

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 PROD,,,PROC;GBQLX

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: PROD

| | | | | |
|-------------------------|--------|------------|-------------|-------|
| PROGRAM ADDRESS BLOCKS: | 010000 | /ABSOLUTE/ | SIZE=000000 | (ABS) |
| | 167400 | /SYSIVR/ | SIZE=000400 | (ABS) |
| | 170000 | /SYSROM/ | SIZE=000047 | (ABS) |
| | 002000 | /PRODL/ | SIZE=002000 | (ABS) |
| | 000000 | /PRODP/ | SIZE=004000 | (REL) |

EXTERNAL DEFINITIONS:

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|---------|--------|
| SIR0 | 002000 | SIRX | 002010 | FETCHW | 002020 | FETCH | 002022 |
| FETCHI | 002024 | FETCHS | 002026 | SRVDD | 002620 | CALLCC | 002042 |
| NOJ | 002426 | CALL | 002046 | MEMPFS | 002225 | PCMOD | 002110 |
| PSHST0 | 002070 | PUSH | 002120 | PUSHI | 002126 | FIROJ | 002152 |
| JUMP | 002433 | RETURN | 002163 | RETCC | 002160 | RETS | 002212 |
| POPST0 | 002166 | POP | 002216 | STKS | 002305 | SYSRET0 | 002374 |
| JUMPCC | 002423 | LD6 | 002454 | LD7 | 002464 | L7S | 002502 |
| LDS | 002513 | AP4 | 002516 | AP7 | 002525 | APS | 002540 |
| SRVRPT | 002611 | SRVNXT | 002616 | SRVRTW | 002634 | ACSDO | 003400 |
| IRNXT | 003611 | ITINT | 003666 | IRINT | 003600 | SRVBSN | 002755 |
| DLSDO | 003000 | LODCF | 003255 | BEEP | 003335 | CLICK | 003351 |
| MTART | 003424 | ITNXT | 003670 | AMLRET | 003550 | | |

EXTERNAL REFERENCES (UNDEFINED SYMBOLS):

SCROML SCRAM RIN16 SYSRET1 IVIOLS MBPAGE SCLST SCROM

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```

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

31,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
42,
43,
44,
45,
46,
47,
48,
49,
50,
51,
52,
53,
54,
55,
56,
57,
58,
59,
60,
61,
62,
63,
64,
65,
66,
67,
68.

```

*
. USEFUL INFORMATION TO BE PASSED ON:
*
. *****
. IF SECTOR CONTAINING SEKBS1 IS WRITE PROTECTED THEN CAN NOT DO A RESTART!!!!
. *****
. DLSDO CODE, ACCSDO & SYSRET FORCE! PRIVED MODE TO BE SURE CAN WRITE/ACCESS
. MEMORY THAT PROTECTED, FXIO, COMM MUST DO THIS ALSO!!
. *****
. ALL SMR'S AND MDW'S MUST HAVE FOLLOWING MWAIT ,MEMPFX EVERYWHERE
. IN THE CODE EVEN IF MWAIT ,S+1 IS USED (IGNORING MEMORY FAULT)
. *****
. FOR TESTING MEMPF, CHANGE MWAIT MACRO TO BE JT,MP S+1 <<<
. CODE THERE, TESTABLE AS USED BUT NO EFFECT!!
. *****
. EACH SERVICE ROUTINE THAT USES THE MAR MUST HAVE A MWAIT ,S+1 BEFORE THE
. MAR CHANGED, IF NOT USED, RETURN THROUGH SRVNXT!
. *****
. ALL INPUTS FROM A PORT INSTRUCTION SHOULD BE LOGICAL (OUTPUT CAN ONLY BE
. LOGICAL OR 'IT').
. ARITHMETICS MAY BE DONE ON INPUTS WITH CAUTION. ONLY IF THE DATA HAS NO SETUP
. TIME, . . MDR IS OK, BUT IMPI OR URI ARE NO-NO'S.
. *****
. DELAY AROUND SECTOR TABLE WRITES IS NOT NECESSARY.
. *****
. DELAY DURING FONT LOAD IS ONLY AFTER LDCH. A 400 SEC. DELAY IS NECESSARY
. FROM AFTER LDCH TO ANY OF RDLM, SOLM, SKCH, MARQL, OR LDMAP.
. ALSO, RDLM AND SOLM SHOULD NOT BE TOO CLOSE TOGETHER.
. *****

```

```

.
.      INC      PROCPARM
*
PRODL  ORG      PROD      LOGICAL SPACE DEFINED IN PLACE
PRODP  ORG      0         PHYSICAL SPACE RELOCATABLE
PRODL  USE      PRODL     USE THEM BOTH
PRODP  USE      PRODP     PUT THE CODE IN PHYSICAL SPACE
PRODP  LOC      PRODL,2   WITH ADDRESSES IN LOGICAL SPACE
.

```

```

002000
000000
002000
000000
002000L

```

```

71.
72. 002000L
73.
74.
75.
76.
77.
78.
79. 002000L 00110111 00001100
    002001L 00110111 01000111
80. 002002L 11011000 11101110
81. 002003L 01010001 00010000
    002004L 00110111 00000001
82. 002005L 00110001 10001001
    002006L 00110001 10101000
83. 002007L 11001111 11100011
84.
85. 002010L
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98. 002010L 00110111 00001100
    002011L 00110111 01000111
99. 002012L 11011000 11101110
100. 002013L 01010101 00000111
    002014L 00110111 00000001
101. 002015L 00110001 10001001
    002016L 00110001 10101000
102. 002017L 11001111 11100011
103.
    
```

```

*
SIR0:
.   (   022)  XA           SELECT XA REGISTER PAIR
.
.   T <= IMP8
.   MAR <= MAR + 1
.   FITCH
.
.   STB   IMAR,SMR        START TO GET THE OP-CODE
.
.   BRA   IVIOL2,F0,IZ    MULTIPLE IMPLICIT IN A ROW IS ERROR
.   LDPI  LIMP,IMP8
.
.   DLDX  MR2PC
.
.   BRA   FETLIMP
.
*
SIRX:
.   (   111)  B           SELECT B OR IMP ODD
.   (   062)  C           SELECT C OR BC OR IMP EVEN
.   (   113)  D           SELECT D
.   (   174)  E           SELECT E OR DE
.   (   115)  H           SELECT H
.   (   176)  L           SELECT L OR HL
.   (   117)  X           SELECT X
.
.   IMP <= T .AND. 7     T CONTAINS OPCODE FROM FETCH
.   MAR <= MAR + 1
.   FITCH
.
.   STB   IMAR,SMR        START TO GET REAL OP-CODE
.
.   BRA   IVIOL2,F0,IZ    MULTIPLE IMPLICIT IN A ROW IS ERROR
.   DOPI  LIMP,ND,7
.
.   DLDX  MR2PC
.
.   BRA   FETLIMP
.
    
```


| | | | | | | |
|-----|---------|----------|----------|-----------------------------|--|--------------------------------------|
| 142 | | | | * | | |
| 143 | 002042L | | | CALLCC: | | |
| 144 | | | | 9.95 (102) CCC,LSB,MSB | | CONDITIONAL CALL |
| 145 | | | | 3.10 IF NOT TAKEN | | |
| 146 | | | | (102) CFC,LSB,MSB | | FALSE CARRY |
| 147 | | | | (112) CFZ,LSB,MSB | | FALSE ZERO (.NE.) |
| 148 | | | | (122) CFS,LSB,MSB | | FALSE SIGN (.GE.) |
| 149 | | | | (132) CFP,LSB,MSB | | FALSE PARITY (EVEN) |
| 150 | | | | (142) CTC,LSB,MSB | | TRUE CARRY |
| 151 | | | | (152) CTZ,LSB,MSB | | TRUE ZERO (.EQ.) |
| 152 | | | | (162) CTS,LSB,MSB | | TRUE SIGN (.LT.) |
| 153 | | | | (172) CTP,LSB,MSB | | TRUE PARITY (ODD) |
| 154 | | | | | | |
| 155 | | | | 9.15 (111 102) UR | | USER RETURN |
| 156 | | | | (IMP 102) UR | | PAGED OPERATION TO A REGISTER |
| 157 | | | | | | |
| 158 | | | | IF IMP NOT ZERO THEN USERET | | |
| 159 | | | | ELSE IF NOT CC THEN NOJ | | |
| 160 | | | | ELSE CALL | | |
| 161 | | | | | | |
| 162 | 002042L | 11001000 | 10010100 | BRA USERET,F0,IZ | | USER RETURN, NOT A CALL! |
| 163 | 002043L | 00110001 | 00110001 | TSTIP ,STUSCF,STATUS | | IS USER CC CORRECT? |
| | 002044L | 01000101 | 00000001 | | | |
| 164 | 002045L | 11010011 | 11101001 | BRA NOJ,TZ | | NO, DO NOJ NOT A RETURN |
| 165 | | | | | | |
| 166 | 002046L | | | * CALL: | | |
| 167 | | | | 9.50 (106) CALL,LSB,MSB | | UNCONDITIONAL CALL |
| 168 | | | | (IMP 106) PAD,LOC | | PAGED ADD TO A REGISTER |
| 169 | | | | | | |
| 170 | | | | MAR <= MAR + 1 | | |
| 171 | | | | IMP <= PC | | PUSH PC ON THE STACK |
| 172 | | | | TEMPL <= (MAR) | | |
| 173 | | | | MAR <= MAR + 1 | | |
| 174 | | | | TEMPH <= (MAR) | | |
| 175 | | | | PC <= MAR <= MAR + 1 | | |
| 176 | | | | PSHSTK (PCMOD) | | RETURN TO PCMOD (LOADS PC FROM TEMP) |
| 177 | | | | | | |
| 178 | 002046L | 00110111 | 00001100 | STB IMAR,SMR | | GET THE ADDRESS TO CALL |
| | 002047L | 00110111 | 01000111 | | | |
| 179 | 002050L | 01010001 | 00001001 | LDPI LIMP,PCL | | PC TO BE PUSHED ONTO THE STACK |
| | 002051L | 00110111 | 00000001 | | | |
| 180 | 002052L | 11000100 | 11010101 | MWAIT ,MEMPF\$ | | |
| | 002053L | 11000111 | 01101010 | | | |
| 181 | 002054L | 00110001 | 00110110 | LDTP MDR | | |
| 182 | 002055L | 00110111 | 00001100 | STB IMAR,SMR | | |
| | 002056L | 00110111 | 01000111 | | | |
| 183 | 002057L | 01101111 | 11110010 | LDRT TEMPL | | SAVE LSB |
| 184 | 002060L | 11000100 | 11001111 | MWAIT ,MEMPF\$ | | |
| | 002061L | 11000111 | 01101010 | | | |
| 185 | 002062L | 00110001 | 00110110 | LDRP TEMPH,MDR | | SAVE MSB |
| | 002063L | 01101111 | 11110001 | | | |
| 186 | 002064L | 00110111 | 00001100 | STB IMAR | | POINT MAR TO THE NEXT INSTRUCTION |

187. 002065L 00110001 10001001
002066L 00110001 10101000
188. 002067L 01010001 10110111

DLDX MR2PC

SO PUSHED P.C. IS CORRECT

BAL ,PCMOD


```

189.
190. 002070L 01101111 10110000
191. 002071L
192.
193.
194.
195.
196.
197.
198.
199.
200. 002071L 00110001 11011011
    002072L 01010100 00000010
201. 002073L 01010011 01000000
    002074L 00110111 10001011
202. 002075L 00110111 11000000
203. 002076L 00110001 11101010
204. 002077L 00110001 11011111
    002100L 00110111 00100001
205. 002101L 00110111 00001001
206. 002102L 00110001 11011111
207. 002103L 11000100 10111100
    002104L 11000111 10111000
208. 002105L 00110111 00001100
209. 002106L 00110111 00100001
210. 002107L 11101111 00000000
211.
    
```

```

+
PSHST0: BAS LINK,CC IN=PAGE ENTRY TO PUSH
PSHSTK OFF=PAGE ENTRY (LINK RETURN PRE-LOADED)
. 2.30 + 0.20 INTO MEMORY WAIT MUST HAVE CARRY CLEAR TO USE THIS ENTRY
.
. NOTE: CAN NOT PARITY CHECK IN HERE OR LOST FOREVER WITH
. MEMPFS & SYS CALL & PUSH !!!
.
. SP <- SP - 2 (MODULO 128)
. (SP,SP+1) <- IMP, IMP-1 LSB THEN MSB
.
. DOTIP ,SB,2,SPIL UPDATE THE SP
.
. DOPI SPOL,OR,0100 MAKE SURE EVEN & WRAP AROUND (MOD 32/64)
.
. LDPT MARDL SAVE UPDATED LSB
. LDX SP2MRH POINT TO MEMORY STACK
. LDPP MDW,IMPI STORE LSB THERE
.
. STB DIMP
. LDTP IMPI PRE=LOAD MSB DATA TO WRITE
. MWAIT ,MEMPFS *** LET THE CALLER WORRY ABOUT IT ***
.
. STB IMAR NOW POINT & DO THE MSB TO FINISH OFF
. LDPT MDW
MEMPFS BRR LINK
. *** NOTE ROUTINE CALLING PSHSTK OR PSHST0 MUST HAVE MWAIT ,ADDR ***
    
```

```

212.
213. 002110L
214.
215.
216.
217.
218. 002110L 11000100 10110111
      002111L 11000111 01101010
219. 002112L 01110001 11110010
      002113L 00110111 11000000
      002114L 01110001 11110001
      002115L 00110111 11100000
      002116L 00110111 01000111
      002117L 11001111 11101001
220.
221.
222. 002120L
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234. 002120L 01010001 11101111
      002121L 01101111 10110000
      002122L 11001000 11000110
235. 002123L 01010001 00000110
      002124L 00110111 00000001
      002125L 11001111 11000110
236.
237.
238. 002126L
239.
240.
241.
242.
243.
244.
245.
246.
247. 002126L 00110111 00001100
      002127L 00110111 01000111
248. 002130L 01010001 00001111
      002131L 00110111 00000001
149. 002132L 11000100 10100101
      002133L 11000111 01101010
150. 002134L 00110001 00110110
151. 002135L 00110111 00001100
      002136L 00110111 01000111
    
```

```

*
PCMOD: RETURN FROM PSHSTK AND SET NEW PC
* * * * 3.60 * * * * WITH WAIT FROM PSHSTK & FETCHS CODE
.
.   MAR <= TEMP
.
.   MWAIT ,MEMPFS
.
.   DL DPR MAR0,TEMP,,SMR LOAD UP SAVED P.C.
.
.   BRA   FETCHS GO TO ADDRESS AT MACRO LEVEL
*
PUSH:
. 6.40 ( 070) PUSH HL PUSHED ONTO THE STACK
. 8.05 (062 070) PUSH BC
. 8.05 (174 070) PUSH DE
. 8.05 (022 070) PUSH XA
.
.   (176 070) PUSH HL UNSPECIFIED
.   (I0D 070) PUSH MIXED REG PAIR UNSPECIFIED
.
.   IF IMP ZERO THEN IMP <= URL
.   PSHSTK (FETCHW)
.
.   BRS   PSHSTK,F0,IZ,FETCHW,CC REG PAIR SPECIFIED!
.
.   LDPI  LIMP,URL DEFAULT
.
.   BRA   PSHSTK
*
PUSH1:
. 9.30 ( 051) PUSH,LSB,MSB PUSH CONSTANT (ADDRESS?) ONTO STACK
      (IMP 051) LINKED LIST INSTRUCTIONS
.
.   IMP <= IMP POINT IMP TO ITS HIDDEN REGISTERS
.   IMP, IMP-1 <= (MAR+1, MAR+2) GET ADDRESS AFTER OPCODE
.   PC <= MAR + 2 CORRECT THE PC
.   PSHST0 (FETCHW)
.
.   STB   IMAR,SMR GET THE IMMEDIATE VALUE
.
.   LDPI  LIMP,IMPL AND HIDING PLACE
.
.   MWAIT ,MEMPFS
.
.   LDTP  MDR
.   STB   IMAR,SMR
    
```

| | | | | | | |
|------|---------|----------|----------|-------|----------------|-----------------------------------|
| 252. | 002137L | 00110111 | 10001111 | LDPT | IMPO,DIMP | GOT LSB, POINT TO MSB |
| | 002140L | 00110111 | 00001001 | | | |
| 253. | 002141L | 00110001 | 10001001 | DLDX | MR2PC | SAVE THE UPDATED P.C. |
| | 002142L | 00110001 | 10101000 | | | |
| 254. | 002143L | 11000100 | 10011100 | MWAIT | ,MEMPFS | |
| | 002144L | 11000111 | 01101010 | | | |
| 255. | 002145L | 00110001 | 00110110 | LDP | IMPO,MDR,IIMP | GOT MSB, POINT CORRECTLY FOR PUSH |
| | 002146L | 00110111 | 10001111 | | | |
| | 002147L | 00110111 | 00001000 | | | |
| 256. | 002150L | 01010001 | 11101111 | BRC | PSHST0,,FETCHW | RETURN FROM PUSH TO FETCH WAIT |
| | 002151L | 11001111 | 11000111 | | | |
| 257. | | | | | | |

```

260,
261, 002152L
262,
263,
264,
265,
266,
267,
268,
269,
270,
271, 002152L 11011011 11100100
272,
273, 002153L
274,
275,
276,
277,
278,
279,
280,
281,
282, 002153L 00110001 11011100
    002154L 01010011 00000100
    002155L 00110111 10001100
283, 002156L 00110111 00000100
284, 002157L 11001111 10001100
285,
286, 002160L
287,
288,
289,
290,
291,
292,
293,
294,
295,
296,
297,
298,
299,
300,
301,
302,
303,
304,
305,
306, 002160L 00110001 00110001
    002161L 01000101 00000001
307, 002162L 11000011 11101011
    
```

```

*
EIROJI
.
. 10.20 (062 050) EUR          ENABLE INTERRUPTS AND USER RETURN
      (IEV 050) EUR          ENABLE INTERRUPTS AND USER RETURN
.
. 8.55 (111 050) EJMP,LSB,MSB  ENABLE INTERRUPTS AND JUMP
      (IOD 050) EJMP,LSB,MSB  ENABLE INTERRUPTS AND JUMP
.
. IF IMP ODD THEN JUMP
.
. BRA JUMP,T0,IO
*
USERET
. 10.20 (062 050) EUR          ENABLE INTERRUPTS AND USER RETURN
      (IEV 050) EUR          EIR UNSPECIFIED
. 9.15 (111 102) UR          USER RETURN
      (IMP 102) UR          UNSPECIFIED UR
.
. MODW <- PSW <- SWUSER .OR, PSW
. RETURN
.
. DOPIP PSW0,OR,SWUSER,PSWI  SET USER MODE
.
. STB MODW
. BRA RETURN
.
. EXTERNALLY ALSO IN EMULATION AIDS
*
RETCCI
. 6.55 ( 003) RET          CONDITIONAL RETURN
. 2.50 IF NOT TAKEN
      (IEV 003) DOPM        FROM MEMORY DOUBLE REGISTER PAIR ARITH'S
      (IOD 003) DOPP,IDX    PAGED DOUBLE REGISTER PAIR ARITHMETICS
      (IMP 003)             UNSPECIFIED
      (IMP 023)             UNSPECIFIED
.
. ( 003) RFC          FALSE CARRY
. ( 013) RFZ          FALSE ZERO (.NE.)
. ( 023) RFS          FALSE SIGN (.GE.)
. ( 033) RFP          FALSE PARITY (EVEN)
. ( 043) RTC          TRUE CARRY
. ( 053) RTZ          TRUE ZERO (.EQ.)
. ( 063) RTS          TRUE SIGN (.LT.)
. ( 073) RTP          TRUE PARITY (ODD)
.
. IF NOT CC THEN FETCH
. ELSE RETURN
.
. TSTIP ,STUSCF,STATUS
.
. BRA FETCHI,TZ
    
```

| | | | | | | |
|-----|---------|----------|----------|---------------------------------|--|--|
| 308 | | | | * | | |
| 309 | 002163L | | | RETURN: | | |
| 310 | | | | . 6.20 (007) RET | | UNCONDITIONAL RETURN |
| 311 | | | | . | | |
| 312 | | | | IMP ← PC | | |
| 313 | | | | POPSTK (RETS) | | |
| 314 | | | | . | | |
| 315 | 002163L | 01010001 | 00001001 | LDPI LIMP,PCL | | POP STACK INTO PROGRAMME COUNTER |
| | 002164L | 00110111 | 00000001 | | | |
| 316 | 002165L | 01010001 | 01110101 | BAL ,RETS | | RETURN FROM POP TO SPECIAL STARTUP |
| 317 | | | | * | | |
| 318 | 002166L | 01101111 | 10110000 | POPST0: BAS LINK,CC | | IN-PAGE CALL |
| 319 | 002167L | | | POPSTK | | POP THE STACK TO IMP AND IMP=1 |
| 320 | | | | . 3.35 | | CARRY MUST BE CLEAR TO USE THIS ENTRY |
| 321 | | | | . | | |
| 322 | | | | POPSTK: | | |
| 323 | | | | IMP, IMP=1 ← (SP, SP+1) | | |
| 324 | | | | SP ← SP + 2 (BASE 64 & MOD 128) | | |
| 325 | | | | | | (BITS 7 TO 15 UNCHANGED, BIT 6 IS 1 & BIT 0 IS 0 ALWAYS) |
| 326 | | | | . | | |
| 327 | 002167L | 00110001 | 11101010 | LDX SP2MRH | | POINT INTO THE STACK AREA |
| 328 | 002170L | 00110001 | 11001011 | LDX SP2MRL,SMR | | (SAVE 150 N.SEC BY INVERTING ORDER) |
| | 002171L | 00110111 | 01000111 | | | |
| 329 | 002172L | 01010101 | 10111111 | DOTI ,ND,-1-0100 | | TURN OFF WRAP-AROUND BIT |
| 330 | 002173L | 01010010 | 00000010 | DOTI ,AC,2 | | SKIP TO NEXT ENTRY TO POP |
| 331 | | | | . | | NOTE: (STKS) ASSUMES FALSE CARRY |
| 332 | | | | . | | GENERATED HERE (TO SAVE BYTE) |
| 333 | 002174L | 01010011 | 01000000 | DOPI SPOL,OR,0100 | | SET BIT ON TO FINISH WRAP-AROUND |
| | 002175L | 00110111 | 10001011 | | | |
| 334 | 002176L | 11000100 | 10000001 | MWAIT ,MEMPFS | | |
| | 002177L | 11000111 | 01101010 | | | |
| 335 | 002200L | 00110001 | 00110110 | LDTP MDR | | |
| 336 | 002201L | 00110111 | 00001100 | STB IMAR,SMR | | GET MSB DATA |
| | 002202L | 00110111 | 01000111 | | | |
| 337 | 002203L | 00110111 | 10001111 | LDPT IMPO | | SAVE LSB DATA |
| 338 | 002204L | 00110111 | 00001001 | STB DIMP | | |
| 339 | 002205L | 11000100 | 01111010 | MWAIT ,MEMPFS | | ** PUT MEMPFS AFTER LDPP, BETTER STACK! |
| | 002206L | 11000111 | 01101010 | | | |
| 340 | 002207L | 00110001 | 00110110 | LDPP IMPO,MDR | | SAVE MSB DATA |
| | 002210L | 00110111 | 10001111 | | | |
| 341 | 002211L | 11101111 | 00000000 | BRR LINK | | |

```

342.
343. 002212L
344.
345.
346.
347. 002212L 00110001 11001001
      002213L 00110001 11101000
      002214L 00110111 01000111
      002215L 11001111 11101001

348.
349.
350. 002216L
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362. 002216L 01010001 11101101
      002217L 01101111 10110000
      002220L 11001000 10001000
363. 002221L 01010001 00000110
      002222L 00110111 00000001
364. 002223L 11001111 10001000
365.
    
```

```

*
RETS:
. 2.40      MAR <= PC
           FETCHS

           DLDX    PC2MR,,SMR                    LOAD THE MAR FROM P.C. VALUE (NO IMAR)

           BRA     FETCHS

*
POP:
. 6.45      (    060)    POP                    HL IS POPPED FROM STACK
. 8.10      (062 060)    POP                    BC
. 8.10      (174 060)    POP                    DE
. 8.10      (022 060)    POP                    XA

           (176 060)    POP                    HL UNSPECIFIED
           (I0D 060)    POP                    MIXED REGS UNSPECIFIED

           IF IMP ZERO THEN IMP <= URL
           POPSTK (FETCH)                    RETURNING TO FETCH

           BRS     POPSTK,F0,IZ,FETCH,CC      THE IMP REG POINTS TO REG-PAIR ALREADY

           LDPI    LIMP,URL                    DEFAULT USES HL

           BRA     POPSTK

.
    
```

```

368.
369. 002224L 01011001 11111011
370.
371. 002225L
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394. 002225L 00110001 00110111
    002226L 01101111 10110001
395.
396.
397. 002227L 00110001 11011100
    002230L 01010101 11111011
    002231L 00110111 00000100
398. 002232L 01010001 00001111
    002233L 00110111 00000001
399. 002234L 00110001 10001111
    002235L 00110111 00001001
    002236L 00110001 10101111
400. 002237L 01010101 11000000
401. 002240L 01000000 10000000
402. 002241L 11000010 01010011
403. 002242L 01010001 10101011
    002243L 00110111 11000000
    002244L 01010001 11101111
    002245L 00110111 11100000
    002246L 00110111 01000111
404. 002247L 00110001 11011111
405. 002250L 11000100 01010111
    002251L 11000111 01010000
406. 002252L 00110010 00110110
    002253L 00110111 10001111
407. 002254L 00110111 00001000
    
```

```

*
MEMPF2 BPGX $ ** IF FROM WRONG EVEN/ODD PAGE PAIR **
.
MEMPFS:
. *****
. P.C. AT ? WHEN GET MEMPF, WILL ALWAYS ASSUME IT CORRECT
. HOPING TO POINT TO THE NEXT INSTRUCTION.
. *****
.
.
. MEMORY PARITY FAULT ANALYSIS
.
. IMP ← IMPL (HIDDEN)
. (IMP, IMP-1) ← MAR
. IF MARH IN BASED REGION (IF BASED, ADD IN BASING FACTOR)
. THEN IMP-1 ← IMP-1 + (SEBRLS)
. PSHSTK
. CASE STEK
. SP: TEMP ← SRPARST SECTOR TABLE PARITY ERROR
. SCDON
. WV: TEMP ← SRVIOLW WRITE VIOLATION (WAS PROTECTED)
. SCOK
. AV: TEMP ← SRVIOLA ACCESS VIOLATION (WAS NOT PRIV'D)
. SCOK
. MP: TEMP ← SRPAREM MEMORY PARITY FAULT
. SCDON
.
. LDRP TEMP1,STEK,CC GET AND SAVE STATUS WORD
.
. STR CMPF CLEAR THE FAULT (NOT NECESSARY)
. (THE NEXT READ/WRITE DOES IT)
. DOPIP MODW,ND,-1-SWUSER,PSWI ENABLE WRITING ACCESS PROTECTED MEMORY
.
. LDPI LIMP,IMPL USING HIDDEN REGS AND PSHSTK
.
. DLDX MR2IM,DIMP SAVE THE MAR ON THE STACK
.
. DOTI ,ND,0300
. TSTIT XR,0200 HIGH BIT ON, NEXT BIT OFF?
. BRA MEMNBAS,FZ NO, IN ANOTHER 16K CHUNK
. DLDPI MAR0,SEBRLS,,SMR IN BASED AREA, GET BASE REG VALUE
.
. LDTP IMPI SO ADDRESS MATCHES 5500 VALUE
. MWAIT ,MEMPO !!! HELP !!!
.
. DOPP IMPO,AC,MDR ADD IN BASING
.
MEMNBAS STB IIMP POINT TO LSB TO SAVE
    
```

| | | | | | | | |
|------|----------|----------|----------|---------|--------|-----------------------------|-------------------------------------|
| 408. | 002255L | 01010001 | 01010000 | BRC | PSHST0 | PUSH MAR VALUE ON THE STACK | |
| | 002256L | 11001111 | 11000111 | | | | |
| 409. | 002257L | 01010001 | 00000110 | MEMPO | LDTI | SRSYSMF | ASSUME THE WORST, GOT ANOTHER FAULT |
| 410. | 002260L | 11000100 | 01001111 | | MWAIT | ,MEMPX | !!! HELP !!! |
| | 002261L | 11000111 | 01000101 | | | | |
| 411. | 002262L | 01110001 | 11110001 | | TSTIR | ,STLSP,TEMP1 | SP: SECTOR TABLE PARITY FAULT |
| | 002263L | 01000101 | 10000000 | | | | |
| 412. | | | | * * * * | 5.55 * | * * * + 1.80 | IF BASED |
| 413. | 002264L | 11000010 | 01000011 | | BRA | MEMPS,FZ | |
| 414. | 002265L | 01000101 | 01000000 | | TSTIT | ,STLW | WV: WRITE VIOLATON FAULT |
| 415. | 002266L | 11000010 | 00111110 | | BRA | MEMPW,FZ | |
| 416. | 002267L | 01000101 | 00000001 | | TSTIT | ,STLA | AV: ACCESS VIOLATION FAULT |
| 417. | 002270L | 11000010 | 01000001 | | BRA | MEMPA,FZ | |
| 418. | 002271L | 01010001 | 00010010 | | LDTI | SRMEMPE | MPI: MEMORY PARITY FAULT |
| 419. | >002272L | 01011001 | 11111111 | MEMPX | BRAX | SCROML | THESE ERRORS GO TO ROM VECTORS! |
| | >002273L | 11001111 | 11111111 | | | | |
| 420. | | | | . | | | |
| 421. | 002274L | 01010001 | 00011011 | MEMPS | LDTI | SRSTPE | SECTOR, ROM VECTOR |
| 422. | 002275L | 11001111 | 01000101 | | BRA | MEMPX | |
| 423. | | | | . | | | |
| 424. | 002276L | 01010001 | 00011000 | MEMPA | LDRI | TEMPL,SVAVIOL | ACCESS, RAM VECTOR |
| | 002277L | 01101111 | 11110010 | | | | |
| 425. | 002300L | 11001111 | 00111100 | | BRA | MEMPR | |
| 426. | | | | . | | | |
| 427. | 002301L | 01010001 | 00010010 | MEMPW | LDRI | TEMPL,SVWVIOL | WRITE, RAM VECTOR |
| | 002302L | 01101111 | 11110010 | | | | |
| 428. | >002303L | 01011001 | 11111111 | MEMPR | BRAX | SCRAM | |
| | >002304L | 11001111 | 11111111 | | | | |
| 429. | | | | . | | | |


```

432.
433. 002305L
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456. 002305L 11001000 00100001
457. 002306L 01010001 00001111
    002307L 00110111 00000001
458. 002310L 01010001 00110101
    002311L 11001111 10001001
459. 002312L 00110001 11000110
    002313L 00110001 11100101
    002314L 00110111 00001000
460. 002315L 00110001 11011111
    002316L 00110111 00100001
461. 002317L 00110001 11010110
    002320L 01010010 00000010
    002321L 00110111 10000110
    002322L 00110001 11010101
    002323L 00110110 10000101
462. 002324L 00110111 00001001
463. 002325L 00110001 11011111
464. 002326L 11000100 00101001
    002327L 11000111 01101010
465. 002330L 00110111 00001100
466. 002331L 00110111 00100001
467. 002332L 01010001 00000000
    002333L 00110111 00000001
468. >002334L 01011001 11111111
    >002335L 11001111 11111111
    
```

```

*
STKS:
. ( 065) STKS          SAVE STACK CONTENTS
. 2.25 + C * 8.70      ** NO SERVICE INTERRUPTS **
. - 0.25 TO SERVICE (+ INSTRUCTION RE-START)
.
. (111 065) STKL       LOAD STACK CONTENTS
. (100 065) STKL       LOAD STACK CONTENTS UNSPECIFIED
. 4.25 + C * 8.50
. - 0.25 TO SERVICE (+ INSTRUCTION RE-START)
.
. 7.15 (022 065) SYM    TO XA, MOVE SYSTEM SAVE AREA
. 7.15 (062 065) SYM    TO BC
. 7.15 (174 065) SYM    TO DE (SWAP WITH REG PAIR)
. 7.15 (176 065) SYM    TO HL
.
. IF NOT IMP ZERO THEN STKL
. ELSE IMP <= IMPL
. POPSTK                (TOS TO HIDDEN)
. (HL, HL+1) <= (IMP, IMP-1)
. HL <= HL + 2
. IMP <= 0
. RIN16
.
. BRA STKL,F0,IZ        LOAD THE STACK (OR MOVE IT)
. LDPI LIMP,IMPL        POINT TO HIDDEN REGISTERS
.
. BRC POPST0           GET THE TOP STACK ENTRY (M. POINTER)
.
. DLDX HL2MR,,IIMP     POINT TO MEMORY AND CORRECT IMP
.
. LDPP MDW,IMPI        STORE THE LSR
.
. DADDP URO+UR,URI+UR,2 UPDATE THE HL POINTER (POPST DOES CC)
.
. STB DIMP
. LDTP IMPI            GET THE MSB
. MWAIT ,MEMPFS
.
. STB IMAR            STORE THE MSB
. LDPT MDW
. LOPI LIMP,0         RESET IMP FOR STKS OP-CODE
.
. BRAX RIN16          DO REPEATED INSTRUCTION
    
```

```

469.
470. 002336L
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489. 002336L 11011010 11111111
490. 002337L 00110001 11000110
    002340L 00110001 11100101
    002341L 00110111 01000111
491. 002342L 01010001 00001110
    002343L 00110111 00000001
492. 002344L 00010111 10110010
493. 002345L 00110001 11010110
    002346L 01010100 00000010
    002347L 00110111 10000110
494. 002350L 11000100 00010111
    002351L 11000111 00010000
495. 002352L 00110001 00110110
496. 002353L 00110111 00001101
    002354L 00110111 01000111
497. 002355L 00110111 10001111
    002356L 00110111 00001000
498. 002357L 00110001 11010101
    002360L 01010100 00000000
    002361L 00110111 10000101
499. 002362L 11000100 00001101
    002363L 11000111 01101010
500. 002364L 00110001 00110110
    002365L 00110111 10001111
501. 002366L 01010001 00000111
    002367L 11001111 11000111
502.
503. 002370L 01010001 00000001
    002371L 00110111 00000001
504. >002372L 01011001 11111111
    >002373L 11001111 11111111
505.
    
```

```

*
STKL
.
. (111 065) STKL LOAD STACK CONTENTS
. (100 065) STKL LOAD STACK CONTENTS UNSPECIFIED
. 4.25 + C * 8.50
.
. 7.15 (062 065) SYSM TO BC, MOVE SYSTEM SAVE AREA
. 7.15 (174 065) SYSM TO DE
. 7.15 (176 065) SYSM TO HL (SWAP WITH REGISTER PAIR)
. 7.15 (022 065) SYSM TO XA
.
. IF NOT IMP ODD THEN SYSM
. ELSE IMP <= IMPL
. (IMP=1, IMP) <= (HL=1, HL)
. PSHSTK
. HL <= HL - 2
. IMP <= 1 (MAKE IMP ODD)
. RIN16
.
BRA SYSM,F0,10 DO SYS AREA SWAP!
DLDX HL2MR,,SMR
.
LDPI LIMP,IMPH POINT TO HIDDEN SAVE MSB
.
CCLR
DOPIP URO+URL,SB,2,URI+URL DECREMENT L
.
MWAIT ,STKLPF IF ERROR BE SURE TO DECR. H
.
LDTP MDR GET MSB AND POINT FOR LSB
STB DMAR,SMR
.
LDPT IMPO,IIMP
.
STKLPF DOPIP URO+URH,SB,0,URI+URH DECREMENT H
.
MWAIT ,MEMPF$
.
LDPP IMPO,MDR GET LSB DATA
.
BRC PSHST0 PUT IT ON THE STACK
.
LDPI LIMP,1 ** MUST HAVE MWAIT FOLLOWING
SET IMPLICIT ODD!
.
BRAX RIN16 AND REPEAT IF NEEDED
.
    
```



```

549.
550. 002423L
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566. 002423L 00110001 00110001
    002424L 01000101 00000001
567. 002425L 11010010 11100100
568.
569. 002426L
570.
571.
572.
573.
574.
575.
576. 002426L 00110111 00001100
    002427L 00110111 00001100
577. 002430L 00110111 00001100
    002431L 00110111 01000111
578. 002432L 11001111 11101001
579.
580. 002433L
581.
582.
583.
584.
585.
586. 002433L 00110111 00001100
    002434L 00110111 01000111
587. 002435L 11001110 11111111
    002436L 11010100 11100001
    002437L 11000111 01101011
588. 002440L 00110001 00110110
589. 002441L 00110111 00001100
    002442L 00110111 01000111
590. 002443L 01101111 11110001
591. 002444L 11010100 11011011
    002445L 11000111 01101011
592. 002446L 00110001 00110110
    
```

```

*
JUMPCC:
. 5.65 ( 100) JCC.LSB.MSB          CONDITIONAL JUMP
. 2.90 IF NOT TAKEN
.
. ( 100) JFC.LSB.MSB          FALSE CARRY
. ( 110) JFZ.LSB.MSB          FALSE ZERO (.NE.)
. ( 120) JFS.LSB.MSB          FALSE SIGN (.GE.)
. ( 130) JFP.LSB.MSB          FALSE PARITY (EVEN)
. ( 140) JTC.LSB.MSB          TRUE CARRY
. ( 150) JTZ.LSB.MSB          TRUE ZERO (.EQ.)
. ( 160) JTS.LSB.MSB          TRUE SIGN (.LT.)
. ( 170) JTP.LSB.MSB          TRUE PARITY (ODD)
.
. IF CC THEN JUMP
. ELSE NOJ
.
TSTIP ,STUSCF,STATUS          TEST CC
.
BRA JUMP,FZ                    YES, IT'S MET, DO THE JUMP
*
NOJ:
. 2.55 ( 045) NOJ.LSB.MSB        3 BYTE NO-OP
.
. MAR <= MAR + 3
. FETCHS                        (SKIP OP-CODE LSB,MSB TO NEXT OP-CODE)
.
. STB IMAR,IMAR
.
. STB IMAR,SMR                    A LITTLE FASTER IN COMPETITION WITH DMA
.
. BRA FETCHS
*
JUMP:
. 5.20 ( 104) JMP.LSB.MSB        UNCONDITIONAL JUMP
.
. MAR <= (MAR+1, MAR+2)
. FETCHS
.
. STB IMAR,SMR                    GET THE LSB
.
MWAIT NOOP,MEMPF2
.
LDTP MDR
. STB IMAR,SMR                    GET THE MSB
.
LDRT TEMP1
MWAIT ,MEMPF2
.
LDPP MARQH,MDR                    GOT THE MSB
    
```

| | | | | | | |
|------|---------|----------|----------|------|-----------------|---------------|
| 593. | 002447L | 00110111 | 11100000 | | | |
| | 002450L | 01110001 | 11110001 | LDPR | MARDL,TEMP1,SMR | AND SAVED LSB |
| | 002451L | 00110111 | 11000000 | | | |
| | 002452L | 00110111 | 01000111 | | | |
| 594. | 002453L | 11001111 | 11101001 | BRA | FETCHS | GO TO IT! |
| 595. | | | | | | |

LOAD'S AND STORE'S

| | | | | | | | |
|-----|---------|----------|----------|------|------|------|---|
| 598 | | | | * | | | |
| 599 | 002454L | | | LD6: | | | |
| 600 | | | | . | | | |
| 601 | | | | 3.75 | (| 006) | LD.VVV |
| 602 | | | | | | | LOAD REG WITH IMMEDIATE VALUE |
| 603 | | | | | (| 006) | LA.VVV |
| 604 | | | | | (| 016) | LB.VVV |
| 605 | | | | | (| 026) | LC.VVV |
| 606 | | | | | (| 036) | LD.VVV |
| 607 | | | | | (| 046) | LE.VVV |
| 608 | | | | | (| 056) | LH.VVV |
| 609 | | | | | (| 066) | LL.VVV |
| 610 | | | | | (| 076) | LX.VVV |
| 611 | | | | | | | LOAD X-REG WITH IMMEDIATE VALUE |
| 612 | | | | | | | |
| 613 | | | | | | | |
| 614 | | | | | | | |
| 615 | | | | | | | |
| 616 | 002454L | 00110111 | 00001100 | | | | |
| | 002455L | 00110111 | 01000111 | | | | |
| 617 | 002456L | 11001110 | 11111111 | | | | |
| | 002457L | 11010100 | 11010000 | | | | |
| | 002460L | 11000111 | 01101011 | | | | |
| 618 | 002461L | 00110001 | 00110110 | | | | |
| | 002462L | 00110111 | 10001101 | | | | |
| 619 | 002463L | 11001111 | 11101011 | | | | |
| 620 | | | | | | | |
| 621 | 002464L | | | * | | | |
| 622 | | | | LD7: | | | |
| 623 | | | | . | | | |
| 624 | | | | 4.80 | (| 3D7) | LRM |
| 625 | | | | | | | LOAD BYTE FROM MEMORY INTO REG R |
| 626 | | | | 6.65 | (062 | 3D7) | LRM |
| 627 | | | | 6.65 | (174 | 3D7) | LRM |
| 628 | | | | 6.65 | (022 | 3D7) | LRM |
| 629 | | | | | | | LOAD BYTE FROM MEMORY USING REG PAIR BC |
| 630 | | | | | | | LOAD BYTE FROM MEMORY USING REG PAIR DE |
| 631 | | | | | | | LOAD BYTE FROM MEMORY USING REG PAIR XA |
| 632 | | | | | | | |
| 633 | | | | | | | |
| 634 | | | | | | | |
| 635 | | | | | | | |
| 636 | 002464L | 11011000 | 11001000 | | | | |
| 637 | 002465L | 01010001 | 00000110 | | | | |
| | 002466L | 00110111 | 00000001 | | | | |
| 638 | 002467L | 00110001 | 11001111 | | | | |
| | 002470L | 00110111 | 00001001 | | | | |
| | 002471L | 00110001 | 11101111 | | | | |
| | 002472L | 00110111 | 01000111 | | | | |
| 639 | 002473L | 11001110 | 11111111 | | | | |
| | 002474L | 11010100 | 11000011 | | | | |
| | 002475L | 11010111 | 11000001 | | | | |

```

*
LD6:
. 3.75 ( 006) LD.VVV          LOAD REG WITH IMMEDIATE VALUE
.      ( 006) LA.VVV          LOAD A-REG
.      ( 016) LB.VVV
.      ( 026) LC.VVV
.      ( 036) LD.VVV
.      ( 046) LE.VVV
.      ( 056) LH.VVV
.      ( 066) LL.VVV
.      ( 076) LX.VVV          LOAD X-REG WITH IMMEDIATE VALUE
.
.      MAR <= MAR + 1
.      I35 <= (MAR)
.      FETCHI
.
.      STB    IMAR,SMR          POINT TO THE DATA
.
.      MWAIT NOOP,MEMPF2
.
.      LDPP  I350,MDR          GET THE DATA
.
.      BRA   FETCHI           AND RETURN
*
LD7:
. 4.80 ( 3D7) LRM             LOAD BYTE FROM MEMORY INTO REG R
. 6.65 (062 3D7) LRM          LOAD BYTE FROM MEMORY USING REG PAIR BC
. 6.65 (174 3D7) LRM          LOAD BYTE FROM MEMORY USING REG PAIR DE
. 6.65 (022 3D7) LRM          LOAD BYTE FROM MEMORY USING REG PAIR XA
.
.      (176 3D7) LRM          USING HL UNSPECIFIED
.      (I0D 3D7) LRM          USING MIXED REGS UNSPECIFIED
.
.      IF IMP ZERO THEN IMP <= URL
.      I35 <= (IMP, IMP=1)
.      FETCH
.
.      BRA   LD7I,F0,I2       SPECIFIED IMP?
.      LDPI  LIMP,URL         NO, THEN USE HL AS REG PAIR
.
.      LD7I  OLDX  IM2MR,OIMP,SMR  POINT AND START MEMORY READ
.
.      MWAIT NOOP,$+1        PUT IN STANDARD 200 NS DELAY FROM READY
    
```

| | | | | |
|------|---------------------------|------|-------------|-------------------------------------|
| 640. | 002476L 00110001 00110110 | LDPP | I350,MDR | GET THE DATA FIRST, THEN |
| | 002477L 00110111 10001101 | | | |
| 641. | 002500L 11000110 11101101 | BRA | FETCH,F0,MP | NO FAULT AFTER READ, THEN DONE |
| 642. | 002501L 11001111 01101011 | BRA | MEMPF2 | ELSE, GO TO FAULT ROUTINE |
| 643. | | | | |
| 644. | 002502L | | | |
| 645. | | | | |
| 646. | | | | |
| 647. | | | | |
| 648. | | | | |
| 649. | | | | |
| 650. | | | | |
| 651. | | | | |
| 652. | | | | |
| 653. | | | | |
| 654. | | | | |
| 655. | | | | |
| 656. | | | | |
| 657. | | | | |
| 658. | 002502L 11011000 10111010 | BRA | L7SI,F0,IZ | ALMOST EXACTLY LIKE LD7 |
| 659. | 002503L 01010001 00000110 | LDPI | LIMP,URL | |
| | 002504L 00110111 00000001 | | | |
| 660. | 002505L 00110001 11001111 | L7SI | OLDX | IM2MR,DIMP |
| | 002506L 00110111 00001001 | | | |
| | 002507L 00110001 11101111 | | | |
| 661. | 002510L 00110001 11011110 | LDPP | MDW,I02I | WRITE REGISTER IN MEMORY |
| | 002511L 00110111 00100001 | | | |
| 662. | 002512L 11001111 11101111 | BRA | FETCHW | |
| 663. | | | | |
| 664. | 002513L | | | |
| 665. | | | | |
| 666. | | | | |
| 667. | | | | |
| 668. | | | | |
| 669. | | | | |
| 670. | | | | |
| 671. | 002513L 00110001 11011110 | LDPP | I350,I02I | SAVE SOURCE REGISTER IN DESTINATION |
| | 002514L 00110111 10001101 | | | |
| 672. | 002515L 11001111 11101011 | BRA | FETCHI | AND DONE! |
| 673. | | | | |

*
L7S:

. 4.45 (37S) LMR STORE A BYTE IN MEMORY
 . 6.30 (062 37S) LMR USE REG PAIR BC
 . 6.30 (174 37S) LMR USE REG PAIR DE
 . 6.30 (022 37S) LMR USE REG PAIR XA
 .
 . (176 37S) LMR USE REG PAIR HL UNSPECIFIED
 . (I0D 37S) LMR USING MIXED REGS UNSPECIFIED

. IF IMP ZERO THEN IMP <= URL
 . (IMP, IMP-1) <= I02
 .

*
LDS:

. 2.55 (3DS) LRR LOAD SOURCE REGISTER INTO ANOTHER REG
 . (IMP 3DS) UNSPECIFIED

. I35 <= I02

. ARITHMETIC LOGICAL UNIT OPERATIONS

| | | | | | | |
|-----|---------|----------|----------|-------------------------|--|--|
| 676 | | | | * | | |
| 677 | 002516L | | | AP4: | | |
| 678 | | | | . 5,00 (0P4) OP,VVV | ALUOP ON IMMEDIATE VALUE | |
| 679 | | | | . 7,00 (IMP 0P4) OP,VVV | ALUOP IS OTHER THAN TO A-REG. | |
| 680 | | | | . +.45 IF 'AC' OR 'SB' | | |
| 681 | | | | . | | |
| 682 | | | | . (004) AD,VVV | ADD IMMEDIATE VALUE TO A-REG | |
| 683 | | | | . (IMP 004) AD,VVV | ADD IMMEDIATE VALUE TO IMP-REG | |
| 684 | | | | . (014) AC,VVV | ADD WITH CARRY IMM. TO A-REG | |
| 685 | | | | . (IMP 014) AC,VVV | ADD WITH CARRY IMM. TO IMP-REG | |
| 686 | | | | . (024) SU,VVV | SUB IMMEDIATE VALUE FROM A-REG | |
| 687 | | | | . (IMP 024) SU,VVV | SUB IMMEDIATE VALUE FROM IMP-REG | |
| 688 | | | | . (034) SB,VVV | SUB WITH CARRY IMM. FROM A-REG | |
| 689 | | | | . (IMP 034) SB,VVV | SUB WITH CARRY IMM. FROM IMP-REG | |
| 690 | | | | . (044) ND,VVV | AND IMMEDIATE VALUE TO A-REG | |
| 691 | | | | . (IMP 044) ND,VVV | AND IMMEDIATE VALUE TO IMP-REG | |
| 692 | | | | . (054) XR,VVV | XOR IMMEDIATE VALUE TO A-REG | |
| 693 | | | | . (IMP 054) XR,VVV | XOR IMMEDIATE VALUE TO IMP-REG | |
| 694 | | | | . (064) OR,VVV | IOR IMMEDIATE VALUE TO A-REG | |
| 695 | | | | . (IMP 064) OR,VVV | IOR IMMEDIATE VALUE TO IMP-REG | |
| 696 | | | | . (074) CP,VVV | SUB IMM. VALUE FROM A-REG TO SET FLAGS | |
| 697 | | | | . (IMP 074) CP,VVV | SUB IMM. VALUE FROM IMP-REG TO SET FLAGS | |
| 698 | | | | . | | |
| 699 | | | | . MAR <= MAR + 1 | | |
| 700 | | | | . TEMP2 <= T | THE OP-CODE (MIDDLE 3 BITS IMPORTANT) | |
| 701 | | | | . T <= (MAR) | THE DATA | |
| 702 | | | | . ALUOP | | |
| 703 | | | | . | | |
| 704 | 002516L | 00110111 | 00001100 | . STB IMAR,SMR | GET FOLLOWING IMMEDIATE DATA | |
| 705 | 002517L | 00110111 | 01000111 | | | |
| 706 | 002520L | 01101111 | 11110010 | . LDRT TEMP2 | SAVE ALU OP-CODE | |
| 707 | 002521L | 11010100 | 10101110 | . MWAIT ,MEMPF2 | | |
| 708 | 002522L | 11000111 | 01101011 | | | |
| 709 | 002523L | 00110001 | 00110110 | . LDTP MDR | THE DATA IS IN T | |
| 710 | 002524L | 11011111 | 10011101 | . BRA ALUOP | | |
| 711 | 002525L | | | * | | |
| 712 | | | | AP7: | | |
| 713 | | | | . 5,55 (2P7) OPM | ALUOP WITH MEMORY DATA | |
| 714 | | | | . 7,55 (IMP 2P7) OPM | DEST. OTHER THAN A-REG | |
| 715 | | | | . +.45 IF 'AC' OR 'SB' | | |
| 716 | | | | . | | |
| 717 | | | | . TEMP2 <= T | THE OP-CODE (MIDDLE 3 BITS IMPORTANT) | |
| 718 | | | | . T <= (HL) | THE DATA | |
| 719 | | | | . ALUOP | | |
| 720 | 002525L | 01101111 | 11110010 | . LDRT TEMP2 | SAVE ALU OP-CODE | |
| 721 | 002526L | 00110001 | 11000110 | . DLDX HL2MR,,SMR | GET THE DATA | |
| | 002527L | 00110001 | 11100101 | | | |
| | 002530L | 00110111 | 01000111 | | | |
| | 002531L | 11001110 | 11111111 | . MWAIT NOOP,MEMPF2 | | |
| | 002532L | 11010100 | 10100101 | | | |
| | 002533L | 11000111 | 01101011 | | | |

722. 002534L 00110001 11001001
002535L 00110001 11101000
723. 002536L 00110001 00110110
724. 002537L 11011111 10011101

DLDX PC2MR
LDTP MDR
BRA ALUOP

SO ALUOP CAN USE FETCHI RETURN
THE DATA
AND GO WORK WITH IT

```

725,
726, 002540L
727,
728,
729,
730,
731,
732,
733,
734,
735, 002540L 01101111 11110010
736, 002541L 00110001 11011110
737,
738,
739, 002542L
740,
741,
742,
743,
744,
745,
746,
747,
748,
749,
750,
751,
752,
753,
754,
755,
756,
757,
758,
759,
760,
761,
762,
763,
764,
765, 002542L 01101111 11110001
766, 002543L 01110001 11110010
767, 002544L 00010111 10010010
768, 002545L 01010101 00011100
769, 002546L 01010010 01111010
770, 002547L 01101111 10110000
771, 002550L 11101111 00000000
    
```

```

+
APS:
. 3,70 ( 2PS) OPR          ALUOP WITH SOURCE REGISTER
. 5,70 (IMP 2PS) OPR       DEST. OTHER THAN A-REG
. +.45 IF 'AC' OR 'SB'
.
.           TEMP2 <= T      THE OP-CODE (MIDDLE 3 BITS IMPORTANT)
.           T <= I02        THE DATA
.           ALUOP
.
.           LDRT  TEMP2     SAVE ALU OP-CODE
.           LDTP  I02I      SELECT SOURCE REG.
.           BRA   ALUOP
.
ALUOP
. 3,45
. +.45 IF 'AC' OR 'SB'
.
. SOURCE IN T-REG AND DO ARITH WITH THE IMPLIED REGISTER
.
.           TEMP1 <= T      SAVE THE DATA
.           CCLR          CLEAR THE CARRY
.           CASE  TEMP2     (OPCODE SELECT)
.           7: LUF <= IMP = TEMP1
.              FETCHI
.           6: IMPF <= IMP ,IOR, TEMP1
.              FETCHI
.           5: IMPF <= IMP ,XOR, TEMP1
.              FETCHI
.           4: IMPF <= IMP ,AND, TEMP1
.              FETCHI
.           3: CARRY <= UCFLG
.              2:
.           2: IMPF <= IMP = TEMP1
.              FETCHI
.           1: CARRY <= UCFLG
.              0:
.           0: IMPF <= IMP + TEMP1
.              FETCHI
.
.           LDRT  TEMP1     SAVE SOURCE REG.
.           LDTR  TEMP2     GET OPCODE
.           SHIFT SR,CC     GET ALUOP IN BITS 4-2
.           DOTI  ,ND,034   SELECT 3 BITS
.           DOTA  ,AC,ALUXQT OFFSET TO THE EXEC TABLE
.           BAS   LINK,CC   CLEAR CARRY AND
.           BRR   LINK      GO TO CORRECT ROUTINE
    
```

```

772.
773.
774.
775.
776.
777. 002551L 00110001 11011111
      002552L 01110100 00110001
      002553L 00110111 00000110
778. 002554L 11001111 11101011
779.
780.
781.
782. 002555L 00110001 11011111
      002556L 01110011 10110001
      002557L 00110111 01101111
783. 002560L 11001111 11101011
784.
785.
786.
787. 002561L 00110001 11011111
      002562L 01110000 10110001
      002563L 00110111 01101111
788. 002564L 11001111 11101011
789.
790.
791.
792. 002565L 00110001 11011111
      002566L 01110101 10110001
      002567L 00110111 01101111
793. 002570L 11001111 11101011
794.
795.
796.
797. 002571L 00110001 00110101
      002572L 01101111 11110010
798. 002573L 01110010 00110010
799. 002574L 11001110 11111111
800.
801.
802.
803. 002575L 00110001 11011111
      002576L 01110100 11110001
      002577L 00110111 01101111
804. 002600L 11001111 11101011
805.
806.
807.
808. 002601L 00110001 00110101
      002602L 01101111 11110010
809. 002603L 01110010 00110010
810. 002604L 11001110 11111111
811.
    
```

```

*
. THE TABLE STARTS HERE AND IS BACKWARDS
.
. ( A7S) CP 2'S COMPLEMENT SUBTRACT TO SET FLAGS
.
. DOPRP LUF,SB,TEMP1,IMPI,C0
.
. BRA FETCHI
.
. ( A6S) OR LOGICAL INCLUSIVE OR
.
. DOPRP IMPFO,OR,TEMP1,IMPI,CC
.
. BRA FETCHI
.
. ( A5S) XR LOGICAL EXCLUSIVE OR
.
. DOPRP IMPFO,XR,TEMP1,IMPI,CC
.
. BRA FETCHI
.
. ( A4S) ND LOGICAL AND
.
. DOPRP IMPFO,ND,TEMP1,IMPI,CC
.
. BRA FETCHI
.
. ( A3S) SB 2'S COMPLEMENT SUBTRACT WITH CARRY
.
. LDRP TEMP2,UCFLG
.
. DOTR ,AC,TEMP2,,C0 SET THE CARRY
. NOOP
.
. ( A2S) SU 2'S COMPLEMENT SUBTRACT
.
. DOPRP IMPFO,SB,TEMP1,IMPI
.
. BRA FETCHI
.
. ( A1S) AC 2'S COMPLEMENT ADD WITH CARRY
.
. LDRP TEMP2,UCFLG
.
. DOTR ,AC,TEMP2,,C0 SET THE CARRY
. NOOP
    
```

• ARITHMETIC LOGICAL UNIT OPERATIONS

812.
813.
814.
815.
816.
817.

002605L
002605L 00110001 11011111
002606L 01110010 11110001
002607L 00110111 01101111
002610L 11001111 11101011

• (A0S) AD 2'S COMPLEMENT ADD
•
ALUXQT
DOPRP IMPFO,AC,TEMP1,IMPI
BRA FETCHI
•

```

820,
821, 002611L
822,
823,
824,
825,
826,
827, 002611L 00110001 11011100
      002612L 01010011 00100000
      002613L 00110111 10001100
828, 002614L 11010100 01110011
      002615L 11000111 01101011
829,
830, 002616L
831,
832,
833,
834,
835,
836,
837,
838, 002616L 00110001 00110000
839, 002617L 11010011 01100011
840,
841, 002620L
842,
843,
844,
845,
846,
847,
848,
849,
850,
851,
852,
853,
854,
855,
856,
857,
858,
859,
860,
861,
862,
863,
864,
865,
866,
867, 002620L 01000101 00010000
868, 002621L 11010010 00001111

```

```

*
SRVRPT:
.
. REPEATED SERVICE REQUESTS
.
. PSW <= PSW ,OR, SWRPT
. SRVNXT
.
. DOPIP PSWO,OR,SWRPT,PSWI SET REPEAT FLAG
.
. MWAIT ,MEMPF2
.
*
SRVNXT:
.
. FOLLOWING TIMING ASSUMED OR CORRECTED TILL THIS POINT REACHED
. (NOT INCLUDING OFF-PAGE ROUTINES)
.
. T <= SRVREG
. IF ZERO SRVRTW
.
. TSTTP FI,SRVREQ RELOAD SERVICE REQUEST FLAGS
. BRA SRVRTW,TZ (SPEEDUP, REALLY NO MORE INTERRUPTS)
*
SRVDO:
.
. SERVICE LOOP - PRIORITY SEQUENCE
. *****
. NOTICE: ALL SERVICE ROUTINES THAT USE THE MAR MUST FIRST HAVE A
. MWAIT ,S+1 SO THAT THE MEMORY BUS IS NOT OVERRUN.
. ALSO, THE MODE-WORD SWUSER MODE BIT MAY NEED TO BE TURNED OFF
. TO ALLOW ACCESS TO PROTECTED MEMORY
. *****
. NOTE: P.C. POINTS TO START OF INSTRUCTION BEING EXECUTED (REPEATED)
. OR TO INST. THAT WAS TO BE EXECUTED (FETCH)
. NOT INCLUDING THE IMP-SPEC.
. *****
.
. SERVICE REQUESTS
.
. CASE SVCREQ
.
. DSPNL: DLDO MEMORY FAULT (IGNORED THROUGH HERE)
. SDLCR: SDOR DISPLAY, KEYBOARD SERVICE
. SDLCT: SDOT SDLC RECEIVE (EXCEPT 3800)
. ONMS: OMDO SDLC TRANSMIT (EXCEPT 3800)
. MBUS: BSDO ONE MILLISECOND INTERRUPT
. HUMS: HUMDO MICRO-I/F BUS INTERRUPT (EXCEPT 3800)
. : UNUSED
. : SRVRTW ELSE RETURN THROUGH MEMORY WAIT
. : FOR THE WASTED I-FETCH
.
. TSTIT ,SCDSPNL
. BRA DLDO,FZ DISPLAY-KEYBOARD SERVICE

```

| | IFS | TYPE | |
|-----|---------------------------|-----------------|---------------------------------------|
| 869 | | | |
| 870 | 002622L 01000101 00000100 | TSTIT ,SCSOLCR | (IF 3800, DO NOT) |
| 871 | 002623L 11010010 01011001 | BRA COMRDO,FZ | COMM RECEIVE |
| 872 | 002624L 01000101 00001000 | TSTIT ,SCSOLCT | (IF 3800, DO NOT) |
| 873 | 002625L 11010010 01010100 | BRA COMTDO,FZ | COMM TRANSMIT |
| 874 | | XIF | |
| 875 | 002626L 01000101 00100000 | TSTIT ,SCONMS | |
| 876 | 002627L 11010010 00001101 | BRA OMDO,FZ | ONE MILLI-SECOND INT |
| 877 | | IFS | |
| 878 | 002630L 01000101 00000010 | TSTIT ,SCMBUS | (IF 3800, DO NOT) |
| 879 | 002631L 11010010 00111101 | BRA BSDD,FZ | BUS INTERRUPT |
| 880 | | XIF | |
| 881 | 002632L 01000101 01000000 | TSTIT ,SCHUMS | |
| 882 | 002633L 11010010 01001101 | BRA HUMDO,FZ | HUNDRED MICRO-SEC INT |
| 883 | | | |
| 884 | 002634L | | |
| 885 | | | |
| 886 | | | |
| 887 | | | |
| 888 | | | |
| 889 | | | |
| 890 | | | |
| 891 | 002634L 11010100 01100011 | MWAIT ,MEMPF2 | FOR TRAILING MDW'S |
| | 002635L 11000111 01101011 | | |
| 892 | 002636L 00110001 11011100 | LDPP MODW,PSWI | RESTORE CORRECT MODE (IF CHANGED) |
| | 002637L 00110111 00000100 | | |
| 893 | 002640L 01000101 00100000 | TSTIT ,SWRPT | REPEATED INSTRUCTION? |
| 894 | 002641L 11000010 11011111 | BRA SRVID,FZ | YES, DO IT ONLY |
| 895 | | | WARNING! SOME REPEATED CODE IN FUTURE |
| 896 | | | MAY NEED MAR TO MATCH THE PC |
| 897 | 002642L 00110001 11001001 | DLDX PC2MR,,SMR | NO, CODE AS AT FETCH WITHOUT SERVICE |
| | 002643L 00110001 11101000 | | |
| | 002644L 00110111 01000111 | | |
| 898 | 002645L 11001111 11100011 | BRA SRVEND | GO TO FETCH-EXECUTE |

*
 SRVRTW:

.
 . MODW ← PSW
 . IF SWRPT THEN IDCOD (IREG)
 . MAR ← PC
 . SRVEND
 .

957,
 958,
 959, 002663L 00010001 10111111
 960, 002664L 11010011 01110001
 961, 002665L 01011001 11111000
 002666L 11011111 11111111

XIF
 IFS APF
 LDTR ACCTL,CC IF ZERO
 BRA SRVNXT,TZ THE CHANNEL IS OFF
 BRAX ACSDO ELSE ON, SO DO IT

*
 . SPECIAL AML INTERRUPT CODE

962,
 963,
 964,
 965, 002667L
 966, 002667L 01000000 00001101
 967, 002670L 11010010 00110000
 968, 002671L 01010001 00001101
 002672L 11011100 01000101
 002673L 00110111 00100110
 002674L 01010001 00000000
 002675L 00110111 00100111
 002676L 00110111 00101001
 969, 002677L 01010001 01010100
 002700L 01101111 11110010
 002701L 11011111 00001010

. AMLNT
 TSTIT XR,AMLADR
 BRA APFNT,FZ
 MBUSC ,AMLADR,MBOFF DISABLE AML INTERRUPTS

LDRI TEMPL,SVAMLI AND USE ITS INTERRUPT VECTOR
 BRA AMLINT TO DO AML INTERRUPT
 XIF

970,
 971,
 972,
 973,
 974, 002702L

*
 BSDO IFS TYPE MICRO-BUS INTERRUPT
 . * * * 2.70 * * * DISK SERVICE
 . 4.20 APF SERVICE (4.30 RECEIVE)
 . 4.85 NO SERVICE

975,
 976,
 977,
 978,
 979,
 980,
 981,
 982,
 983,
 984,
 985,
 986,
 987,
 988,
 989,
 990,

MIFIAK
 TEMP1 <= MIFIN ACKNOWLEDGE AND GET INTERRUPT NUM/ADDR
 IF MBSTAT ZERO THEN BSDON
 T <= TEMP1 ,XOR, MADR FIADR BITS ZERO IF ADDRESS MATCHES
 IF (T .AND. FIADR) ZERO THEN MBPAGE

. BSDON:
 TEMP1 <= TEMP1 .AND. FIADR SELECT ADDRESS BITS
 MBUS (TEMP1, FCOINT, 0) CLEAR INTERRUPT MASK
 MBUS (TEMP1, FCLEAR, FKMAST-FKRWMF) AND MOST INTERRUPTS
 SRVBSN

991, 002702L 00110111 00101000
 992, 002703L 11001110 11111111
 002704L 11011100 00111011
 002705L 00110001 00010101
 993, 002706L 01101111 11110010
 994, 002707L 01010101 00001111
 995, 002710L 00010000 11000011

MBWAIT HERE FOR SAFETY
 STB MIFIAK ACKNOWLEDGE THE INTERRUPT (700NSEC IAK)
 MBIN NOP WAIT ON ACK-DATA

LDRT TEMP2 SAVE DATA AWAY FOR WHOEVER NEEDS IT
 DOTI ,ND,FIADR SELECT THE ADDRESS BITS TO BE CHECKED
 TSTRT XR,MADR WAS THE ADDRESS OK?

996,
 997,
 998, 002711L 11010010 01001000
 999, 002712L 00010001 11000101

IFS APF
 BRA AMLNT,FZ NOT CORRECT, NOT DISKETTE
 TSTRT FI,MBSTAT NON-ZERO IF DISKETTE CODE ACTIVE

1000, 002713L 11010011 01001000
 1001,
 1002,
 1003,
 1004,
 1008,
 1009,
 1010,
 1011, 002714L 01110001 11110010
 1012, >002715L 01011001 11111111
 >002716L 11001111 11111111
 1013,
 1014.

BRA AMLNT,TZ NOT ACTIVE, NOT DISKETTE CODE
 XIF

IFS TYPE
 IFC APF
 XIF

IFS TYPE
 LDTR TEMP2
 BRAX MBPAGE

LOAD HIGH BITS FOR INTERRUPT CODE
 OK, DO MICRO-DISKETTE INTERRUPT

IFS APF

```

1017
1018
1019
1020
1021
1022
1023
1024
1025 002717L
1026 002717L 01000000 00001110
1027 002720L 11010010 00100000
1028
1029
1030
1031 002721L 11010100 00101110
      002722L 11000111 01101011
1032 002723L 00110001 11011100
      002724L 01010101 11111011
      002725L 00110111 00000100
1033 002726L 01010001 01011010
      002727L 00110111 11000000
      002730L 01010001 11101111
      002731L 00110111 11100000
1034 002732L 01011001 11111000
1035 002733L 01110001 11110010
      002734L 01000101 10000000
1036 002735L 11010010 01001001
1037 002736L 11011111 01111111
1038
1039
1040
1041
1042 002737L
1043
1044
1045
1046
1047
1048
1049
1050
1051 002737L 01101111 11110010
1052 002740L 01010011 10110000
      002741L 11011100 00011110
      002742L 00110111 00100110
      002743L 01010001 00000000
      002744L 00110111 00100111
      002745L 00110111 00101001
1053 002746L 01110001 11110010
      002747L 01010011 00110000
      002750L 11011100 00010111
      002751L 00110111 00100110
    
```

```

*
*APFNT:
*   INTERRUPT APFNT                                INTERRUPT INTERFACE
*   BEGIN
*   IAkode := MBIN (IAK);                          GET INTERRUPT IAK CODE
*   T := IAkode .XOR. MTADR;                       SET BITS OFF IF CORRECT DEV
*   IF (T .AND. FIADR) = ZERO                      IS IT THE EXPECTED ADDRESS
*   THEN                                           YES, GO TO 1 OF 4 SERVICES
*
*APFNT                                           ** ADDRESS HERE ***
*   TSTIT XR,MTADR
*   BRA   BSDON,FZ                                ** CONTINUE FROM WHERE LEFT OFF **
*
*   EXIT (CASE T OF
*         (IRNXT, IREOT, ITNXT, ITBOT))
*
*   MWAIT ,MEMPF2                                ** ITRPT SPECIAL CASE OF ITBOT **
*                                               BYPASS WASTED IFETCH
*
*   DOPIP MODW,ND,-1=SWUSER,PSWI TEMPORARILY DISABLE USER MODE
*
*   DLDPI MARO,MTLEN                             (SPACE HERE & COMMON CODE)
*
*   BPGX  IRNXT
*   TSTIR ,MIRNXT.XOR.MITNXT,TEMP2 *OTHER METHODS OF BRANCHING*
*
*   BRA   ITINT,FZ                               *ARE JUST AS POSSIBLE*
*   BRA   IRINT
*   XIF
*   IFS   TYPE
*
*   ELSE BEGIN
*
*   BSDON                                         *REGULAR BSDON CODE HERE*
*   MBOUT (IAKODE .AND. FIADR, 0260, 0);
*
*   MBOUT (IAKODE .AND. FIADR, 060, 037)        LOAD DISKETTE INTERRUPT MASK
*
*   END;                                         CLEAR DISKETTE INTERRUPTS
*
*   RETURN
*   END APFNT;
*
*   LDRT  TEMP2                                  SAVE THE ADDRESS (ONLY)
*   MBUS  ,FCOINT,0                              NO MORE INTERRUPTS FROM IT
*
*   MBUS  TEMP2,FCLEAR,FKMAST=FKRWMF           AND CLEAR MOST INTERRUPTS
    
```


APF INTERFACE INTERRUPT ROUTINES

1098.
1099.
1100.
1101.

002774L 11111111 11111111
002775L 11111111 11111111
002776L 11111111 11111111
002777L 11111111 11111111

- IF IMP NON-ZERO MUST BACK UP PC TO RE-DO IMP SPEC CODE WHEN MACRO-RETURNS
- COULD SIMPLIFY, IF IMP-NON-ZERO MUST BACKUP BECAUSE IS IMP-SPEC REPEATED

TABPAGE PRODL

```

1104.
1105. 003000L
1106.
1107.
1108. 003000L 01010001 11110111
      003001L 01101111 11110000
1109. 003002L 11000100 11111101
      003003L 11000111 11001101
1110. 003004L 00110001 11011100
      003005L 01010101 11111011
      003006L 00110111 00000100
      003007L 11001111 11001011
1111.
1112.
1113. 003010L 00110001 00110001
      003011L 01000010 10000000
1114. 003012L 00010001 11110000
1115. 003013L 11000000 11100101
1116.
1117. 003014L 11000100 11110011
      003015L 11000111 11001101
1118. 003016L 01010001 01101011
      003017L 00110111 11000000
      003020L 00110111 01000111
1119. 003021L 01010001 00000000
1120. 003022L 11000100 11101101
      003023L 11000111 11001101
1121. 003024L 00110010 00110110
      003025L 00110111 00100001
1122.
1123. 003026L 00010001 10111111
1124. 003027L 11000011 11100110
1125. 003030L 00000110 01111111
1126. 003031L
1127.
1128.
1130.
1131. 003031L 01010001 10011100
1132.
1133. 003032L 11000100 11100101
      003033L 11000111 11001101
1134. 003034L 00110111 11000000
      003035L 00110111 01000111
1135. 003036L 01010100 00000010
      003037L 00000111 11110000
1136. 003040L 11000100 11011111
      003041L 11000111 11001101
1137. 003042L 00110001 00110110
1138. 003043L 00110111 00001100
      003044L 00110111 01000111
1139. 003045L 01101111 11110001
1140. 003046L 01010001 11001111
      003047L 01101111 11110000
    
```

```

*
DLSDO:
* * * * 4.60 * * * * + 2.15 IF BOTTOM LINE
*
      BAL      LINK,DLNL          SET UP RETURN FROM KEY SERVICE
      MWAIT   ,MEMPFDL          WAIT ON WAISTED IFETCH
      DOPIP   MDW,ND,-1-SWUSER,PSWI ** DISABLE USER MODE FOR READS/Writes **
      BRA     KBD$DO           GO DO FIRST KEYBOARD SERVICE
      DLNL    TSTIP AC,STBOTLN,STATUS SET CARRY IF BOTTOM LINE STATUS
      LDTR    PDLNP
      BRA     DLNXT,FC          NOT YET
      MWAIT   ,MEMPFDL          WAIT ON KBD STATUS BIT WRITE
      LDPI    MAROL,SELFREQ,SMR  GET FREQ COUNTER, SAME PAGE AS KBD STUFF
      TCLR
      MWAIT   ,MEMPFDL          SET T-REG ZERO (NOTE: CARRY IS TRUE!!)
      DOPP    MDW,AC,MDR        ADD ONE AND RESTORE IT
      IFS     APF
      LDTR    ACCTL,CC
      BRA     DLACNINC,TZ      DO NOT INCREMENT IF CHANNEL OFF
      INCR    ACCTL
      DLACNINC
      XIF
      IFC     APF
      XIF
      LDTI    SEDLBT          POINT TO BOTTOM LINE FOR SCAN
      DLNXT   MWAIT ,MEMPFDL
      LDPT    MAROL,SMR        SET LSB OF ADDRESS AND READ ADDRESS LSB
      DORI    PDLNP,SB,2      SUBTRACT 2 FROM LIST POINTER
      MWAIT   ,MEMPFDL
      LDTP    MDR              GET LSB OF DATA POINTER
      STB     IMAR,SMR        GET MSB ALSO
      LDRT    TEMP1           HOLD LSB FOR THE MOMENT
      BAL     LINK,DLNLR      SET UP RETURN ADDRESS (VERY EARLY!)
    
```

| | | | | | | |
|-------|---------|----------|----------|---------|-------------------|--|
| 1141. | 003050L | 11000100 | 11010111 | MWAIT | ,MEMPFDL | |
| | 003051L | 11000111 | 11001101 | | | |
| 1142. | 003052L | 00110001 | 00110110 | LDPP | MAR0H,MDR | GET MSB POINTER |
| | 003053L | 00110111 | 11100000 | | | |
| 1143. | 003054L | 01110001 | 11110001 | LDPR | MAR0L,TEMP1,LDMAP | AND LSB, AND STROBE TO LOAD THE DMA PNTR |
| | 003055L | 00110111 | 11000000 | | | |
| | 003056L | 00110111 | 01000001 | | | |
| 1144. | 003057L | 11001111 | 11001011 | BRA | KBDS00 | GO DO THE SECOND KEYBOARD SCAN |
| 1145. | | | | | | |
| 1146. | 003060L | 01011001 | 11111010 | DLNLR | BRAX | SRVNXT |
| | 003061L | 11011111 | 01110001 | | | |
| 1147. | | | | | | |
| 1148. | 003062L | 00110111 | 01000001 | MEMPFDL | STB | LDMAP |
| 1149. | 003063L | 11001111 | 00100100 | BRA | MEMPF3 | CLEAR INTERRUPT ON ANY MEMORY FAULT |

```

1150,
1151, 003064L
1152,
1153,
1154,
1155,
1156,
1157,
1158,
1159,
1160,
1161,
1162,
1163,
1164,
1165,
1166,
1167,
1168,
1169,
1170,
1171,
1172,
1173,
1174, 003064L 01010001 01101001
003065L 00110111 11000000
003066L 01010001 11101111
003067L 00110111 11100000
1175, 003070L 00110001 00110001
003071L 01000101 01000000
1176, 003072L 11000011 11000111
1177,
1178, 003073L 00010001 10110001
003074L 01000100 00010000
003075L 11000000 10011111
1179,
1180,
1181, 003076L 00010010 00110001
1182, 003077L 01010010 10011111
1183, 003100L 01101111 11110001
1184, 003101L 11101111 00000001
1185,
1186,
1187, 003102L 01010001 00000010
1188, 003103L 11001111 10000000
1189, 003104L 01010001 00000001
1190, 003105L 11001111 10000000
1191, 003106L 01010001 00000010
1192, 003107L 11001111 10000001
1193, 003110L 01010001 00001000
1194, 003111L 11001111 10000001
1195, 003112L 01010001 10000000
1196, 003113L 11001111 01110100
    
```

```

+
KBD$DO
. * * * 5.60 * * * NEW KEY JUST WENT DOWN
. 2.95 THE KEY IS UP AND WAS NOT LAST TO GO DOWN
. 4.85 REPEATED KEY UP
. 4.65 REPEATED KEY DOWN
. + 0.70 IF <SPACE>, <CANCEL>, <.,>, <BACKSPACE>
. OR + 0.50 IF <ENTER>
.
. * * * 4.20 * * * KBKS1 STATUS BIT (UP OR DOWN)
. OR 4.35 KBKS2 STATUS BIT
. OR 4.35 KBREINT STATUS BIT DOWN
. OR 5.50 KBREINT STATUS BIT UP
. OR 5.50 RESTART = IMP ZERO TO SCROM
. OR 5.85 RESTART = NON-RPT'D INST. TO SCROM
. OR 6.70 RESTART = RPT'D & IMP NON-ZERO TO SCROM
.
. SCAN THE KEYBOARD BY:
. 1) WAIT FOR THE LAST SCAN TO COMPLETE
. 2) IF SPECIAL KEY, DO IT'S TYPE OF SERVICE (SET OR CLEAR STATUS BIT)
. 3) IF NEW CLOSURE GET KEYCODE, SAVE IT, AND SAVE THE SCAN NUMBER
. 4) IF NOT THE SAME KEY AS SPECIFIED BY LAST SCAN NUMBER THEN DONE
. 5) IF SAME, AND STILL DOWN, SET STATUS BIT ELSE, CLEAR IT
.
. DLDPI MARD,SEKBS1 MIDDLE OF 3 BYTES THAT WILL WORK WITH
.
KBDWAIT TSTIP ,STKBRDY,STATUS
.
BRA KBDWAIT,TZ (LOOPS ONCE ON SECOND SERVICE, MAYBE)
(NOT READY TILL 5.3 MICRO-S FROM KBSC)
IS IT FUNCTION KEY (OR THAT GROUP?)
.
TSTIR SB,16,KBSCNT,CC
.
BRA KBKEY,FC NO, REGULAR KEY
.
DOTR ,AC,KBSCNT,,C0 DOUBLE THE NUMBER FOR JUMP TABLE
DOTA ,AC,KBTRL MAKE IT A BACKWARDS TABLE OFFSET
BAS TEMP1 PUSH
BRR TEMP1 RETURN
.
. INTO THE TABLE THIS IS TOP OF
.
LDTI SEKRDKY 17 KEYBOARD KEY
BRA KBKS1
LDTI SEDSPKY 16 DISPLAY KEY
BRA KBKS1
LDTI SEFUNC2 15 F2 KEY
BRA KBKS2
LDTI SEFUNC4 14 F4 KEY
BRA KBKS2
LDTI SEINTKY 13 INTERRUPT KEY
BRA KBREINT
    
```


. DISPLAY = KEYBOARD SERVICE ROUTINES

| | | | | | | |
|-------|---------|----------|----------|-------|---------|------------------|
| 1197, | 003114L | 01010001 | 00010000 | LDTI | SEFUNC5 | 12 F5 KEY |
| 1198, | 003115L | 11001111 | 10000001 | BRA | KBKS2 | |
| 1199, | 003116L | 01010001 | 00000100 | LDTI | SEFUNC3 | 11 F3 KEY |
| 1200, | 003117L | 11001111 | 10000001 | BRA | KBKS2 | |
| 1201, | 003120L | 01010001 | 00000100 | LDTI | SEKBRDY | 10 . (POINT) KEY |
| 1202, | 003121L | 11001111 | 10011110 | BRA | KBREG | |
| 1203, | 003122L | 01010001 | 00000100 | LDTI | SEKBRDY | 7 NO=KEY |
| 1204, | 003123L | 11001111 | 10011110 | BRA | KBREG | |
| 1205, | 003124L | 01010001 | 00000001 | LDTI | SEFUNC1 | 6 F1 KEY |
| 1206, | 003125L | 11001111 | 10000001 | BRA | KBKS2 | |
| 1207, | 003126L | 01010001 | 01000000 | LDTI | SEATTKY | 5 ATTENTION KEY |
| 1208, | 003127L | 11001111 | 10000001 | BRA | KBKS2 | |
| 1209, | 003130L | 01010001 | 00100000 | LDTI | SERSTKY | 4 RESTART KEY |
| 1210, | 003131L | 11001111 | 01110100 | BRA | KBREINT | |
| 1211, | 003132L | 01010001 | 00000100 | LDTI | SEKBRDY | 3 SPACE BAR |
| 1212, | 003133L | 11001111 | 10011110 | BRA | KBREG | |
| 1213, | 003134L | 01010001 | 00000100 | LDTI | SEKBRDY | 2 BACKSPACE KEY |
| 1214, | 003135L | 11001111 | 10011110 | BRA | KBREG | |
| 1215, | 003136L | 01010001 | 00000100 | LDTI | SEKBRDY | 1 CANCEL KEY |
| 1216, | 003137L | 11001111 | 10011110 | BRA | KBREG | |
| 1217, | 003140L | | | | | |
| 1218, | 003140L | 01010001 | 00000100 | KBTL | LDTI | 0 ENTER KEY |
| 1219, | | | | KBKEY | BRA | KBREG |

| | | | | | | |
|------|---------|----------|----------|--------|-------|--|
| 1220 | | | | | | |
| 1221 | 003141L | 01101111 | 11110001 | KBREG | LDRT | TEMP1 |
| 1222 | 003142L | 00110001 | 00110001 | | TSTIP | ,STKBKC,STATUS |
| | 003143L | 01000101 | 00010000 | | | NEW CLOSURE? |
| 1223 | 003144L | 11000010 | 10010101 | | BRA | KBGET,FZ |
| 1224 | 003145L | 00010001 | 11110001 | | TSTRR | XR,SCANSV,KBSCNT |
| | 003146L | 00010000 | 11000010 | | | YES, DO IT |
| 1225 | 003147L | 11000010 | 10000111 | | BRA | KBDON,FZ |
| 1226 | 003150L | 01010001 | 00001000 | | LDTI | SEKBDWN |
| 1227 | 003151L | 11001111 | 10000000 | | BRA | KBKS1 |
| 1228 | | | | | | CONTINUOUSLY SET THE BIT IF STILL DOWN |
| 1229 | | | | | | CLEAR BIT CONTINUOUSLY IF UP |
| 1230 | | | | | | IF GOES DOWN AGAIN GET NEW CLOSURE |
| 1231 | | | | | | RATHER THAN REPEATED CLOSURE! |
| 1232 | 003152L | 00110111 | 00001100 | KBGET | STB | IMAR |
| 1233 | 003153L | 00110001 | 01010000 | | LDPP | MDW,KBDD |
| | 003154L | 00110111 | 00100001 | | | KEYBOARD CHARACTER SAVE AREA |
| 1234 | 003155L | 00010001 | 11110001 | | LDRR | SCANSV,KBSCNT |
| | 003156L | 00000111 | 11110010 | | | OUTPUT THE DATA |
| 1235 | 003157L | 11000100 | 10010000 | | MWAIT | ,MEMPFDL |
| | 003160L | 11000111 | 11001101 | | | SAVE THE SCAN NUMBER (FOR REPEATS) |
| 1236 | 003161L | 00110111 | 00001101 | | STB | DMAR,SMR |
| | 003162L | 00110111 | 01000111 | | | POINT BACK TO SEKBS1 |
| 1237 | 003163L | 01110001 | 11110001 | KBCLS | LDTR | TEMP1 |
| 1238 | 003164L | 11000100 | 10001011 | | MWAIT | ,MEMPFDL |
| | 003165L | 11000111 | 11001101 | | | GET BIT TO SET |
| 1239 | 003166L | 00110011 | 00110110 | | DOPP | MDW,OR,MDR |
| | 003167L | 00110111 | 00100001 | | | SET IT |
| 1240 | 003170L | 00010001 | 11110001 | KBDOON | DOTRR | ,IT,0,KBSCNT,C1 |
| | 003171L | 00010110 | 01110010 | | | POINT TO THE NEXT SCAN ENTRY (CLEAR) |
| 1241 | 003172L | 01010101 | 00111111 | | DOPI | KBSC,ND,077 |
| | 003173L | 00110111 | 01000100 | | | KEEPING THE NUMBER IN RANGE (CARRY ALSO) |
| 1242 | 003174L | 00000111 | 11110001 | | LDRT | KBSCNT,CC |
| 1243 | 003175L | 11101111 | 00000000 | | BRR | LINK |
| 1244 | | | | | | (5.3 MICRO-SEC TIME TILL READY FROM NOW) |
| 1245 | 003176L | 00110111 | 00001101 | | | AND RETURN FROM KEY SERVICE ON ERROR TOO |
| 1246 | | | | | | |
| 1247 | 003177L | 00110111 | 01000111 | KBKS2 | STB | DMAR |
| 1248 | 003200L | 01101111 | 11110001 | | | POINT TO SEKBS2 |
| 1249 | 003201L | 00110001 | 00110001 | KBKS1 | STB | SMR |
| | 003202L | 01000101 | 00100000 | | LDRT | TEMP1 |
| 1250 | 003203L | 11000010 | 10001100 | | TSTIP | ,STKBNS,STATUS |
| 1251 | | | | | | KEY CLOSED? |
| 1252 | 003204L | 01110001 | 11110001 | | BRA | KBCLS,FZ |
| | 003205L | 01010000 | 11111111 | | | YES |
| 1253 | 003206L | 11000100 | 01111001 | KBOPN | DOTIR | ,XR,0377,TEMP1 |
| | 003207L | 11000111 | 11001101 | | | NO, OPEN, TURN THE BIT OFF |
| 1254 | 003210L | 00110101 | 00110110 | | MWAIT | ,MEMPFDL |
| | 003211L | 00110111 | 00100001 | | | BY ANDING IT OUT |
| 1255 | 003212L | 11001111 | 10000111 | | DOPP | MDW,ND,MDR |
| | | | | | | |
| | | | | | BRA | KBDOON |

```

1256.
1257. 003213L 01101111 11110001
1258. 003214L 00110111 00001101
      003215L 00110111 01000111
1259. 003216L 00110001 00110001
      003217L 01000101 00110000
1260. 003220L 11000010 10001100
1261.
1262. 003221L 01010001 10100000
1263. 003222L 11000100 01101101
      003223L 11000111 11001101
1264. 003224L 00110101 00110110
1265. 003225L 01000000 10100000
1266. 003226L 11000010 01111011
1267. 003227L 00110001 00110110
      003230L 01110000 11110001
      003231L 00110111 00100001
1268. 003232L 01010001 00000001
      003233L 00010010 00110001
      003234L 00110111 01000100
      003235L 00000111 11110001
1269.
1270.
1271.
1272.
1273.
1274.
1275.
1276.
1277.
1278.
1279.
1280.
1281.
1282.
1283. 003236L 01010001 00000011
      003237L 01101111 11110010
1284. 003240L 11000100 01011111
      003241L 11000111 11001101
1285. >003242L 01011001 11111111
      >003243L 11001001 11111111
1286. 003244L 00110001 11011100
      003245L 01000101 00100000
1287. >003246L 11000011 11111111
1288. 003247L 00110001 11001001
      003250L 00110001 11101000
      003251L 00110111 00001101
1289. 003252L 00110001 10001001
      003253L 00110001 10101000
1290. >003254L 11001111 11111111
    
```

```

+
KBREINT LDRT TEMP1 SAVE THE BIT
      STB DMAR,SMR IN SEKBK92
      TSTIP ,STKBKC+STKBNS,STATUS NEW CLOSURE OR STILL CLOSED?
      BRA KBCLS,FZ YES, ITS DOWN, SO SET IT!
      LDRT SEINTKY+SERSTKY
      MWAIT ,MEMPFDL IT IS UP! SEE IF BOTH KEYS WERE DOWN
      DOTP ,ND,MDR SELECT RESTART AND INTERRUPT KEYS
      TSTIT XR,SEINTKY+SERSTKY WERE BOTH SET?
      BRA KBOPN,FZ NO, ONLY ONE
      DOPRP MDW,XR,TEMP1,MDR YES, TURN THE BIT OFF & DO A RESTART
      DOPRI KBSC,AC,KBSCNT,1,C0 FINISH LIKE REGULAR, BUT INLINE
      LDRT KBSCNT ONLY KEYBOARD INVOLVED (INLINE SRVRTW)
*
*** NOTE: MAY WANT TO CHANGE OR ADD CODE TO STOP COMMUNICATIONS IN PROGRESS! **
      (WILL DO IT IN MACRO-LEVEL CODE)
.
.
. SPECIAL SERVICE REQUESTED INSTRUCTION LEVEL SYSTEM CALL INTERRUPT
. THE P.C. MUST BE BACKED UP IF IMP NON-ZERO, TO THE IMP SPEC.
. ELSE IT IS POINTING TO THE INSTRUCTION TO EXECUTE NEXT
. OR BEING EXECUTED IF REPEATED.
. NOTE: THE PRESENT WAY CODE WORKS IF NOT REPEATED (OR NOT STARTED YET)
. THEN IMP MUST BE ZERO BECAUSE CAN'T INTERRUPT BETWEEN IMP, SPEC & OPCODE
. BUT CODE BELOW WILL WORK ANYWAY (TOO GENERALIZED)
. I.E. IF IMP-NON-ZERO MUST BE REPEATED THEREFORE MUST BACK IT UP.
.
      LDRI TEMPL,SRRSTR SET RESTART ADDRESS
      MWAIT ,MEMPFDL
      BRAX SCROM,T0,IZ IMP ZERO, POINTING TO INSTRUCTION
      TSTIP ,SWRPT,PSWI IS IT REPEATED?
      BRA SCROM,TZ NO, THEN P.C. POINTS TO INSTRUCTION
      DLDX PC2MR,,DMAR YES, THEN BACK THE PC UP
      DLDX MR2PC BEFORE DOING THE RESTART
      BRA SCROM
    
```

LOAD CHARACTER FONT

```

1293
1294 003255L
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312 003255L 00110001 11011100
      003256L 01000101 00000100
1313 003257L 11000010 00100111
1314 003260L 00110001 11010000
      003261L 01010101 01111111
1315 003262L 00110111 01000011
1316 003263L 00010111 10110010
1317
1318 003264L 01010100 00001010
1319 003265L 11000001 00101011
1320
1321 003266L 00110111 01000010
      003267L 00110111 01000010
1321 003270L 00110111 01000010
      003271L 00110111 01000010
1321 003272L 00110111 01000010
      003273L 00110111 01000010
1321 003274L 00110111 01000010
      003275L 00110111 01000010
1321 003276L 00110111 01000010
      003277L 00110111 01000010
1322 003300L 11000010 01001011
1323
1324
1325 003301L 01010001 00000111
      003302L 00110111 00000001
1326 003303L 00110001 11100101
1327 003304L 00110001 11000110
      003305L 00110111 01000111
1328 003306L 00110111 00001001
1329 003307L 11000100 00111000
      003310L 11000111 00100101
1330 003311L 00110111 00001100
    
```

```

*
LDCFP:
.      ( 155) LDCFP          LOAD CHARACTER FONT
.                                     (MAKE REPEATED USING C?)
.
. * * * 21.8 + A * 0.15 + INT (A / 10) * 0.4 * * *
.          = 1.0 IF REM (A / 10) = 0 AND A # 0
.
.      IF USER THEN IVDLS
.      SDLM                  SET DISPLAY LOAD MODE
.      REPEAT 'A' TIMES SKCH SKIP CHARACTER FONTS
.      FOR IMP = 7 TO 1
.          MARL <- (HL)
.          HL <- HL + 1
.          LDCH              LOAD FONT ROW AND BACK UP TO NEXT
.          END
.      RDLM                  RESET DISPLAY LOAD MODE
.
.      TSTIP ,SWUSER,PSWI
.
.      BRA IVDL3,FZ
.      DOTIP ,ND,0177,URI+URA ONLY 7 BITS OF DATA!
.
.      STB SDLM              PUT IT IN LOAD FONT MODE
.      CCLR
.
.      LDCFPT DOTI ,SB,10    COUNT IN STEPS OF TEN
.      BRA LDCFNL,TC        IF AT END, DO FINAL STUFF
.      LDCFLP RPT 5          THERE ARE TEN STROBES IN THE SPEEDUP
.      STB SKCH,SKCH
.
.      STB SKCH,SKCH
.
.      STB SKCH,SKCH
.
.      STB SKCH,SKCH
.
.      STB SKCH,SKCH
.
.      STB SKCH,SKCH
.
.      BRA LDCFPT,FZ        DONE AND DONE WELL (I HOPE)
.      QUESTION, IF ROOM, DO IN 16'S AND JUMP ENTRY WITH IMP COUNTING GROUPS LEFT
.
.      LDPI LIMP,7          SET UP FOR 7 ROWS PER FONT
.
.      LDX HL2MRH           INIT MARH TO FONT STRING
.      LDCFLD LDX HL2MRL,SMR NOW CAN FINALLY START ACTUAL FONT LOAD
.
.      STB DIMP            LOADING MARL >400 S AFTER LDCH
.      MWAIT ,MEMPFCE      IF ERROR, RESET LOAD MODE
.
.      STB IMAR
    
```

. LOAD CHARACTER FONT

1331 003312L 00110001 10000110
 1332 003313L 00110001 00110110
 003314L 00110111 11000000
 003315L 00110111 01000000
 1333 003316L 11001111 00110000
 1334 003317L 11001000 00111011
 1335 003320L 00110001 10100101
 1336 003321L 00110111 01000101
 1337 003322L 01011001 11111011
 003323L 11001111 11101101
 1338
 1339 003324L
 1340
 1341
 1342
 1343
 1344 003324L 01010010 01001000
 1345 003325L 01101111 10110000
 1346 003326L 01010001 00000000
 1347 003327L 11101111 00000000
 1348
 1349 >003330L 01011001 11111111
 >003331L 11001111 11111111
 1350
 1351 003332L 00110111 01000101
 1352 003333L 01011001 11111011
 003334L 11001111 01101010
 1353
 1354
 1561
 1562

LDX MR2XXL+URL UPDATE HL THE FASTEST WAY
 LDPP MAROL,MDR,LDCH GET THE CHARACTER OUT AND LOAD THE FONT
 BRA S+1 200 SEC. DELAY ADDED FOR LDCH (400 S)
 BRA LDCFLD,F0,IZ LOOP TILL ALL SEVEN ROWS DONE
 LDX MR2XXH+URH GET NEW HL CAUSE L UPDATED
 STB RDLM RESET LOAD MODE (>400 S AFTER LDCH)
 BRAX FETCH
 .
 LDCFNL
 . FOR THE FINAL STEP ONLY GO PARTIALLY THROUGH THE LOOP
 . T >=0=10 STEPS, -1=9 STEPS, -2=8 STEPS, ... -9=1 STEP
 . SKIPS 0 STEPS, 1 STEP 2 STEPS, ... 9 STEPS
 .
 DOTA ,AC,LDCFLP+1 ADDS ADDRESS TO NEGATIVE VALUE (CRY SET)
 BAS LINK,CC MOVES FORWARD FROM LDCFSL (1 TO 9 STEPS)
 TCLR MARK ENDED
 BRR LINK DO AS MANY STEPS AS NEEDED
 *
 IVIOL3 BRAX IVIOLS
 .
 MEMPFCF STB RDLM ** SPECIAL FOR LDCF, OUT OF LOAD MODE **
 MEMPF3 BRAX MEMPF3
 .
 IFC APF
 XIF
 IFS APF

MICRO-PROCESSOR EMULATION SUPPORT CODE - HJS - 78JUL20 11:44
 . NOISE-MAKER INTERFACE. BEEPS, CLICKS & MAKES OTHER FUNNY NOISES

```

1565,
1566,
1567,
1568,
1569,
1570,
1571,
1572,
1573,
1574,
1575,
1576,
1577,
1578,
1579,
1580, 000203
1581, 000253
1582, 000310
1583, 000316
1584, 000360
1585,
1586, 003335L
1587,
1588,
1589,
1590, 003335L 01010001 10000011
1591, 003336L 01101111 11110001
1592, 003337L 00110001 11011100
      003340L 01000101 00000100
1593, 003341L 11000010 00100111
1594, 003342L 01011001 11111011
1595, 003343L 00010001 11001111
1596, 003344L 11000010 11101011
1597, 003345L 01110001 11110001
      003346L 00000111 11111111
1598, 003347L 11001111 11101011
1599,
1600, 003350L 11001110 11111111
1601,
1602, 003351L
1603,
1604,
1605,
1606,
1607, 003351L 01010001 11110000
1608, 003352L 11001010 00100001
1609, 003353L 01010001 10101011
1610, 003354L 11001111 00100001
1611,
1612, 003355L 11111111 11111111
      003356L 11111111 11111111
      003357L 11111111 11111111
    
```

```

*
. DEFINE THE CONTROL BITS FOR THE AUDIO CHANNEL
.
. TOGGLE B7      0  LOW OUTPUT (IF BEEPING)
                  1  HIGH OUTPUT (ALWAYS)
. FCN      B6,5  00  BEEP WITH LONG DELAY (NORMAL)
                  01  BEEP WITH SHORT DELAY
                  10  DELAYING
                  11  CLICKING
. DONE      B4      0  CONTINUE DOING OPERATION
                  1  STOP, THE END (IF B7 IS 0)
. COUNTER B3,2,1,0 COUNT AT A 60/50HZ RATE
*
. NOW, DEFINE THE CONTROL COMMANDS (INITIAL/FINAL VALUES)
.
BPINIT EQU      B7+B4-13      STANDARD BEEP  ENDS WITH B7+B4
BPSHRT EQU      B7+B5+B4-5     SHORT LENGTH  ENDS WITH B7+B5+B4
DLYLNG EQU      B7+B6+B4-8     STANDARD DELAY ENDS WITH B7+B6+B4
DLYSRT EQU      B7+B6+B4-2     SHORT DELAY   ENDS WITH B7+B6+B4
CLKINI EQU      B7+B6+B5+B4-0   NORMAL CLOCK  ENDS WITH B7+B6+B5+B4
.
BEEP:
.      ( 151)  EX BEEP          START AUDIBLE CHANNEL IF QUIET
.      (IMP 151) EX BEEP       START AUDIBLE CHANNEL IF QUIET
.
CLICKS
.      LD TI  BPINIT          SETUP FOR BEEP OPERATION
.      LD RT  TEMP1          SAVE AWAY THE CONTROL BYTE
.      TSTIP ,SWUSER,PSWI    IS IT LEGAL TO DO THE INSTRUCTION?
.
.      BRA   IVIOL3,FZ        NO, ERROR IT OUT
.      BPGX  FETCHI
.      TSTRT FI,ACCTL        ANY SOUNDS IN PROGRESS
.      BRA   FETCHI,FZ        YES, SO DO NOTHING
.      LD RR ACCTL,TEMP1     NO, SO INIT TO DO THE NEW SOUND
.
.      BRA   FETCHI
.
.      NOOP                  (SO APF & NON-APF CLICKS ALIGN)
*
CLICK:
.      ( 151)  EX CLICK       START AUDIBLE CHANNEL IF QUIET
.      (IEV 151) EX CLICK     START AUDIBLE CHANNEL IF QUIET
.      (IOD 151) CLICKR      SPECIAL NOISE IF QUIET
.
.      LD TI  CLKINI          ASSUME NORMAL CLICK
.      BRA   CLICKS,F0,IO    IT WAS!
.      LD TI  BPSHRT         NO, WAS SPECIAL SHORT BEEP
.      BRA   CLICKS
.
.      TABPAGE PRODL
    
```


APF SPECIAL INTERFACE CONTROL INSTRUCTION

```

1646,
1647, 003424L
1648,
1649,
1650,
1651,
1652,
1653,
1654,
1655,
1656,
1657,
1658,
1659,
1660,
1661,
1662,
1663,
1664, 003424L 00110001 11011100
      003425L 01010101 00000100
1665, 003426L 11000010 00100111
1666, 003427L 01101111 11110010
1667, 003430L 11011000 11011101
1668, 003431L 00110111 00001100
1669, 003432L 00110001 10001001
      003433L 00110001 10101000
1670, 003434L 01010001 01011010
      003435L 00110111 11000000
      003436L 01010001 11101111
      003437L 00110111 11100000
1671, 003440L 01011001 11111000
      003441L 11011111 01000111
1672,
1673,
1674, 003442L
1675,
1676, 003442L 00110111 00001001
1677, 003443L 01010001 11101111
      003444L 00110111 11100000
1678, 003445L 11011010 10110101
1679,
1680, 003446L 00110111 00001001
1681, 003447L 11001110 11111111
1682, 003450L 11011000 11001111
1683,
1684, 003451L 01010001 01011110
      003452L 00110111 11000000
      003453L 00110111 01000111
1685, 003454L 00010001 11111100
      003455L 00110111 10000010
1686, 003456L 00010001 10111011
1687, 003457L 11011111 11001001
    
```

```

*
MTART:
.      (      147) WR61          WRITE TO THE 61 (START)
.      (111 147) APFRL        LOAD THE RECEIVER CONTROLS
.      (062 147) APFRS        SAVE (GET) THE RECEIVER CONTROLS
.      (113 147) APFTL        LOAD THE TRANSMITTER CONTROLS
.      (174 147) APFTR        SAVE (GET) THE TRANSMITTER CONTROLS
.
.
.      INSTRUCTION MTART (ADDR)          PROCESS B STARTUP
* * * * 1.95 * * * * + INTERRUPT ROUTINE + FETCH AS EXTENDED
.      BEGIN
.      IF SWUSER IN PSW THEN EXIT (I VIOL$);
.      MTADR := ADDR;                  ASSUMED BY DEFAULT
.      PC := MAR + 1;                  UPDATE PC & IMP BECAUSE
.      IMP := 0;                       ABNORMAL EXIT TAKEN
.      IAKCODE := 0;                   INTERRUPT ITNXT, NO REPEAT
.      EXIT (ITNXT)
.      TSTIP ,SWUSER,PSWI,TW
.
.      BRA      I VIOL3,FZ
.      LDRT     TEMP2                    (IAKODE IS ZERO!)
.      BRA      APFCTL,F0,IZ
.      STB      IMAR
.      DLDX     MR2PC
.
.      DLDPI    MAR0,MTTLEN
.
.      BRAX     ITNXT                    ** COULD SAVE A BYTE HERE **
.
.      END MTART;
*
APFCTL
.
.      STB      DIMP                      IN RANGE 0 TO 3 NOW
.      LDPI     MAR0H,MTTLEN>8
.
.      BRA      APFL0D,F0,I0
.
.      STB      DIMP                      2,4 CASE NOW 0,2
.      NOOP
.      BRA      APFGXMT,F0,IZ            SEE IF RCV OR XMT INFO WANTED
.
.      LDPI     MAR0L,MTRPNT,SMR        WAS RECEIVE, GET MEMORY AND CONTROL DATA
.
.      LDPR     UR0+URC,APFRK
.
.      LDTR     APFRP,CC
.      BRA      APFGRCV
    
```



```

1688.
1689. 003460L
1690. 003460L 01010001 01011011
      003461L 00110111 11000000
      003462L 00110111 01000111
1691. 003463L 00010001 11111110
      003464L 00110111 10000010
1692. 003465L 00010001 10111101
1693.
1694. 003466L
1695. 003466L 00110111 10000110
1696. 003467L 11010100 11001000
      003470L 11000111 00100100
1697. 003471L 00110001 00110110
1698. 003472L 00110111 00001101
      003473L 00110111 01000111
1699. 003474L 00110111 10000101
1700. 003475L 11010100 11000010
      003476L 11000111 00100100
1701. 003477L 00110001 00110110
1702. 003500L 00110111 00001101
      003501L 00110111 01000111
1703. 003502L 00110111 10000001
1704. 003503L 11010100 10111100
      003504L 11000111 00100100
1705. 003505L 00110001 00110110
      003506L 00110010 00110110
      003507L 00110111 00000110
1706. 003510L 01011001 11111011
      003511L 11001111 11101101
1707.
1708. 003512L
1709. 003512L 11011000 10101011
1710. 003513L 01010001 01011110
      003514L 00110111 11000000
1711. 003515L 00110001 11010101
      003516L 00110111 00100001
1712. 003517L 00110001 11010110
      003520L 00000111 11111011
1713. 003521L 00110001 11010010
      003522L 00000111 11111100
1714. 003523L 11011111 10100011
1715.
1716. 003524L
1717. 003524L 01010001 01011011
      003525L 00110111 11000000
1718. 003526L 00110001 11010101
      003527L 00110111 00100001
1719. 003530L 00110001 11010110
      003531L 00000111 11111101
1720. 003532L 00110001 11010010
      003533L 00000111 11111110
    
```

APFGXMT

LDPI MAROL,MTTPNT,SMR WAS TRANSMIT DATA THAT DESIRED

LDPR URD+URC,APFTK

LDTR APFTP,CC

APFGRCV

LDPT URD+URL GOT THE LSB POINTER

MWAIT ,MEMPF3

LDTP MDR GOT THE MSB POINTER FROM MEMORY

STB DMAR,SMR

LDPT URD+URH

MWAIT ,MEMPF3

LDTP MDR GOT THE MSB LENGTH

STB DMAR,SMR

APFCC

LDPT URD+URB

MWAIT ,MEMPF3

DOPPP LUF,AC,MDR,MDR SET THE USER FLAGS ON THE FLAG BYTE

BRAX FETCH

APFL0D

BRA APFLXMT,F#,IZ LOAD UP THE XMIT CONTROL BYTES

LDPI MAROL,MTRPNT NO, THE RECEIVE BYTES

LDPP MDW,URI+URH MSB POINTER

LDRP APFRP,URI+URL

LDRP APFRK,URI+URC LSB POINTER AND LENGTH

BRA APFLRCV

APFLXMT

LDPI MAROL,MTTPNT

LDPP MDW,URI+URH MSB POINTER

LDRP APFTP,URI+URL

LDRP APFTK,URI+URC LSB POINTER AND LENGTH

```

1721.
1722. 003534L
1723. 003534L 11010100 10100011
      003535L 11000111 00100100
1724. 003536L 00110111 00001101
1725. 003537L 00110001 11010001
      003540L 00110111 00100001
1726. 003541L 01010001 00000000
1727. 003542L 11010100 10011101
      003543L 11000111 00100100
1728. 003544L 00110111 00001101
1729. 003545L 00110111 00100001
1730. 003546L 01011001 11111011
      003547L 11001111 11101111

1731.
1732. 003550L
1733.
1734.
1735.
1736.
1737. 003550L 00110001 11011100
      003551L 01000101 00000100
1738. 003552L 11000010 00100111
1739. 003553L 11011000 10001000
1740.
1741. 003554L 01010001 00001101
      003555L 11011100 10010010
      003556L 00110111 00100110
      003557L 01010001 00000001
      003560L 00110111 00100111
      003561L 00110111 00101001
1742. 003562L 01010001 00001001
      003563L 00110111 00000001
1743. 003564L 01011001 11111011
      003565L 01010001 00000011
      003566L 11001111 10001001

1744.
1745. 003567L 01010001 01011100
1746. 003570L 11011011 10000101
1747.
1748. 003571L 01010001 01011001
1749.
1750. 003572L 00110111 11000000
1751. 003573L 01010001 11101111
      003574L 00110111 11100000
      003575L 00110111 01000111
1752. 003576L 00010111 10110010
1753. 003577L 11011111 10111100

```

* APFLRCV

```

MWAIT ,MEMPF3
STB DMAR
LDPP MDW,URI+URB MSB LENGTH
TCLR
MWAIT ,MEMPF3
STB DMAR
LDPT MDW AND ZERO THE FLAG BYTE
BRAX FETCHW

```

* AMLRET:

```

( 177) AMLRET RE-ENABLE AML CHANNEL THEN DO SYSTEM RET
(111 177) APFRST GET APF RECEIVER STATUS
(062 177) APFTST GET APF TRANSMITTER STATUS

TSTIP ,SWUSER,PSWI INSURE THAT ALLOWED TO DO IT

BRA IVIOL3,FZ
BRA APFSTAT,F0,IZ WILL BE APF STATUS OPS

MBUSC ,AMLADR,MBON RE-ENABLE THE AML CHANNEL

LDPI LIMP,PCL THEN DO A SYSTEM RETURN

BRCX POPST0,,SYSRET0

```

*

```

APFSTAT LDTI MTRFLG WANT THE RECEIVER STATUS?
BRA APFRST,T0,IO

```

*

```

LDTI MTTFLG NO, WANT THE TRANSMITTER STATUS

```

* APFRST

```

LDPT MAROL
LDPI MAROH,MTTFLG>8,SMR

```

```

CCLR
BRA APFCC USE COMMON CODE POINT

```

```

1756,
1757, 003600L
1758,
1759,
1760,
1761,
1762, 003600L 01010001 01011101
      003601L 00110111 11000000
1763,
1764, 003602L 01010001 01001110
      003603L 11011100 01111100
      003604L 00110111 00100110
      003605L 00110111 00101001
1765, 003606L 01110001 11110010
      003607L 01000101 01000000
      003610L 11010010 01010011
1766,
1767,
1768, 003611L
1769,
1770,
1771,
1772, 003611L 00010001 10111100
1773, 003612L 11010010 01101010
1774, 003613L 00110111 01000111
1775, 003614L 01010001 11111111
1776, 003615L 11010100 01110010
      003616L 11010111 00001100
1777, 003617L 00110010 00110110
1778, 003620L 11010000 01010101
1779, 003621L 00110111 00100001
1780, 003622L 00010001 10111100
1781, 003623L 11010100 01101100
      003624L 11010111 00001100
1782, 003625L
1783, 003625L 00110111 00001100
      003626L 00110111 01000111
1784, 003627L 01010100 00000001
      003630L 00000111 11111100
1785, 003631L 00010001 11111011
      003632L 00000110 01111011
1786, 003633L 11010100 01100100
      003634L 11010111 00001100
1787, 003635L 00110001 00110110
1788, 003636L 11010000 01011100
1789, 003637L 00110110 00100001
1790, 003640L 11001110 11111111
      003641L 11010100 01011110
      003642L 11010111 00001100
1791, 003643L
1792, 003643L 00110111 11100000
1793, 003644L 00010001 11111011
      003645L 00110111 11000000
    
```

```

*
IRINT:
. (+0.10 ADDED FOR EXTRA TIME TO GET HERE RELATIVE ITINT)
.
. NOTE: THERE IS ROOM TO EXTEND CODE FOR SPEED, I.E. ELIMINATE COMMON ENDS
.
      LDPI   MAROL,MTRLEN           (POINT TO RECEIVER BUFFER)
.
      MBIN (MTADR, MCDIN);         STROBE FOR THE DATA BYTE
      MBUSC ,MTADR+MCDIN
.
      TSTIR ,MIRNXT,XOR,MIREOT,TEMP2
.
      BRA   IREOT,FZ               (RECEIVED END OF TRANSFER)
*
IRNXT:
. ROUTINE IRNXT                     CHANNEL A INPUT NEXT BYTE
. BEGIN
.
      END;
      LDTR  APFRK,CC               GET LENGTH LSB
      BRA  IRNXT0,FZ
      STB  SMR                     WAS ZERO, GET ITS MSB
      LDTI -1
      MWAIT ,MEMPFAP
.
      DOTP ,AC,MDR                 IF MSB ALSO ZERO,
      BRA  IRTOVR,FC               OVERFLOW, IGNORE THE DATA
      LDPT MDW                     UPDATE LENGTH MSB
      LDTR  APFRK,CC               (LOADS ZERO)
      MWAIT ,MEMPFAP
.
IRNXT0
      STB  IMAR,SMR
.
      DORI  APFRK,SB,1             UPDATE LENGTH LSB
.
      INCR  APFRP,APFRP            UPDATE POINTER LSB
.
      MWAIT ,MEMPFAP
.
      LDTP  MDR                     GET POINTER MSB
      BRA  IRNXT1,FC
      DOP  MDW,IT                  UPDATE POINTER MSB
      MWAIT NOOP,MEMPFAP
.
IRNXT1
      LDPT  MAROH                   LOAD UP POINTER MSB
      LDPR  MAROL,APFRP,DMAR       AND LSB (WHICH IS HIGH BY ONE)
    
```

```

1794. 003646L 00110111 00001101
1795.
1796. 003647L 11011100 01011000
      003650L 00110001 00010101
1797. 003651L 11011111 01001100
1798.
1799.
1800. 003652L 01010001 01000000
1801. 003653L 11011111 01010010
1802.
1803.
1804. 003654L
1805.
1806.
1807.
1808.
1809.
1810. 003654L 01010001 10000000
1811. 003655L
1812. 003655L 00110111 00001101
      003656L 00110111 01000111
1813. 003657L
1814. 003657L 11001110 11111111
      003660L 11010100 01001111
      003661L 11010111 00001100
1815. 003662L 00110011 00110110
1816. 003663L 00110111 00100001
1817. 003664L 01011001 11111010
      003665L 11011111 00010010
1818.
1819.
    
```

```

.      (MTRPNT) := MIFIN;          STORE THE DATA
.      RETURN
.      MBIN                          (GET THE DATA)
.
.      JMP      IRTDTA
.      END IRNXT;
*
*
IRTOVR  LDTI  MFOVR          SET OVERFLAG
        BRA  IRTEND
.
.
IREOT
. 3.15
.      ROUTINE IREOT              CHANNEL A END TRANSFER
.      BEGIN
.      (MTRFLG) := (MTRFLG) .OR. MFDON;  SET TRANSFER COMPLETE FLAG
.      RETURN
.      LDTI  MFDON
IRTEND  (SPECIAL COMMON ENDPOINT)
        STB  DMAR,SMR
.
IRTFND  MWAIT  NOOP, MEMPFAP      (?? SET MEMPF IN STATUS?)
.
.      DOTP  ,OR, MDR          JUST SET THE FLAG
IRTDTA  LDPT  MDW              (SAVE DATA OR FLAGS AWAY)
IRTFIN  BRAX  SRVBSN
.
.      END IREOT;
.
    
```

```

1820,
1821, 003666L
1822, 003666L 01000101 01000000
1823, 003667L 11010010 00010100
1824,
1825,
1826, 003670L
1827,
1828,
1829,
1830,
1831,
1832,
1833,
1834,
1835, 003670L 00010001 10111110
1836, 003671L 11010010 00111011
1837, 003672L 00110111 01000111
1838, 003673L 01010001 11111111
1839, 003674L 11010100 01000011
      003675L 11010111 00001100
1840, 003676L 00110010 00110110
1841, 003677L 11010000 00010110
1842, 003700L 00110111 00100001
1843, 003701L 00010001 10111110
1844, 003702L 11010100 00111101
      003703L 11010111 00001100
1845, 003704L
1846, 003704L 00110111 00001100
      003705L 00110111 01000111
1847, 003706L 01010100 00000001
      003707L 00000111 11111110
1848, 003710L 00010001 11111101
      003711L 00000110 01111101
1849, 003712L 11010100 00110101
      003713L 11010111 00001100
1850, 003714L 00110001 00110110
1851, 003715L 11010000 00101101
1852, 003716L 00110110 00100001
1853, 003717L 11001110 11111111
      003720L 11010100 00101111
      003721L 11010111 00001100
1854, 003722L
1855, 003722L 00110111 11100000
1856, 003723L 00010001 11111101
      003724L 00110111 11000000
1857, 003725L 00110111 00001101
      003726L 00110111 01000111
1858,
1859,
1860, 003727L 11011100 00101000
1861, 003730L 01010001 01011110
    
```

```

+
ITINT:
      TSTIT ,MITNXT,XOR,MITBOT          *MULTI-WAY-BRANCH CONTINUED*
      BRA  ITBOT,FZ                      ** FZ IT SHOULD BE!! **
      BRA  ITNXT
.
*
ITNXT:
.   ROUTINE ITNXT                          CHANNEL B NEXT BYTE
.   BEGIN
.   IF CARRY                              BUFFER WENT EMPTY
.   THEN BEGIN
.       MBOUT (MTADR, MCTEOT);           MARK END OF THE BUFFER
.       (MTTFLG) := (MTTFLG) .OR, MFEOT; MARK TRANSFER DONE
.       RETURN
.
      LDTR  APFTK,CC                       TEST LENGTH LSB
      BRA  ITNXT0,FZ                       SKIP MSB UPDATE IF NOT ZERO
      STB  SMR
      LDTI  -1
      MWAIT ,MEMPFAP
.
      DOTP  ,AC,MDR                       TEST LENGTH MSB
      BRA  ITEOT,FC                       WAS ZERO, COMPLETED TRANSFER
      LDPT  MDW                             UPDATE LENGTH MSB
      LDTR  APFTK,CC
      MWAIT ,MEMPFAP
.
ITNXT0
      STB  IMAR,SMR
.
      DORI  APFTK,SB,1                     UPDATE LENGTH LSB
.
      INCR  APFTP,APFTP                     UPDATE POINTER LSB
.
      MWAIT ,MEMPFAP
.
      LDTP  MDR
      BRA  ITNXT1,FC
      DDP  MDW,IT                          UPDATE POINTER MSB IF NEEDED
      MWAIT NOOP,MEMPFAP
.
ITNXT1
      LDPT  MAROH                           USE POINTER MSB AND LSB
      LOPR  MAROL,APFTP
.
      STB  DMAR,SMR                       BUT CORRECT FOR AUTO-INCREMENT
.
      MBOUT (MTADR, MCDOUT, (MTTPNT));    OUTPUT THE DATA
      MBUSA ,MTADR+MCDOUT
      MBWAIT (WAIT ON MICRO-BUS)
      LDPI  MIFADR,MTADR+MCDOUT
    
```

| | | | |
|-------|---------------------------|--|---------------------------------|
| 1862. | 003731L 00110111 00100110 | | |
| | 003732L 11010100 00100101 | MWAIT ,MEMPFAP | (??? = WAIT IN MEMORY) |
| | 003733L 11010111 00001100 | | |
| 1863. | 003734L 00110001 00110110 | MBUSP MDR | |
| | 003735L 00110111 00100111 | | |
| 1864. | 003736L 00110111 00101001 | MBUSS | (STROBE DATA OUT) |
| 1865. | | IF MIRPT IN IAKCODE | WAS REPEAT BIT SET? |
| 1866. | | THEN (MTTFLG) := (MTTFLG) .OR. MFRPT; | YES, THEN SET REPEAT FLAG |
| 1867. | | RETURN | |
| 1868. | 003737L 01110001 11110010 | TSTIR ,MFRPT>1,TEMP2 | (IS REPEAT NAK BIT SET?) |
| | 003740L 01000101 00100000 | | |
| 1869. | 003741L 11010011 01001011 | BRA IRTFIN,TZ | (NO, DONE) |
| 1870. | 003742L 01010001 01011001 | DLOPI MARD,MTTFLG,,SMR | |
| | 003743L 00110111 11000000 | | |
| | 003744L 01010001 11101111 | | |
| | 003745L 00110111 11100000 | | |
| | 003746L 00110111 01000111 | | |
| 1871. | 003747L 01010001 01000000 | LDTI MFRPT | (YES, THEN SET IT IN FLAG BYTE) |
| 1872. | 003750L 11011111 01010000 | BRA IRTFND | |
| 1873. | | | |
| 1874. | 003751L | ITEOT | |
| 1875. | 003751L 01010001 01101110 | LDTI MTADR+MCTEOT | (INFORM GOT END OF TRANSFER) |
| 1876. | 003752L 11011111 00010011 | BRA ITEND | |
| 1877. | | END; | |
| 1878. | | END ITNXT; | |
| 1879. | | | |
| 1880. | 003753L | ITBOT | |
| 1881. | | ROUTINE ITBOT | CHANNEL B READY OR REPEAT |
| 1882. | | BEGIN | |
| 1883. | | MBOUT (MTADR, MCTRDY); | SET THAT READY TO TRANSFER |
| 1884. | 003753L 01010001 01111110 | LDTI MTADR+MCTRDY | |
| 1885. | | (MTTFLG) := (MTTFLG) .OR. (IAKODE < 1 .AND. 0300); | SET READY FLAG AND MAYBE |
| 1886. | | | THE REPEAT FLAG = NAK |
| 1887. | | | |
| 1888. | | BRA ITEND | (USE COMMON ROUTINE) |
| 1889. | | RETURN | |
| 1890. | | END ITBOT; | |
| 1891. | | | |
| 1892. | 003754L | ITEND | (THE COMMON UPDATE POINT) |
| 1893. | 003754L 11011100 00010011 | MBWAIT | |
| 1894. | 003755L 00110111 00100110 | STB MIFADR,MIFSTB | |
| | 003756L 00110111 00101001 | | |
| 1895. | 003757L 01110001 11110010 | LDTR TEMP2 | (GET INTERRUPT CODE BITS) |
| 1896. | 003760L 00010111 10100010 | SHIFT SL | |
| 1897. | 003761L 01010101 11000000 | DOTI ,ND,MITRPT<1 | (FOR MFDON & MFRPT STATUS BITS) |
| 1898. | 003762L 11011111 01010010 | BRA IRTEND | |

```

1899.
1900.
1901. 003763L 01010001 00001110
      003764L 11011100 00001011
      003765L 00110111 00100110
      003766L 01010001 00000000
      003767L 00110111 00100111
      003770L 00110111 00101001

1902.
1903. 003771L 01011001 11111011
      003772L 11001111 01101010

1904.
1905.
1906.
1947.
1948.
1949. 003773L 11111111 11111111
      003774L 11111111 11111111
      003775L 11111111 11111111
      003776L 11111111 11111111
      003777L 11111111 11111111

1950. 002000
1951. 002000
1952. 002000
1953.

```

```

*
MEMPFAP MBUSC ,MTADR+MCEINT,MBOFF TURN THE BUSS OFF

BEFORE DO INTERRUPT CODE

BRAX MEMPFS

XIF
IFC TYPE
XIF

TABPAGE PRODL

PRODLN EQU S-PRODP
USE PRODL
SKIP PRODLN
END

```


| | | | | | | | | | | | | | |
|--------|---------|-------|------|------|------|------|------|------|------|------|------|------|------|
| | I350 | 618 | 640 | 671 | | | | | | | | | |
| | IDCODH | 138 | | | | | | | | | | | |
| | IDCODL | 138 | | | | | | | | | | | |
| | IIMP | 255 | 407 | 459 | 497 | 539 | | | | | | | |
| | IM2MRH | 538 | 638 | 660 | | | | | | | | | |
| | IM2MRL | 638 | 660 | | | | | | | | | | |
| | IMAR | 79 | 98 | 129 | 178 | 182 | 186 | 208 | 247 | 251 | 336 | 465 | 576 |
| | | 577 | 586 | 589 | 616 | 704 | 1138 | 1232 | 1330 | 1668 | 1783 | 1846 | |
| | IMP8 | 81 | | | | | | | | | | | |
| | IMPFO | 782 | 787 | 792 | 803 | 815 | | | | | | | |
| | IMPH | 491 | | | | | | | | | | | |
| | IMPI | 204 | 206 | 404 | 460 | 463 | 535 | 777 | 782 | 787 | 792 | 803 | 815 |
| | IMPL | 248 | 398 | 457 | | | | | | | | | |
| | IMPO | 252 | 255 | 337 | 340 | 406 | 497 | 500 | 539 | 540 | | | |
| 020005 | IO | *411A | 271 | 489 | 1608 | 1678 | 1746 | | | | | | |
| 003654 | IREOT | *1804 | 1766 | | | | | | | | | | |
| 003600 | IRINT | *1757 | 1037 | | | | | | | | | | |
| 003611 | IRNXT | *1768 | 1034 | | | | | | | | | | |
| 003625 | IRNXT0 | *1782 | 1773 | | | | | | | | | | |
| 003643 | IRNXT1 | *1791 | 1788 | | | | | | | | | | |
| 003663 | IRTDTA | *1816 | 1797 | | | | | | | | | | |
| 003655 | IRTEND | *1811 | 1801 | 1898 | | | | | | | | | |
| 003664 | IRTFIN | *1817 | 1869 | | | | | | | | | | |
| 003657 | IRTFND | *1813 | 1872 | | | | | | | | | | |
| 003652 | IRTOVR | *1800 | 1778 | | | | | | | | | | |
| | IT | 461 | 1125 | 1240 | 1785 | 1789 | 1848 | 1852 | | | | | |
| 003753 | ITBOT | *1880 | 1823 | | | | | | | | | | |
| 003754 | ITEND | *1892 | 1876 | | | | | | | | | | |
| 003751 | ITEOT | *1874 | 1841 | | | | | | | | | | |
| 003666 | ITINT | *1821 | 1036 | | | | | | | | | | |
| 003670 | ITNXT | *1826 | 1671 | | | | | | | | | | |
| 003704 | ITNXT0 | *1845 | 1836 | | | | | | | | | | |
| 003722 | ITNXT1 | *1854 | 1851 | | | | | | | | | | |
| | ITW | 913 | 927 | 995 | 999 | 1224 | 1595 | | | | | | |
| | IVIOL5 | 545 | 1349 | | | | | | | | | | |
| 002421 | IVIOL2 | *545 | 80 | 99 | 534 | | | | | | | | |
| 003330 | IVIOL3 | *1349 | 1313 | 1593 | 1665 | 1738 | | | | | | | |
| 020004 | IZ | *401A | 80 | 99 | 162 | 234 | 362 | 456 | 636 | 658 | 1285 | 1334 | 1667 |
| | | 1682 | 1709 | 1739 | | | | | | | | | |
| 002433 | JUMP | *580 | 271 | 567 | | | | | | | | | |
| 002423 | JUMPC | *550 | | | | | | | | | | | |
| 003163 | KRCLS | *1237 | 1250 | 1260 | | | | | | | | | |
| 003064 | KBD\$DO | *1151 | 1111 | 1144 | | | | | | | | | |
| | KBDD | 1233 | | | | | | | | | | | |
| 003170 | KBDON | *1240 | 1225 | 1255 | | | | | | | | | |
| 003070 | KBDWAIT | *1175 | 1176 | | | | | | | | | | |
| 003152 | KBGET | *1232 | 1223 | | | | | | | | | | |
| 003140 | KBKEY | *1218 | 1179 | | | | | | | | | | |
| 003177 | KBKS1 | *1247 | 1188 | 1190 | 1227 | | | | | | | | |
| 003176 | KBKS2 | *1245 | 1192 | 1194 | 1198 | 1200 | 1206 | 1208 | | | | | |
| 003204 | KBOPN | *1252 | 1266 | | | | | | | | | | |
| 003141 | KBREG | *1221 | 1202 | 1204 | 1212 | 1214 | 1216 | | | | | | |

