             522     524
         SWSTDT            196     213     228
         SWUSER            151     174     194     211     226     263     319     357     446     568     613     861
                          1791    1812    1826
         SYSESR            112     113
         SYSIVR            506
000505   SYSRET           *688     618
         SYSRET0           715
000512   SYSRET1          *717
         SYSROM            110     125     434     517     912
000415   SYSSAV           *580     574
000757   SYSTAT           *923
000771   SYSTND           *953     948     951
030001   TEMP1             *83:A    85:A    330     371     380     877     879     881     888     898     915    1015
                          1018    1019    1026    1029    1030    1035    1104    1107    1151    1158    1161    1244
                          1254    1258    1298    1299    1320    1321    1336    1337    1356    1357    1499    1500
                          1553    1554
030002   TEMP2             *84:A    86:A    642     644     866     872     882     883     911    1144    1207    1220
                          1242    1249
030001   TEMPH             *85:A    434     506     805     808    1632    1644    1660
030002   TEMPL             *86:A    114     118     420     433     484     792     795     805     807    1625    1642
                          1649    1658    1665
000001   TYPE               *5:A    12:A   117:A   119:A   124:A
         UCFLG              646    1248    1290    1305    1349    1368
         UDEH              1020    1031    1102    1246
         UDEL              1020    1031    1102    1246
000321   UDOP             *456     447     944
000314   UDPOP            *437
         URA               1005    1014    1026    1471    1525
         URB                953    1035    1154    1162    1745
         URC                377     866     873     946    1056    1178    1744    1747
         URE               1749
         URFO               377
         URH                329     883     886     911    1006    1013    1025    1106    1241    1289    1303
         URI                377     866     873     882     883    1002    1006    1014    1026    1035    1056    1154
                          1162    1178    1471    1525
         URL                329     882    1002    1013    1025    1106    1241    1289    1303    1470    1494    1524
                          1548    1696
         URO                329     886     911     946     953    1005    1013    1020    1025    1031    1056    1102
                          1106    1178    1241    1246    1289    1303
030012   URPNTR           *97:A
```

```
        URX             1404    1443    1636
030010  UXPNTR           *95:A
000002  VER               *1:A
030016  XCRCH           *101:A
030017  XCRCL           *102:A
030015  XDATA           *100:A
030014  XPNTR            *99:A
030013  XSTAT            *98:A
        XX2MRH          1404    1443    1636    1745
        XX2MRL          1744
```