
Title:

CPU810

MICROPROGRAM

 **REGNECENTRALEN**

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 30-M 195

Edition: November 1978

Author: Aage Jørgensen

Lars Bone Jørgensen

Flemming Kam

Keywords: RC8000/15, RC8000/25, RC8000/35, CPU810.

Abstract: This manual contains:

- Definition of Microinstruction Parameters and Microinstructions.
- CPU810 Microprogram Listing
- Listing of Instruction Decoding Table.

Copyright A/S Regnecentralen, 1978
Printed by A/S Regnecentralen, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

INTRODUCTION.

This manual consists of three parts:

1. Definition of Microinstruction Parameters and Microinstructions.
2. CPU810 Microprogram Listing.
3. Instruction Decoding Table Listing.

The CPU810 microprogram is written in a symbolic assembly language and assembled by means of the RC3600 macroassembler, DOMAC. Microinstructions are 36 bits long and occupies three 16-bit RC3600 words. The control store addresses generated by the assembler are therefore multiplied with 3.

To facilitate the definition of microinstructions, they are generated by means of two macros, XVFD for arithmetic-logic microinstructions and AVFD for jump microinstructions. In the AVFD macro the address parameter of the microinstruction is divided by 3.

The listing generated by the DOMAC assembler is printed by means of a special printprogram, MCLIST, which generates a listing with correct control store addresses. A line of the microprogram listing contains 5 columns with the following information.

1. Line number
2. Control store address, octal
3. Contents of control store location, 12 octal digits.
4. Microprogram statement
5. Comments preceded by a semicolon.

The contents of a control store location is represented by 12 digits in the following way:

<u>DIGIT NO.</u>	<u>MICROINSTRUCTION BITS</u>
1	1:3
2	4:6
3	7:9
4	10:12
5	13:15
6	16:17 NB! only 2 bits
7	18:20
8	21:23
9	24:26
10	27:29
11	30:32
12	33:35

The parity bit, bit(0), of the microinstruction is not represented in the listing. It is generated by the program, which generates control tapes for the PROM programming device.

The instruction decoding table contains addresses referring to control store locations identified by labels. The contents indicated in the listing of the table must be divided by 20 (octal) to obtain the correct control store address.

```

0000 ;*****
0001 ;DEFINITION OF MICROINSTRUCTION PARAMETERS
0002 ;*****
0003

```

780814 AAJ

```

0004 ;ALU FUNCTION AND CARRY,MIR(13:17)

```

```

0006
0007     ADD=0
0008     ADDO=1
0009     ADDC=2
0010     ADDX=3
0011     SUN=4
0012     SUNO=5
0013     SUNC=6
0014     SUNX=7
0015     SUB=10
0016     SUBO=11
0017     SUBC=12
0018     SUBX=13
0019     OR=14
0020     AND=20
0021     CAND=24
0022     EXOR=30
0023     EXNOR=34

```

```

0024
0025 ;GENERAL REGISTER ADDRESSES,MIR(28:31) AND MIR(32:35)

```

```

0026
0027
0028     W0=0
0029     W1=1
0030     W2=2
0031     W3=3
0032     STAT=4
0033     IC=5
0034     CAUSE=6
0035     SB=7
0036     PC=10
0037     CNTR=11
0038     WRK0=12
0039     WRK1=13
0040     WRK2=14
0041     WRK3=15
0042     WRK4=16
0043     WRK5=17
0044

```

```

;INSTR. COUNTER FOR LOGICAL ADDRESS

```

```

;INSTR. COUNTER FOR ABSOLUTE ADDRESS
;WORKS IN CONJUNCTION WITH CONTR. OUTP. REG.

```

```

0045 ;INDIRECT ADDRESSING OF GENERAL REGISTERS
0046
0047 ; THE FIRST 4 WORDS OF THE B-PART OF THE GENERAL REGISTERS
0048 ; CAN ONLY BE INDIRECTLY ADDRESSED
0049
0050 ;GRB ADDR=W-FIELD OF INSTR REG
0051 ;GRB ADDR=W-PRE OF INSTR REG
0052 ;GRB ADDR=X-FIELD OF INSTR REG
0053 ;GRB ADDR=I/OADDR(21:22)
0054
0055
0056 ;SCRATCHPAD ADDRESSES, MIR(24:27)
0057
0058 ;COMMON PROTECTED AREA LIMIT
0059
0060 ;LOWER LIMIT
0061 ;UPPER LIMIT
0062 ;INTERRUPT LIMIT COPY
0063
0064
0065 ;MONITOR TOP
0066 ;REAL TIME CLOCK
0067 ;CONSTANT 00000002
0068 ;CONSTANT 00000003
0069 ;CONSTANT 00000377
0070 ;CONSTANT 00007777
0071 ;INTERRUPT LIMIT
0072
0073
0074
0075
0076 ;SOURCE REGISTER ADDRESSES
0077
0078 ;IMMEDIATE OPERAND
0079 ;SIGN EXTENSION
0080 ;HALF WORD EXCHANGE
0081 ;DATA IN
0082 ;INTERRUPT LEVEL
0083 ;TCP DATA IN
0084

```

```

0085 ;DESTINATION REGISTER ADDRESSES AND I/O CONTROL, MIR(18:23)
0086
0087 ; ADDRESSING OF THE I/O ADDR REG STARTS AN I/O OPERATION
0088
0089 ;O=NO LOAD
0090 ;CPU STATUS
0091 ;I/O READ
0092 ;I/O READ WITH PROTECTION
0093 ;I/O WRITE
0094 ;I/O WRITE WITH PROTECTION
0095 ;DATA OUT
0096 ;CONTROL OUTPUT REGISTER
0097 ;TCP DATA OUT
0098 ;INSTRUCTION REGISTER
0099 ;MICRO INDEX REGISTER
0100 ;INTERRUPT REGISTER
0101
0102
0103 ;JUMP CONDITION CONTROL, MIR(18)
0104
0105 F=0 ;THE COMPLEMENTED VALUE OF THE COND. IS SELECTED
0106 T=1 ;THE TRUE VALUE OF THE COND. IS SELECTED
0107
0108
0109 ;JUMP CONDITIONS, MIR(19:23)
0110
0111 FALSE=0
0112 NNEG=1 ;THIS CONDITION IS ALWAYS FALSE
0113 NZ=2 ;RESULT>=0
0114 OVFL=3 ;RESULT<>0
0115 CARRY=4 ;ARITHMETIC OVERFLOW
0116 NORM=5 ;ARITHMETIC CARRY
0117 ;RES(0)<>RES(1)
0118 ;
0119 ;
0120 ;MONITOR MODE
0121 ;ESCAPE MODE
0122 ;AFTER AM
0123 ;AFTER ESCAPE
0124 ;INTEGER MASK
0125 ;FLOATING POINT MASK
0126 ;INDIRECT ADDRESSING, IR(9)
0127 ;W-FIELD<>0
0128

```

```

0129 NMADR=20 ;NOT MEMORY ADDRESS, ADDR<8
0130 NWADR=21 ;NOT W-REG ADDRESS, 0>ADDR>=8
0131 ODD=22 ;ADDR(23)=1
0132 ;
0133 BERR=24 ;BUS ERROR
0134 BTIM=25 ;BUS TIME OUT
0135 BNACK=26 ;BUS NACK
0136 BPAR=27 ;BUS PARITY ERROR
0137 ;
0138 INTR=30 ;INTERRUPT
0139 NTPIN=31 ;NOT TCP INPUT READY
0140 TPACK=32 ;TCP OUTPUT ACK
0141 RSTRT=33 ;RESTART ENABLE
0142 SHORT=34 ;TEST MODE SWITCH IN POSITION 'SHORT'
0143 TSTON=35 ;TEST SWITCH IN POSITION 'ON'
0144 PLOW=36 ;POWER LOW (PINT)
0145 ;
0146 ;
0147 ;
0148 ;SHIFT-IN CONTROL, MIR(24:25)
0149 ;
0150 Z=0 ;ZERO
0151 LNK=1 ;SHIFT LINK
0152 ADC=2 ;ADDCOND, LEFT SHIFTS ONLY
0153 SGN=2 ;SIGN EXTENSION, RIGHT SHIFTS ONLY
0154 ;3 IS UNUSED
0155 ;
0156 ;
0157 ;TEST CONTROL, MIR(26:27)
0158 ;
0159 NL=0 ;NO LOAD
0160 MC=1 ;ADDCOND:=MULTIPLY CONDITION
0161 DC=2 ;ADDCOND:=DIVIDE CONDITION
0162 DVS=3 ;DIVSIGN:=RESULT(0)
0163 ;
0164 ;
0165 ;

```

0166 ;SEQUENCE CONTROL, MIR(1:3)

0167

0168

0169

0170

0171

0172

0173

0174

0175

0176

CONT=0

PUSH=2

POP=3

RTN=4

LRTN=5

ADDR=6

EXEC=7

;CONTINUE

;1 IS UNUSED

;PUSH AND CONTINUE

;POP AND CONTINUE

;SUBROUTINE RETURN

;LOOP RETURN, CONDITIONAL

;CALL ADDRESS CALCULATION

;CALL INSTRUCTION EXECUTION

```

0177 ;*****
0178 ;DEFINITION OF MICROINSTRUCTIONS
0179 ;*****
0180
0181
0182 ;FORMAT 0: LOAD IMMEDIATE OPERAND REGISTER
0183 ;*****
0184
0185 ; LDIM OCTAL NO.,OCTAL NO. NEXT
0186
0187 XVFD LDIM 0,2000,0,35,12,47,12,15,3
0188
0189
0190 ;FORMAT 1: ARITHMETIC-LOGIC MICROINSTRUCTIONS WITH SCRATCHPAD DESTINATION
0191 ;*****
0192
0193 ;LOAD SCRATCHPAD
0194
0195 ; AQS ALUFUNC GRA,SP NEXT
0196 ; ABS ALUFUNC GRA,GRB,SP NEXT
0197 ; ZQS ALUFUNC SP NEXT
0198 ; ZBS ALUFUNC GRB,SP NEXT
0199 ; ZAS ALUFUNC GRA,SP NEXT
0200
0201 XVFD AQS 0,22000,0,29,5,43,4,39,4,15,3
0202 XVFD ABS 0,22200,0,29,5,43,4,47,4,39,4,15,3
0203 XVFD ZQS 0,22400,0,29,5,39,4,15,3
0204 XVFD ZBS 0,22600,0,29,5,47,4,39,4,15,3
0205 XVFD ZAS 0,23000,0,29,5,43,4,39,4,15,3
0206
0207
0208 ;LOAD GRB,SCRATCHPAD
0209
0210 ; AQS ALUFUNC GRA,GRB,SP NEXT
0211 ; ABS ALUFUNC GRA,GRB,SP NEXT
0212 ; ZQS ALUFUNC GRB,SP NEXT
0213 ; ZBS ALUFUNC GRB,SP NEXT
0214 ; ZAS ALUFUNC GRA,GRB,SP NEXT
0215
0216 XVFD AQS 0,26000,0,29,5,43,4,47,4,39,4,15,3
0217 XVFD ABS 0,26200,0,29,5,43,4,47,4,39,4,15,3
0218 XVFD ZQS 0,26400,0,29,5,47,4,39,4,15,3
0219 XVFD ZBS 0,26600,0,29,5,47,4,39,4,15,3
0220 XVFD ZAS 0,27000,0,29,5,47,4,47,4,39,4,15,3
0221

```

```

0222 ;LOAD Q, SCRATCHPAD
0223
0224 ;      AQQS      ALUFUNC GRA,SP  NEXT
0225 ;      ABQS      ALUFUNC GRA,GRB,SP  NEXT
0226 ;      ZQQS      ALUFUNC SP  NEXT
0227 ;      ZBQS      ALUFUNC GRB,SP  NEXT
0228 ;      ZAQS      ALUFUNC GRA,SP  NEXT
0229
0230      XVFD AQQS 0,20000,0,29,,5,43,,4,39,,4,15,,3
0231      XVFD ABQS 0,20200,0,29,,5,43,,4,47,,4,39,,4,15,,3
0232      XVFD ZQQS 0,20400,0,29,,5,39,,4,15,,3
0233      XVFD ZHQS 0,20600,0,29,,5,47,,4,39,,4,15,,3
0234      XVFD ZAQS 0,21000,0,29,,5,43,,4,39,,4,15,,3
0235
0236
0237 ;FORMAT 2: ARITHMETIC-LOGIC MICROINSTRUCTIONS WITH SCRATCHPAD SOURCE
0238 ;*****
0239
0240
0241 ;NO LOAD
0242
0243 ;      SA      ALUFUNC SP,GRA  NEXT
0244 ;      SQ      ALUFUNC SP  NEXT
0245 ;      SZN     ALUFUNC SP  NEXT
0246
0247      XVFD SA 0,43200,0,29,,5,39,,4,43,,4,15,,3
0248      XVFD SQ 0,43400,0,29,,5,39,,4,15,,3
0249      XVFD SZN 0,43600,0,29,,5,39,,4,15,,3
0250
0251
0252 ;LOAD GRB
0253
0254 ;      SAB      ALUFUNC SP,GRA,GRB  NEXT
0255 ;      SQB      ALUFUNC SP,GRB  NEXT
0256 ;      SZB      ALUFUNC SP,GRB  NEXT
0257
0258      XVFD SAB 0,47200,0,29,,5,39,,4,43,,4,47,,4,15,,3
0259      XVFD SQB 0,47400,0,29,,5,39,,4,47,,4,15,,3
0260      XVFD SZB 0,47600,0,29,,5,39,,4,47,,4,15,,3
0261

```

```

0262 ;LOAD Q
0263
0264 ; SAG ALUFUNC SP, GRA NEXT
0265 ; SQQ ALUFUNC SP NEXT
0266 ; SZQ ALUFUNC SP NEXT
0267
0268 XVFD SAG 0,41200,0,29,,5,39,,4,43,,4,15,,3
0269 XVFD SQQ 0,41400,0,29,,5,39,,4,15,,3
0270 XVFD SZQ 0,41600,0,29,,5,39,,4,15,,3
0271
0272 ;LOAD DESTINATION REGISTER
0273
0274 ; SAR ALUFUNC SP, GRA, DR NEXT
0275 ; SQR ALUFUNC SP, DR NEXT
0276 ; SZR ALUFUNC SP, DR NEXT
0277
0278 XVFD SAR 0,43200,0,29,,5,39,,4,43,,4,35,,6,15,,3
0279 XVFD SQR 0,43400,0,29,,5,39,,4,35,,6,15,,3
0280 XVFD SZR 0,43600,0,29,,5,39,,4,35,,6,15,,3
0281
0282
0283 ;LOAD GRB, DESTINATION REGISTER
0284
0285 ; SAGR ALUFUNC SP, GRA, GRB, DR NEXT
0286 ; SQBR ALUFUNC SP, GRB, DR NEXT
0287 ; SZBR ALUFUNC SP, GRB, DR NEXT
0288
0289 XVFD SAGR 0,47200,0,29,,5,39,,4,43,,4,47,,4,35,,6,15,,3
0290 XVFD SQBR 0,47400,0,29,,5,39,,4,47,,4,35,,6,15,,3
0291 XVFD SZBR 0,47600,0,29,,5,39,,4,47,,4,35,,6,15,,3
0292
0293
0294 ;LOAD Q, DESTINATION REGISTER
0295
0296 ; SAGR ALUFUNC SP, GRA, DR NEXT
0297 ; SQBR ALUFUNC SP, DR NEXT
0298 ; SZBR ALUFUNC SP, DR NEXT
0299
0300 XVFD SAGR 0,41200,0,29,,5,39,,4,43,,4,35,,6,15,,3
0301 XVFD SQBR 0,41400,0,29,,5,39,,4,35,,6,15,,3
0302 XVFD SZBR 0,41600,0,29,,5,39,,4,35,,6,15,,3
0303

```

0304 ;FORMAT 3: ARITHMETIC-LOGIC MICROINSTRUCTIONS WITH EXT. SOURCE REGISTER AND EXT. DEST. REGISTER
 0305 ;*****
 0306 ;*****

0307 ;NO LOAD

0310 ; AQ ALUFUNC GRA NEXT
 0311 ; AB ALUFUNC GRA,GRB NEXT
 0312 ; ZQ ALUFUNC NEXT
 0313 ; ZB ALUFUNC GRB NEXT
 0314 ; ZA ALUFUNC GRA NEXT
 0315 ; RA ALUFUNC SR,GRA NEXT
 0316 ; RQ ALUFUNC SR NEXT
 0317 ; RZ ALUFUNC SR NEXT

0318 XVFD AQ 0,62000,0,29,,5,43,,4,15,,3
 0319 XVFD AB 0,62200,0,29,,5,43,,4,47,,4,15,,3
 0320 XVFD ZQ 0,62400,0,29,,5,15,,3
 0321 XVFD ZB 0,62600,0,29,,5,47,,4,15,,3
 0322 XVFD ZA 0,63000,0,29,,5,43,,4,15,,3
 0323 XVFD RA 0,63200,0,29,,5,39,,4,43,,4,15,,3
 0324 XVFD RQ 0,63400,0,29,,5,39,,4,15,,3
 0325 XVFD RZ 0,63600,0,29,,5,39,,4,15,,3

0327 ;LOAD GRB

0330 ; AQB ALUFUNC GRA,GRB NEXT
 0331 ; ABB ALUFUNC GRA,GRB NEXT
 0332 ; ZQB ALUFUNC GRB NEXT
 0333 ; ZBB ALUFUNC GRB NEXT
 0334 ; ZAB ALUFUNC GRA,GRB NEXT
 0335 ; RAB ALUFUNC SR,GRA,GRB NEXT
 0336 ; RQB ALUFUNC SR,GRB NEXT
 0337 ; RZB ALUFUNC SR,GRB NEXT
 0338 ; NEXT

0339 XVFD AQB 0,66000,0,29,,5,43,,4,47,,4,15,,3
 0340 XVFD ABB 0,66200,0,29,,5,43,,4,47,,4,15,,3
 0341 XVFD ZQB 0,66400,0,29,,5,47,,4,15,,3
 0342 XVFD ZBB 0,66600,0,29,,5,47,,4,15,,3
 0343 XVFD ZAB 0,67000,0,29,,5,43,,4,47,,4,15,,3
 0344 XVFD RAB 0,67200,0,29,,5,39,,4,43,,4,47,,4,15,,3
 0345 XVFD RQB 0,67400,0,29,,5,39,,4,47,,4,15,,3
 0346 XVFD RZB 0,67600,0,29,,5,39,,4,47,,4,15,,3
 0347 XVFD RZB 0,67600,0,29,,5,39,,4,47,,4,15,,3
 0348

```

0349 ;LOAD Q
0350
0351 ; AQQ ALUFUNC GRA NEXT
0352 ; ABQ ALUFUNC GRA,GRB NEXT
0353 ; ZQQ ALUFUNC NEXT
0354 ; ZBQ ALUFUNC GRB NEXT
0355 ; ZAQ ALUFUNC GRA NEXT
0356 ; RAQ ALUFUNC SR,GRA NEXT
0357 ; RQQ ALUFUNC SR NEXT
0358 ; RZQ ALUFUNC SR NEXT
0359
0360 XVPD AQQ 0,60000,0,29,,5,43,,4,15,,3
0361 XVPD ABQ 0,60200,0,29,,5,43,,4,47,,4,15,,3
0362 XVPD ZQQ 0,60400,0,29,,5,15,,3
0363 XVPD ZBQ 0,60600,0,29,,5,47,,4,15,,3
0364 XVPD ZAQ 0,61000,0,29,,5,43,,4,15,,3
0365 XVPD RAQ 0,61200,0,29,,5,39,,4,43,,4,15,,3
0366 XVPD RQQ 0,61400,0,29,,5,39,,4,15,,3
0367 XVPD RZQ 0,61600,0,29,,5,39,,4,15,,3
0368
0369
0370 ;LOAD DEST. REG.
0371
0372 ; AQR ALUFUNC GRA,DR NEXT
0373 ; ABR ALUFUNC GRA,GRB,DR NEXT
0374 ; ZQR ALUFUNC DR NEXT
0375 ; ZBR ALUFUNC GRB,DR NEXT
0376 ; ZAR ALUFUNC GRA,DR NEXT
0377 ; RAR ALUFUNC SR,GRA,DR NEXT
0378 ; RQR ALUFUNC SR,DR NEXT
0379 ; RZR ALUFUNC SR,DR NEXT
0380
0381 XVPD AQR 0,62000,0,29,,5,43,,4,35,,6,15,,3
0382 XVPD ABR 0,62200,0,29,,5,43,,4,47,,4,35,,6,15,,3
0383 XVPD ZQR 0,62400,0,29,,5,35,,6,15,,3
0384 XVPD ZBR 0,62600,0,29,,5,47,,4,35,,6,15,,3
0385 XVPD ZAR 0,63000,0,29,,5,43,,4,35,,6,15,,3
0386 XVPD RAR 0,63200,0,29,,5,39,,4,43,,4,35,,6,15,,3
0387 XVPD RQR 0,63400,0,29,,5,39,,4,35,,6,15,,3
0388 XVPD RZR 0,63600,0,29,,5,39,,4,35,,6,15,,3
0389

```

```

0390 ;LOAD GRB, DEST. REG.
0391
0392 ; AQBR ALUFUNC GRA,GRB,DR NEXT
0393 ; ABBR ALUFUNC GRA,GRB,DR NEXT
0394 ; ZQBR ALUFUNC GRB,DR NEXT
0395 ; ZBBR ALUFUNC GRB,DR NEXT
0396 ; ZABR ALUFUNC GRA,GRB,DR NEXT
0397 ; RABR ALUFUNC SR,GRA,GRB,DR NEXT
0398 ; RQBR ALUFUNC SR,GRB,DR NEXT
0399 ; RZBR ALUFUNC SR,GRB,DR NEXT
0400
0401 XVFD AQBR 0,66000,0,29,5,43,4,47,4,35,6,15,3
0402 XVFD ABBR 0,66200,0,29,5,43,4,47,4,35,6,15,3
0403 XVFD ZQBR 0,66400,0,29,5,47,4,35,6,15,3
0404 XVFD ZBBR 0,66600,0,29,5,47,4,35,6,15,3
0405 XVFD ZABR 0,67000,0,29,5,43,4,47,4,35,6,15,3
0406 XVFD RABR 0,67200,0,29,5,39,4,43,4,47,4,35,6,15,3
0407 XVFD RQBR 0,67400,0,29,5,39,4,47,4,35,6,15,3
0408 XVFD RZBR 0,67600,0,29,5,39,4,47,4,35,6,15,3
0409
0410
0411 ;LOAD Q, DEST. REG.
0412
0413 ; AQBR ALUFUNC GRA,DR NEXT
0414 ; ABQR ALUFUNC GRA,GRB,DR NEXT
0415 ; ZQQR ALUFUNC DR NEXT
0416 ; ZBQR ALUFUNC GRB,DR NEXT
0417 ; ZAQR ALUFUNC GRA,DR NEXT
0418 ; RAQR ALUFUNC SR,GRA,DR NEXT
0419 ; RQQR ALUFUNC SR,DR NEXT
0420 ; RZQR ALUFUNC SR,DR NEXT
0421
0422 XVFD AQQR 0,60000,0,29,5,43,4,35,6,15,3
0423 XVFD ABQR 0,60200,0,29,5,43,4,47,4,35,6,15,3
0424 XVFD ZQQR 0,60400,0,29,5,35,6,15,3
0425 XVFD ZBQR 0,60600,0,29,5,47,4,35,6,15,3
0426 XVFD ZAQR 0,61000,0,29,5,43,4,35,6,15,3
0427 XVFD RAQR 0,61200,0,29,5,39,4,43,4,35,6,15,3
0428 XVFD RQQR 0,61400,0,29,5,39,4,35,6,15,3
0429 XVFD RZQR 0,61600,0,29,5,39,4,35,6,15,3
0430

```

```

0431 ;SIGN EXTENSION
0432
0433 ;      EXT      GRA,GRB NEXT      ;GRB:=12 EXT GRA(12) CON GRA(12:23)
0434
0435      XVFD EXT 0,65660,400,43.,4,47.,4,15.,3
0436
0437
0438 ;LOAD DEST. REG. & EXTEND SIGN
0439
0440 ;      EXTR      GRA,GRB,DR      NEXT      ;DR:=GRA
0441 ;      ;GRB:=12 EXT GRA(12) CON GRA(12:23)
0442
0443      XVFD EXTR 0,65660,400,43.,4,47.,4,35.,6,15.,3
0444
0445
0446 ;EXCHANGE HALF-WORDS
0447
0448 ;      SWAP      GRA,GRB NEXT      ;GRB:=GRA(12:23) CON GRA(0:11)
0449
0450      XVFD SWAP 0,65660,1000,43.,4,47.,4,15.,3
0451
0452
0453 ;CLEAR INTERRUPT BIT
0454
0455 ;      CLIN      GRA      NEXT      ;INTRG(GRA):=0
0456
0457      XVFD CLIN 0,63060,100000,43.,4,15.,3
0458

```

```

0459 ;FORMAT 4: SHIFT MICROINSTRUCTIONS
0460 ;*****
0461
0462
0463 ;SHIFT LEFT GRB
0464
0465 ;      SSLA      ALUFUNC,GRB,SI,TST  NEXT
0466 ;      SSLB      ALUFUNC,GRB,SI,TST  NEXT
0467
0468      XVFD SSLA 0,116200,0,29,,5,43,,4,47,,4,37,,2,39,,2,15,,3
0469      XVFD SSLB 0,116600,0,29,,5,47,,4,37,,2,39,,2,15,,3
0470
0471
0472 ;SHIFT LEFT GRB CON Q
0473
0474 ;      DSLA      ALUFUNC,GRB,SI,TST  NEXT
0475 ;      DSLB      ALUFUNC,GRB,SI,TST  NEXT
0476
0477      XVFD DSLA 0,114200,0,29,,5,43,,4,47,,4,37,,2,39,,2,15,,3
0478      XVFD DSLB 0,114600,0,29,,5,47,,4,37,,2,39,,2,15,,3
0479
0480
0481 ;SHIFT RIGHT GRB
0482
0483 ;      SSRA      ALUFUNC,GRB,SI,TST  NEXT
0484 ;      SSRB      ALUFUNC,GRB,SI,TST  NEXT
0485
0486      XVFD SSRA 0,112200,0,29,,5,43,,4,47,,4,37,,2,39,,2,15,,3
0487      XVFD SSRB 0,112600,0,29,,5,47,,4,37,,2,39,,2,15,,3
0488
0489
0490 ;SHIFT RIGHT GRB CON Q
0491
0492 ;      DSRA      ALUFUNC,GRB,SI,TST  NEXT
0493 ;      DSRB      ALUFUNC,GRB,SI,TST  NEXT
0494
0495      XVFD DSRA 0,110200,0,29,,5,43,,4,47,,4,37,,2,39,,2,15,,3
0496      XVFD DSRB 0,110600,0,29,,5,47,,4,37,,2,39,,2,15,,3
0497

```

```

0498 ;SHIFT LEFT GRB, CONDITION TEST
0499
0500 ; SLAB ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0501 ; SLZB ALUFUNC, GRB, SI, TST, T/F, COND NEXT
0502 ; SLZA ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0503
0504 XVFD SLAB 0,116200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0505 XVFD SLZB 0,116600,0,29,,5,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0506 XVFD SLZA 0,117000,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0507
0508
0509 ;SHIFT LEFT GRB CON Q, CODITION TEST
0510
0511 ; DLAB ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0512 ; DLZB ALUFUNC, GRB, SI, TST, T/F, COND NEXT
0513 ; DLZA ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0514
0515 XVFD DLAB 0,114200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0516 XVFD DLZB 0,114600,0,29,,5,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0517 XVFD DLZA 0,115000,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0518
0519
0520 ;SHIFT RIGHT GRB, CONDITION TEST
0521
0522 ; SRAB ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0523 ; SRZB ALUFUNC, GRB, SI, TST, T/F, COND, NEXT
0524 ; SRZA ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0525
0526 XVFD SRAB 0,112200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0527 XVFD SRZB 0,112600,0,29,,5,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0528 XVFD SRZA 0,113000,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0529
0530
0531 ;SHIFT RIGHT GRB CON Q, CONDITION TEST
0532
0533 ; DRAB ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0534 ; DRZB ALUFUNC, GRB, SI, TST, T/F, COND NEXT
0535 ; DRZA ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT
0536
0537 XVFD DRAB 0,110200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0538 XVFD DRZB 0,110600,0,29,,5,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0539 XVFD DRZA 0,111000,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0540
0541
0542

```

```

0543 ;NO SHIFT, LOAD GRB, CONDITION TEST
0544
0545 ;      NSAB      ALUFUNC, GRA, GRB, TST, T/F, COND      NEXT
0546 ;      NSZB      ALUFUNC, GRB, TST, T/F, COND      NEXT
0547 ;      NSZA      ALUFUNC, GRA, GRB, TST, T/F, COND      NEXT
0548
0549      XVFD NSAB 0,106200,0,29,,5,43,,4,47,,4,39,,2,30,,1,35,,5,15,,3
0550      XVFD NSZB 0,106600,0,29,,5,47,,4,39,,2,30,,1,35,,5,15,,3
0551      XVFD NSZA 0,107000,0,29,,5,43,,4,47,,4,39,,2,30,,1,35,,5,15,,3
0552

```

```

0553 ;FORMAT 5: MULTIPLY MICROINSTRUCTIONS
0554 ;*****
0555 ;*****
0556 ;*****
0557 ;      ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT      ;MULTIPLY, DOUBLE SHIFT
0558 ;      ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT      ;MULTIPLY, SINGLE SHIFT
0559 ;      ALUFUNC, GRA, GRB, TST, T/F, COND NEXT          ;MULTIPLY, NO SHIFT
0560
0561 XVPD MULD 0,130200,0,29.,5,43.,4,47.,4,37.,2,39.,2,30.,1,35.,5,15.,3
0562 XVPD MULS 0,132200,0,29.,5,43.,4,47.,4,37.,2,39.,2,30.,1,35.,5,15.,3
0563 XVPD MULN 0,126200,0,29.,5,43.,4,47.,4,39.,2,30.,1,35.,5,15.,3
0564

```

```

0565 ;FORMAT 6: DIVIDE MICROINSTRUCTIONS
0566 ;*****
0567 ;*****
0568 ;
0569 ;      ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT      ;DIVIDE, DOUBLE SHIFT
0570 ;      ALUFUNC, GRA, GRB, SI, TST, T/F, COND NEXT      ;DIVIDE, SINGLE SHIFT
0571 ;      ALUFUNC, GRA, GRB, TST, T/F, COND NEXT          ;DIVIDE, NO SHIFT
0572
0573      XVPD DIVD 0,154200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0574      XVPD DIVS 0,156200,0,29,,5,43,,4,47,,4,37,,2,39,,2,30,,1,35,,5,15,,3
0575      XVPD DIVN 0,146200,0,29,,5,43,,4,47,,4,39,,2,30,,1,35,,5,15,,3
0576

```

0577 ;FORMAT 7: JUMP MICROINSTRUCTIONS
0578 ;*****
0579 ;*****
0580 ;*****

0581 ;CONDITIONAL JUMPS

0582 ; JMP T/F,COND,ADDR NEXT
0583 ; JMX T/F,COND NEXT

0584 ;
0585 AVFD JMP 0,162000,0,30,,1,35,,5,47,,12,,15,,3
0586 AVFD JMX 0,162004,0,30,,1,35,,5,15,,3

0587 ;
0588 ;
0589 ;
0590 ;CONDITIONAL SUBROUTINE CALLS

0591 ; CAL T/F,COND,ADDR NEXT
0592 ; CAX T/F,COND NEXT

0593 ;
0594 AVFD CAL 0,162010,0,30,,1,35,,5,47,,12,,15,,3
0595 AVFD CAX 0,162014,0,30,,1,35,,5,15,,3

0596 ;
0597 ;
0598 ;
0599 ;CONDITIONAL JUMPS WITH I/O SYNCHRONIZATION

0600 ;
0601 ; WJMP T/F,COND,ADDR NEXT
0602 ; WJMX T/F,COND NEXT

0603 ;
0604 AVFD WJMP 0,162020,0,30,,1,35,,5,47,,12,,15,,3
0605 AVFD WJMX 0,162024,0,30,,1,35,,5,15,,3

0606 ;
0607 ;
0608 ;CONDITIONAL SUBROUTINE CALLS WITH I/O SYNCHRONIZATION

0609 ;
0610 ; WCAL T/F,COND,ADDR NEXT
0611 ; WCAX T/F,COND NEXT

0612 ;
0613 AVFD WCAL 0,162030,0,30,,1,35,,5,47,,12,,15,,3
0614 AVFD WCAX 0,162034,0,30,,1,35,,5,15,,3

0615

0001 CPU810 MICROPROGRAM, REVISION: 1, 781113 AAJ

01 000001

02 .NOMA 1

03 ;INTERRUPT JUMP TABLE

04 ;*****

05 ; THE FIRST 16 LOCATIONS CONTAIN A JUMP TABLE
06 ; WITH JUMPS TO THE INTERRUPT SERVICE ROUTINES

07 08 0000 0710000000020

09 0001 0710000002404

10 0002 0710000002663

11 0003 0710000000062

12 0004 0710000000070

13 0005 0710000000176

14 0006 0710000000207

15 0007 0710000000207

16 17 0010 0710000000207

18 0011 0710000000207

19 0012 0710000000207

20 0013 0710000000207

21 0014 0710000000207

22 0015 0710000000207

23 0016 0710000000207

24 0017 0710000000207

JMP F,F,POWUP

JMP F,F,TCPIN

JMP F,F,SINGL

JMP F,F,OCPAL

JMP F,F,REMAI

JMP F,F,RTINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

JMP F,F,EXINT

;POWER ON

;INPUT FROM TCP

;SINGLE INSTRUCTION

;OCP AUTOLoad

;REMOTE AUTOLoad

;0.1 MILLISECOND TIMER

;POWER LOW

;INTERVAL TIMER

;LOCATIONS 10 TO 17 ARE USED

;FOR DEVICE INTERRUPTS

;

;

;

;

;

;

```

0002
01
02
03
04 0020 033340040011
05 0021 071002000214
06
07
08
09 0022 001000000010
10 0023 030730000000
11 0024 011220001000
12
13 0025 001073777777
14 0026 030730000000
15 0027 011220001400
16
17 0030 001100000000
18 0031 033730010004
19
20 0032 071000760032
21 0033 001012000000
22 0034 071002000240
23 0035 071002000233
24
25 0036 071000330046
26 0037 001000000012
27 0040 031730220000
28 0041 031230000000
29 0042 071010640051
30 0043 033730001405
31 0044 033730001410
32 0045 0710000000243
33
34 0046 0010000000100
35 0047 033530040231
36 0050 071000330050

;POWER UP
;*****
POWUP:  ZBRR      AND,CNTR,CNTR0
        CAL       F,F,INITR

        LDIM      0,10
        RZQ       OR,IMOP
        ZQS       SUB,LLIM

        LDIM      3777,7777
        RZQ       OR,IMOP
        ZQS       SUB,ULIM

        LDIM      4000,0
        RZBR      OR,IMOP,STAI,CPUST

        JMP       T,PLOW,WL1
        LDIM      600,0
        CAL       F,F,DELAY
        CAL       F,F,CINTR

        JMP       F,RSTRT,NREST
        LDIM      0,12
        RZR       OR,IMOP,READP
        ZQ        OR
        WJMP      T,BERR,SYFLT
        RZB       OR,DATI,IC
        RZB       OR,DATI,PC
        JMP       F,F,MLOOP

        LDIM      0,100
        RABR      OR,IMOP,CNTR,CNTR0
        JMP       F,RSTRT,WL2

        NREST:
        WL2:

```

```

;CNTR:=CNTR0:=0
;BASE:=ILIMC:=ILIM:=RTC:=0
;INF:=1, SP16:=SP17:=-1
;C2:=2, M2:=3, M8:=377, M12:=7777

```

```

;LLIM:=10

```

```

;ULIM:=37777777

```

```

;STAI:=CPUST:=40000000 (MMODE:=1)

```

```

;WAIT UNTIL NOT POWER LOW
;WAIT 630 MS
;FOR MEMORY AND TCP POWER ON
;CLEAR ALL INTERRUPTS

```

```

;IF AUTOLOAD PB IS ON THEN GOTO NREST

```

```

;FETCH IC
;NOOP

```

```

;WAIT, IF BUSERROR THEN GOTO SYSTEM FAULT

```

```

;PC:=IC:=MEM(12)
;GOTO MLOOP

```

```

;RUN:=1
;WAIT UNTIL AUTOLOAD PB IS RELEASED

```

!0003

01 ;IDLE LOOP AND SYSTEM FAULT
02 ;*****
03 ;*****

04 SYFLT:
05 IDLE:
06 ILOOP:

07 0051 033340040011
08 0052 071000300052
09 0053 033730002012
10 0054 001000000377
11 0055 033540070252
12 0056 001000000004
13 0057 031510000240
14 0060 071000410052
15 0061 071001000000

AND,CNTR,CNTR0
F,INTR,ILOOP
OR,ILEV,WRKO
O,377
AND,IMOP,WRKO,WRKO,MIX
O,4
SUN,IMOP,WRKO
T,NNEG,ILOOP
F,F

ZBBR
JMP
RZB
LDIM
RABR
LDIM
RA
JMP
JMX

;CNTR:=CNTR0:=0, (RUN:=0)
;WAIT FOR INTERRUPT
;WRKO:=ILEV
;MIX:=WRKO:=12 EXT 0 CON WRKO(16:23)
;-4+WRKO-1
;IF ILEV>4 THEN GOTO ILOOP
; ELSE GOTO MIX
;

```

!0004
01 ;AUTOLOAD SEQUENCES
02 ;*****
03 ;
04 ; AUTOLOAD OF THE CPU810 MAY BE ACTIVATED FROM THE AUTOLOAD
05 ; PUSH-BUTTON ON THE OCP OR BY MEANS OF THE REMOTE AUTOLOAD
06 ; SIGNAL.
07 ;
08 ;
09 ;OCP AUTOLOAD
10 ;
11 ; IF THE SWITCH "TEST" ON THE CPU810 PCBA FRONT PANEL IS IN
12 ; "ON" POSITION, THE MICRODIAGNOSTIC ROUTINES FOR INITIAL TEST
13 ; OF CPU AND MAIN MEMORY WILL BE RUN BEFORE AUTOLOAD.
14 ;
15 0062 00100000000040 OCPAL: LDIM 0,40 ;CNTR:=CNTR:=40, AUTOLOADING:=1
16 0063 033730040011 RZBR OR,IMOP,CNTR,CNTR ;WAIT UNTIL AUTOLOAD PB IS RELEASED
17 0064 071000330064 JMP F,RSTRT,.+0 ;IF TEST ON THEN GOTO AENTR
18 0065 071000753023 JMP T,ISTON,AENTR
19 ;
20 ; THE MICRODIAGNOSTIC RETURNS HERE
21 ;
22 0066 001173777754 AUTO1: LDIM 7777,7754 ;IMOP:=14-4*10, DEV BASE FOR DISC
23 0067 071000000073 JMP F,F,AUTOL ;GOTO COMMON AUTOLOAD SEQUENCE
24 ;
25 ;
26 ;REMOTE AUTOLOAD
27 ;
28 0070 0010000000004 REMAL: LDIM 0,4
29 0071 031730100000 RZR OR,IMOP,INTRG ;CLEAR REM AUTOLOAD INTERRUPT
30 0072 001173777774 LDIM 7777,7774 ;IMOP:=14-2*10, DEV BASE FOR FPA

```

10005

;COMMON AUTOLOAD SEQUENCE

01					
02					
03	0073	0337300000014	AUTOL:	RZB	OR,IMOP,WRK2
04	0074	0010000000200		LDIM	0,200
05	0075	0337300400011		RZBR	OR,IMOP,CNTR,CNTR0
06	0076	0010000000027		LDIM	0,27
07	0077	0710020000240		CAL	F,F,DELAY
08	0100	0333400400011		ZBBR	AND,CNTR,CNTR0
09	0101	0010000000373		LDIM	0,373
10	0102	0710020000240		CAL	F,F,DELAY
11					
12	0103	0710020000214		CAL	F,F,INITR
13					
14					
15	0104	0333400100004		ZBBR	AND,STAT,CPUST
16	0105	0010000001026		LDIM	0,1026
17	0106	0307300000000		RZQ	OR,IMOP
18	0107	0112200000000		ZQS	SUB,CPA
19	0110	0112200001000		ZQS	SUB,LLIM
20	0111	0112200001400		ZQS	SUB,ULIM
21	0112	023611004405		SQB	SUNO,C2,IC
22	0113	0010000001015		LDIM	0,1015
23	0114	0307300000000		RZQ	OR,IMOP
24	0115	0112200002400		ZQS	SUB,INF
25	0116	001073777776		LDIM	3777,7776
26	0117	0307300000000		RZQ	OR,IMOP
27	0120	0112200003000		ZQS	SUB,SIZE
28	0121	0010000003777		LDIM	0,3777
29	0122	0307300000000		RZQ	OR,IMOP
30	0123	0112200003400		ZQS	SUB,MONT
31					
32	0124	0313300300014		ZBR	OR,WRK2,DATO
33	0125	0010000000010		LDIM	0,10
34	0126	0337304200013		RZBR	OR,IMOP,WRK1,WRT
35	0127	0307300000000		RZQ	OR,IMOP
36	0130	0010000000167		LDIM	0,BOOTL/3
37	0131	0337300700012		RZBR	OR,IMOP,WRK0,MIX
38	0132	07100000000137		JMP	F,F,AUTO3

```

;WRK2:=DEV BASE
;CPU SYSTEM RESET:=1
;WAIT 9.4 MICROSEC
;CPU SYSTEM RESET:=0
;WAIT 100 MICROSEC FOR RESET OF DEVICES
;BASE:=ILIMC:=ILIM:=RTC:=0
;INF:=1, SP16:=SP17:=-1
;C2:=2, M2:=3, M8:=377, M12:=7777
;STAT:=CPUST:=0

;CPA:=LLIM:=ULIM:=22.+512.=TOP OF LOAD
;IC:=22.+512.-2 =LAST OF LOAD

;INF:=22.+512.-9

;SIZE:=37777776 =MAX
;MONITOR TOP:=3777
;ONLY MONITOR CALL 0 IS ALLOWED

;DATAOUT:=DEV BASE
;WRK1:=10, MEM(10):=DEV BASE
;Q:=8.
;MIX:=WRK0:=BOOTL
;GOTO AUTO3

```

```

0006
001 0133 071003000000
002 0134 031730030000
003 0135 023500424673
004 0136 033301070012
005 0137 030210000000
006 0140 071010640051
007 0141 071000420133
008
009 0142 001031100000
010 0143 031730030000
011 0144 033430420130
012 0145 071002000233
013 0146 071010640051
014
015 0147 001100000014
016 0150 033521420314
017 0151 001100000040
018 0152 071010240243
019
020
021 0153 031521000300
022 0154 071000420051
023
024
025
026
027 0155 001173777764
028 0156 031730030000
029 0157 001000000010
030 0160 031730420000
031 0161 001100000030
032 0162 071010640051
033 0163 031730420000
034 0164 031230000000
035 0165 071010640051
036 0166 071000000051
037
;IMOP:=CS(WRK0)
;WRK1:=WRK1+2, MEM(WRK1):=CS(WRK0)
;MIX:=WRK0:=WRK0+1
;Q:=Q-1
;WAIT, IF BUSERROR THEN GOTO SYFLT
;IF Q<>0 THEN GOTO AUTO2
;PC:=IC, MEM(IC):=JL. 0
;CLEAR ALL INTERRUPTS
;WAIT, IF BUSERROR THEN GOTO SYFLT
;I/O ADDR:=WRK2:=40000000+DEV NO*10
;START DEVICE
;WAIT,
;IF NOT BUSERROR THEN GOTO MLOOP
;IF WRK2<>DISC ADDR
;THEN GOTO SYFLT
;MEM(10):=14-3*10,
;DEV BASE FOR FPA TRANSMITTER
;WAIT, IF BUSERROR THEN GOTO SYFLT
;START FPA TRANSMITTER
;NOOP
;WAIT, IF BUSERROR THEN GOTO SYFLT
;GOTO IDLE

```

```

; IF THERE IS NO RESPONSE FROM THE DISC
; THEN TRY THE FPA TRANSMITTER

```

[illegible]

```

0008
01 ;REAL TIME CLOCK INTERRUPT
02 ;*****
03 ;*****
04 ; RTC IS INCREMENTED BY 1 EVERY 0.1 MILLISECOND.
05 ; EVERY 25.6 MILLISECONDS THE INTERVAL TIMER INTERRUPT IS SET.
06 ; AT ENTRY: WRKO=ILEV
07
08 0176 031430100240 RTINT: CLIN WRKO ;CLEAR 0.1 MS TIMER INTR
09 0177 020701004000 SZQ ADDO,RTC
10 0200 011220004000 ZQS SUB,RTC
11 0201 021640005400 SQ AND,M8
12 0202 071000420246 JMP T,NZ,MLOP1
13
14 0203 001000000010 LDIM 0,10
15 0204 033530040231 RABR OR,IMOP,CNTR,CNTR,CNTR,CNTR
16 0205 033550040231 KABR CAND,IMOP,CNTR,CNTR,CNTR,CNTR
17 0206 071000000246 JMP F,F,MLOP1
;SET IVTIMER:=1
;SET IVTIMER:=0
;GOTO MLOP1 (MAIN LOOP)

```

!0009

;EXTERNAL INTERRUPT
;*****

; AT ENTRY: WRKO=ILEV

06 0207 021521006640
07 0210 071000010246
08 0211 031430100240
09 0212 047430000246
10 0213 071000001762

EXINT: SA SUBO,ILIM,WRKO
JMP F,NNEG,MLOP1
CLIN WRKO
SLZA OR,WRKO,CAUSE,Z,NL,F,F
JMP F,F,ICALL

;IF ILEV>ILIM
;THEN GOTO MLOP1 (MAIN LOOP)
;CLEAR INTERRUPT(WRKO)
;CAUSE:=2*ILEV
;GOTO INTERRUPT ROUTINE

```

10010
01
02
03 0214 010120000652
04 0215 011230002000
05 0216 011230006400
06 0217 011230004000
07 0220 010210002400
08 0221 010210004400
09 0222 011210005000
10 0223 011240007000
11 0224 011240007400
12
13 0225 001000000377
14 0226 030730000000
15 0227 011220005400
16
17 0230 001000000777
18 0231 030730000000
19 0232 411220006000
20
21
22
23
24 0233 001000000017
25 0234 033730000012
26 0235 031430100240
27 0236 033310000012
28 0237 471000420235
29
30
31
32
33 0240 033730000012
34 0241 033310000012
35 0242 471000420241

;SUBROUTINE INITIALIZE REGISTERS AND CONSTANTS
INITR:  ARQS      SUB,WRKO,WRKO,BASE
        ZQS       OR,ILIMC
        ZQS       OR,ILIM
        ZQS       OR,RTC
        ZQQS      SUN,INF
        ZQQS      SUN,C2
        ZQS       SUN,M2
        ZQS       AND,SP16
        ZQS       AND,SP17
;BASE:=0, Q:=-1
;ILIMC:=0
;ILIM:=0
;RTC:=0
;INF:=1, Q:=-2
;C2:=2, Q:=-3
;M2:=3
;SP16:=-1
;SP17:=-1
;M8:=377
;M12:=7777
RTN

;SUBROUTINE CLEAR INTERRUPTS
CINTR:  LDIM      0,17
        RZB      OR,IMOP,WRKO
        CLIN     WRKO
        ZBB      SUN,WRKO
        JMP      T,NZ,CLEAR
RTN

;SUBROUTINE DELAY
DELAY:  RZB      OR,IMOP,WRKO
        ZBB      SUN,WRKO
        JMP      T,NZ,DELLP
RTN

```

```

!0011
01      -EOT
02      ;MAINLOOP
03      ;*****
04
05      ; THE MAINLOOP FETCHES AN INSTRUCTION, TESTS FOR INTERRUPT AND BUSERRORS
06      ; DURING FETCH, AND CALLS SUBROUTINES FOR ADDRESS CALCULATION AND
07      ; INSTRUCTION EXECUTION
08
09      MLOOP: LDIM      0,100
10             RABR      OR,IMOP,CNTR,CNTR,CNTR
11             ZAR      OR,PC,READP
12
13             JMP      T,NMADR,NMEM
14             WJMP     T,BERR,FINT
15             RZB      OR,DATI,WRKO
16
17             EXTR      WRKO,WRKO,IR
18
19             JMP      T,AFESC,AFTES      ADDR
20
21             SAB      ADD,C2,IC,IC
22             JMP      T,INDIR,INADR      EXEC
23
24             SABR     ADD,C2,PC,PC,READP
25             JMP      T,INTR,INTT
26             JMP      F,F,MLOP1
27
28
29      ;INTERRUPT
30
31      INTT:  RZB      OR,ILEV,WRKO
32             SABR     AND,M8,WRKO,WRKO,MIX
33             JMX      F,FALSE

```

```

; RUN:=1
; I/O ADDR:=PC, READP
; IF NOT MEM ADDRESS THEN GOTO NMEM
; WAIT, IF BUSERROR THEN GOTO FINT
; WRKO:= DATAIN
; IR:=WRKU(0:11),
; WRKO:=12 EXT WRKO(12) CON WRKO(12:23)
; IF AFTER ESCAPE THEN GOTO AFTES
; ELSE CALL ADDR CALCULATION
; IC:= IC + 2
; IF INDIRECT THEN GOTO INADR
; ELSE CALL INSTR EXECUTION
; PC:= PC + 2, READP
; IF INTERRUPT THEN GOTO INTT
; GOTO MLOP1
; WRKO:= ILEV, BITS(0:15) IS UNDEFINED
; MIX:= WRKO:= 16 EXT 0 CON WRKO(16:23)
; GOTO (MIX)

```

10012

;NOT MEMORY ADDRESS

; PC<8, IN CASE OF NEGATIVE ADDRESS IC AND PC IS CLEARED,
; IC AND PC COUNTS MODULO 4194302

NMEM: JMP T,NWADR,NGADR ; IF NOT W-ADDRESS THEN GOTO NGADR
ZBQ OR,GRX ; Q:= W(I/O ADDR)
ZQB OR,WRKO ; WRKO:= Q, WRKO = INSTRUCTION
JMP F,FALSE,AFTCH ; GOTO AFTCH
ZBB AND,IC ; IC:=0
ZBBR AND,PC,READP ; I/O ADDR:=PC:=0, READP
JMP F,FALSE,MLOOP ; GOTO MLOOP

;AFTER ESCAPE

AFTES: JMP T,AFAM,INESC ; IF AFTER AM THEN GOTO INESC
JMP T,EMODE,ESCM ; IF ESCAPE MODE THEN GOTO ESCM
LDIM 0400,0000 ; IMOP:= 0400 0000
RABR CAND,IMOP,STAT,STAT,CPUST ; AFTER ESCAPE:= 0
SAB ADD,C2,IC,IC EXEC ; IC:= IC + 2, CALL INSIR. EXECUTION
JMP F,FALSE,FETCH ; GOTO FETCH

;AFTER ESCAPE IN INDIRECT

INESC: LDIM 0400,0000 ; AFTER ESC:= 0
RABR CAND,IMOP,STAT,STAT,CPUST ; IC:=IC+2
SAB ADD,C2,IC,IC ; GOTO INAD1
JMP F,FALSE,INAD1

0300 001010000000
0301 033550010104
0302 023500004525
0303 071000000331

0272 071000520300
0273 071000510276
0274 001010000000
0275 033550010104
0276 723500004525
0277 071000000255

0263 071000610267
0264 030330000003
0265 033230000012
0266 071000000251
0267 033340000005
0270 033340220010
0271 071000000243

13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

!0013

```
01 ;ADDRESS CALCULATION SUBROUTINES
02 ;*****
03 ;
04 ; THE ADDRESS CALCULATIONS ARE PERFORMED BY SUBROUTINES
05 ; WHICH ARE CALLED VIA A SEPARATE ADDRESS TABLE
06 ; INPUTS TO THE TABLE IS: AFTERAM, IR(8), IR(10,11)
07 ; WRKO = EXTENDED DISPLACEMENT
```

```

!0014
01 ;AFTER AM = 0
02 ;*****
03
04 0304 4334300000247 DIR: ZAB OR,WRKO,SB RTN ; SB:= WRKO,RETURN
05
06 0305 0334300000247 REL: ZAB OR,WRKO,SB ; SB:= WRKO
07 0306 4331000000127 ABB ADD,IC,SB RTN ; SR:= SB + IC, RETURN
08
09 0307 0301000000242 INDEX: ABQ ADD,WRKO,X ; Q:= WRKO + X
10 0310 4332300000007 ZQB OR,SB RTN ; SB:= Q, RETURN
11
12 0311 0301000000242 RELX: ARQ ADD,WRKO,X ; Q:= WRKO + X
13 0312 4330000000127 AQB ADD,IC,SB RTN ; SR:= IC + Q, RETURN
14
15
16
17 ;AFTER AM = 1
18 ;*****
19 0313 0331000000127 AMR: ABB ADD,IC,SB ; SB:= SB + IC
20 0314 0331000000247 AMD: ABB ADD,WRKO,SB ; SB:= SB + WRKO
21 0315 0010200000000 LDIM 1000,0000 ; IMOP:= 10000000
22 0316 433550010104 RABR CAND,IMOP,STAT,STAT,CPUST RTN ; AFTER AM:= 0, RETURN
23
24 0317 0331000000127 AMRX: ABB ADD,IC,SB ; SB:=SB+IC
25 0320 0301000000242 AMX: ABQ ADD,WRKO,X ; Q:=X+WRKO
26 0321 0330000000167 AQB ADD,SB,SB ; SB:=Q+SB
27 0322 0010200000000 LDIM 1000,0
28 0323 433550010104 RABR CAND,IMOP,STAT,STAT,CPUST RTN ;AFTER AM:=0, RETURN
29

```

```

!0015
01
02
03
04 0324 0010200000000
05 0325 033530010104
06 0326 071000110331
07 0327 001000010002
08 0330 071002002226
09
10 0331 071002000340
11 0332 001020000000
12 0333 033550010104
13 0334 733430000247
14
15 0335 023500224610
16 0336 071000700260
17 0337 071000000246
18

;INDIRECT ADDRESSING
;*****
INADR:  LDIM 1000,0
        RABR OR,IMOP,STAT,STAT,CPUST
        JMP F,EMODE,INAD1
        LDIM 1,2
        CAL F,F,ESCAP

INAD1:  CAL F,F,GETOP
        LDIM 1000,0
        RABR CAND,IMOP,STAT,STAT,CPUST
        ZAB OR,WRKO,SB EXEC
        SABR ADD,C2,PC,PC,READP
        JMP T,INTR,INTT
        JMP F,F,MLOP1

;AFTERAM:=1
;IF NOT ESC MODE THEN GO10 INAD1
;CALL ESCAPE ROUTINE
;WRKO:=ADDRESS
;AFTER AM:=0
;SB:=ADDRESS,
;CALL INSTR EXECUTION
;I/O ADDR:=PC+2, READP
;IF INTERRUPT THEN GOTO INTT
;GOTO MLOP1 (MAIN LOOP)

```

```

!0016
01 ;SUBROUTINE GET OPERAND
02 ;*****
03 ;
04 ;   FETCHES AN OPERAND FROM THE MEMORY LOCATION ADDRESSED BY
05 ;   SB + BASE, IF ADDRESS IS WITHIN UPPER AND LOWER LIMIT OR FROM
06 ;   LOCATION ADDRESSED BY SB, IF ADDRESS IS WITHIN CPA AREA.
07 ;   RETURNS WITH WRKO = FETCHED OPERAND
08 ;   WRK1 = PHYSICAL ADDRESS
09 ;   SB = LOGICAL ADDRESS
10 ;
11 ;   ADD,BASE,SB,WRK1,READP
12 ;   T,NMADR,NOTM
13 ;   SUNO,LLIM,WRK1
14 ;   F,NNEG,GTCPA
15 ;   SUB,ULIM,WRK1
16 ;   F,NNEG,GTCPA
17 ;   T,BERR,INTB
18 ;   OR,DATI,WRKO      RTN
19 ;
20 ;   F,FALSE,NOTM
21 ;   OR,SB,WRK1,READP
22 ;   F,NWADR,GTREG
23 ;   T,NMADR,INTO
24 ;   SUB,CPA,SB
25 ;   F,NNEG,INTO
26 ;   T,BERR,INTB
27 ;   OR,DATI,WRKO      RTN
28 ;
29 ;   Q:= W(I/O ADDR)
30 ;   WRKO:= Q, RETURN
31

```

!0017

01 ; INSTRUCTION EXECUTION SUBROUTINES
02 ; *****
03 ; *****
04 ; *****

05 ; THE INSTRUCTION EXECUTION SUBROUTINES ARE CALLED
06 ; VIA A SEPARATE ADDRESS TABLE.
07 ; INPUTS TO THE TABLE IS: ESCAPE MODE, IR(0:5)
; SB = EFFECTIVE ADDRESS

```

!0018
01 ;ADDRESS HANDLING
02 ;*****
03
04
05 ;MODIFY NEXT ADDRESS, AM, OPCODE = 11
06
07 0362 001000001101 AME: LDIM 0000,1101
08 0363 071002002226 CAL F,FALSE,ESCAP ; CALL ESCAPE ROUTINE
09
10 0364 001020000000 AM: LDIM 1000,0000
11 0365 433530010104 RABR OR,IMOP,STAT,STAT,CPUST RTN ; AFTER AM:= 1, RETURN
12
13
14
15 ;LOAD ADDRESS, AL, OPCODE = 13
16
17 0366 001000001301 ALE: LDIM 0000,1301
18 0367 071002002226 CAL F,FALSE,ESCAP ; CALL ESCAPE ROUTINE
19
20 0370 433430000160 AL: ZAB OR,SB,W RTN ; W(IR):= SB, RETURN
21
22
23 ;LOAD ADDRESS COMPLEMENTED, AC, OPCODE = 41
24
25 0371 001000004101 ACE: LDIM 0000,4101
26 0372 071002002226 CAL F,FALSE,ESCAP ; CALL ESCAPE ROUTINE
27
28 0373 023550005104 AC: SAB CAND,M2,STAT,STAT
29 0374 033421000160 ZAB SUBO,SB,W
30 0375 071000430377 JMP T,OVFL,INOFL
31 0376 433302000004 ZBB ADDC,STAT RTN
32
33 ;INTEGER OVERFLOW
34
35 0377 023502004504 INOFL: SAB ADDC,C2,STAT,STAT
36 0400 471000542253 JMP T,IMSK,INTA RTN

```

```

; EX(22):= 1, EX(23):= CARRY
; IF INT MASK=1 THEN GOTO INTA ELSE RTN

```

```

!0019
01 ;WORD INSTRUCTIONS WITH MEMORY REFERENCE
02 ;*****
03 ;*****
04 ;LOAD REGISTER, RL, OPCODE = 24
05
06 0401 0010000002402 RLE: LDIM 0000,2402
07 0402 071002002226 CAL F,FALSE,ESCAP
08
09 0403 071002000340 RL: CAL F,FALSE,GETOP
10 0404 433430000240 ZAB OR,WRKO,W RTN
11
12
13 ;LOGICAL AND, LA, OPCODE = 4
14
15 0405 001000000402 LAE: LDIM 0000,0402
16 0406 071002002226 CAL F,FALSE,ESCAP
17
18 0407 071002000340 LA: CAL F,FALSE,GETOP
19 0410 433140000240 ABB AND,WRKO,W RTN
20
21
22 ;LOGICAL OR, LO, OPCODE = 5
23
24 0411 001000000502 LOE: LDIM 0000,0502
25 0412 071002002226 CAL F,FALSE,ESCAP
26
27 0413 071002000340 LO: CAL F,FALSE,GETOP
28 0414 433130000240 ABB OR,WRKO,W RTN
29
30
31 ;LOGICAL EXCLUSIVE OR, LX, OPCODE = 6
32
33 0415 001000000602 LXE: LDIM 0000,0602
34 0416 071002002226 CAL F,FALSE,ESCAP
35
36 0417 071002000340 LX: CAL F,FALSE,GETOP
37 0420 433160000240 ABB EXOR,WRKO,W RTN

```

```

; CALL ESCAPE ROUTINE

```

```

; CALL GETOP, WRKO:=OPERAND
; W(IR):= WRKO, RETURN

```

```

; CALL ESCAPE ROUTINE

```

```

; CALL GETOP, WRKO:= OPERAND
; W(IR):= W(IR) AND WRKO, RETURN

```

```

; CALL ESCAPE ROUTINE

```

```

; CALL GETOP, WRKO:= OPERAND
; W(IR):= W(IR) OR WRKO, RETURN

```

```

; CALL ESCAPE ROUTINE

```

```

; CALL GETOP, WRKO:= OPERAND
; W(IR):= W(IR) EXOR WRKO

```

```

10020
01
02
03 0421 001000000702
04 0422 071002002226
05
06 0423 071002000340
07 0424 023550005104
08 0425 033100000240
09 0426 071000430377
10 0427 433302000004
11
12
13
14
15 0430 0010000001002
16 0431 071002002226
17
18 0432 071002000340
19 0433 023550005104
20 0434 033111000240
21 0435 071000430377
22 0436 433302000004

;ADD INTEGER WORD, WA, OPCODE = 7
WAE: LDIM 0000,0702
CAL F,FALSE,ESCAP
RTN

WA: CAL F,FALSE,GETOP
SAB CAND,M2,STAT,STAT
ABB ADD,WRKO,W
JMP T,OVFL,INOFI
ZBB ADDC,STAT
RTN

;SUBTRACT INTEGER WORD, WS, OPCODE = 10
WSE: LDIM 0000,1002
CAL F,FALSE,ESCAP
RTN

WS: CAL F,FALSE,GETOP
SAB CAND,M2,STAT,STAT
ABB SUNO,WRKO,W
JMP T,OVFL,INOFI
ZBB ADDC,STAT
RTN

; CALL ESCAPE ROUTINE
; CALL GETOP, WRKO:= OPERAND
; EX(22):= EX(23):= 0
; W(IR):= W(IR) + WRKO
; IF OVERFLOW THEN GOTO INOFI
; EX(23):= CARRY

; CALL ESCAPE ROUTINE
; CALL GETOP,WRKO:= OPERAND
; EX(22):= EX(23):= 0
; W(IR):= W(IR) - WRKO
; IF OVERFLOW THEN GOTO INOFI
; EX(23):= CARRY

```

10021								
01					;MULTIPLY INTEGER WORD, WM, OPCODE=12			
02								
03	0437	001000001202	WME:	LDIM	0,1202			
04	0440	071002002226	CAL	F,F,ESCAP				
05								
06	0441	071002000340	WM:	CAL	F,F,GETOP			
07	0442	030330000000	ZBQ	OR,W				
08	0443	033340000001	ZBB	AND,WPRES				
09	0444	044330000401	DRZB	OR,WPRES,Z,MC,F,F				
10								
11	0445	001000000027	LDIM	0,27				
12	0446	233730000013	RZB	OR,IMOP,WRK1	PUSH			
13	0447	033310000013	ZBB	SUN,WRK1				
14								
15	0450	554100024641	MULD	ADD,WRKO,WPRES,SGN,MC,F,NZ	LRTN			
16								
17								
18								
19								
20	0451	054111004241	MULD	SUNO,WRKO,WPRES,SGN,NL,F,F				
21								
22								
23	0452	433230000000	ZQB	OR,W				
								RTN

```
;CALL ESCAPE ROUTINE
```

```
;WRKO:=OPERAND
```

```
;Q:=W
```

```
;WPRES:=0
```

```
;ADCND:=IF Q(23)=1 THEN 0 ELSE 1
```

```
;WPRES CON Q:=0 CON WPRES CON Q(0:22)
```

```
;FOR WRK1:=23
```

```
;STEP -1 UNTIL 0 DO
```

```
;BEGIN
```

```
;WPRES:=IF ADCND=0 THEN WPRES+WRKO
```

```
;ELSE WPRES+0
```

```
;ADCND:=IF Q(23)=1 THEN 0 ELSE 1
```

```
;WPRES CON Q:=WPRES(0) CON WPRES CON Q(0:22)
```

```
;END
```

```
;WPRES:=IF ADCND=0 THEN WPRES-WRKO
```

```
;ELSE WPRES-0
```

```
;WPRES CON Q:=WPRES(0) CON WPRES CON Q(0:22)
```

```
;W:=Q, RTN
```

```

10022
01
02
03 0453 0010000003002
04 0454 071002002226
05
06 0455 071002000340
07 0456 0235500005104
08 0457 043330001412
09 0460 0710000020517
10 0461 0303300000001
11 0462 0332300000013
12 0463 0303300000000
13
14 0464 046330001013
15
16
17 0465 066103005253
18
19
20
21
22
23
24
25 0466 0710000440472
26 0467 0313300000001
27 0470 0710000410474
28 0471 0710000000517
29
30 0472 0333300000001
31 0473 0710000410517

;DIVIDE INTEGER WORD, WD, OPCODE=30
WDE: LDIM 0,3002
CAL F,F,ESCAP
WD: CAL F,F,GETOP
SAB CAND,M2,STAT,STAT
NSZB OR,WRKO,DVS,F,F
JMP F,NZ,WDOFL
ZBQ OR,WPRES
ZQB OR,WRK1
ZBQ OR,W
DLZB OR,WRK1,Z,DC,F,F
DIVD ADDX,WRKO,WRK1,ADC,DC,F,F
;
; AFTER THE FIRST STEP OF DIVISION, QUOTIENT OVERFLOW IS ANTICIPATED BY
; COMPARING THE CARRY FROM BIT(0) OF THE ALU WITH THE SIGN OF THE DIVIDEND
JMP T,CARRY,WD2
ZB OR,WPRES
JMP T,NNEG,WD1
JMP F,F,WDOFL
WD2: ZB OR,WPRES
JMP T,NNEG,WDOFL

;CALL ESCAPE ROUTINE
;WRKO:=OPERAND, (=DIVISOR)
;EX(22):=EX(23):=0
;DVS:=WRKO(0), (=DIVISOR SIGN)
;IF DIVISOR=0 THEN GOTO WDOFL
;WRK1:=WPRES, (=DIVIDEND(0:23))
;Q:=W, (=DIVIDEND(24:47))
;ADCND:=IF WRK1(0)=DVS THEN 1 ELSE 0
;WRK1 CON Q:=WRK1(1:23) CON Q CON 0
;WRK1:=IF ADCND=1 THEN WRK1-WRKO
; ELSE WRK1+WRKO
;ADCND:=IF WRK1(0)=DVS THEN 1 ELSE 0
;WRK1 CON Q:=WRK1(1:23) CON Q CON ADCND
;IF CARRY=1 THEN GOTO WD2
;IF CARRY=0 AND DIVIDEND>=0
;THEN GOTO WD1
;ELSE GOTO WDOFL
;IF CARRY=1 AND DIVIDEND>=0
;THEN GOTO WDOFL

```

```

!0023
01 0474 001000000026      LDIM      0,26
02 0475 233730000014      RZB      OR,IMOP,WRK2      PUSH
03 0476 033310000014      ZBB      SUN,WRK2
04
05 0477 566103025253      DIVD      ADDX,WRK0,WRK1,ADC,DC,F,NZ LRTN
06
07
08
09
10 0500 046330004015      DLZB      OR,WRK3,ADC,NL,F,F
11 0501 063103001253      DIVN      ADDX,WRK0,WRK1,DC,F,F
12
13
14 0502 046330004015      DLZB      OR,WRK3,ADC,NL,F,F
15 0503 053100000253      MULN      ADD,WRK0,WRK1,NL,F,F
16
17
18
19
20
21 0504 031121000272      AB      SUBO,WRK1,WRK0
22 0505 071000020512      JMP      F,NZ,WDCOR
23 0506 031430000260      ZA      OR,WRK1
24 0507 071000020515      JMP      F,NZ,WDEND
25 0510 031160000261      AB      EXOR,WRK1,WPRE
26 0511 071000410515      JMP      T,NNEG,WDEND
27 0512 033111000253      WDCOR:   ABB      SUNO,WRK0,WRK1
28 0513 030201000000      ZQQ      ADDO
29 0514 071000430517      JMP      T,OVFL,WDOFL
30 0515 033430000261      WDEND:   ZAB      OR,WRK1,WPRE
31 0516 433230000000      ZQB      OR,W
32
33 0517 023530004504      WDOFL:   SAB      OR,C2,STAT,STAT
34 0520 471000542253      JMP      T,IMSK,INTA      RTN

```

```

;FOR WRK2:=22
;STEP -1 UNTIL 0 DO
;BEGIN
;WRK1:=IF ADCND=1 THEN WRK1-WRK0
;      ELSE WRK1+WRK0
;ADCND:=IF WRK1(0)=DVS THEN 1 ELSE 0
;WRK1 CON Q:=WRK1(1:23) CON Q CON ADCND
;END
;Q:=Q(1:23) CON ADCND
;WRK1:=IF ADCND=1 THEN WRK1-WRK0
;      ELSE WRK1+WRK0
;ADCND:=IF WRK1(0)=DVS THEN 1 ELSE 0
;Q:=Q(1:23) CON ADCND
;WRK1:=IF ADCND=0 THEN WRK1+WRK0
;      ELSE WRK1+0
;
; AT THIS POINT Q=QUOTIENT AND WRK1=REMAINDER
; THE EQUATION: DIVIDEND=QUOTIENT*DIVISOR+REMAINDER, IS FULFILLED
;
; IF REMAINDER=DIVISOR
; THEN GOTO WDCOR
; IF REMAINDER=0
; THEN GOTO WDEND
; IF SIGN OF DIVIDEND=SIGN OF REMAINDER
; THEN GOTO WDEND
; REMAINDER:=REMAINDER-DIVISOR
; QUOTIENT:=QUOTIENT+1
; IF OVERFLOW THEN GOTO WDOFL
; WPRE:=REMAINDER
; W:=QUOTIENT, RTN
;EX(22):=1
; IF INT MASK=1 THEN GOTO INTA ELSE RTN

```

10024

```

- EOT
; HALF-WORD INSTRUCTIONS WITH MEMORY REFERENCE
; *****

```

```

; LOAD HALF REGISTER, HL, OPCODE=3

```

```

HLE:      LDIM      0000,0302
          CAL       F,FALSE,ESCAP
          HL:
          0521 001000000302
          0522 071002002226
          11
          0523 071002000340
          12
          0524 071000620526
          13
          0525 032730001252
          14
          0526 023540006252
          15
          0527 030330000000
          16
          0530 023650006000
          17
          0531 433130000240
          18
          19
          20
          21
          22
          23
          24
          25
          26
          27
          28
          29
          30
          31
          32
          33
          34
          35
          36
          37
          38
          39
          40

```

```

; CALL ESCAPE ROUTINE

```

```

; WRKO:= OPERAND
; IF ODD ADDR THEN GOTO HLRGT
; WRKO:= WRKO(12:23) CON WRKO(0:11)
; WRKO:= 12 EXT 0 CON WRKO(12:23)
; Q:= W
; W:=W(0:11) CON 12 EXT 0
; W:=W OR WRKO, RTN

```

```

; LOAD INTEGER HALF-WORD (BYTE), ZERO EXTENSION, BZ, OPCODE=23

```

```

BZE:      LDIM      0,2302
          CAL       F,F,ESCAP
          BZ:
          0532 0010000002302
          0533 071002002226
          25
          0534 071002000340
          26
          0535 071000620537
          27
          0536 032730001252
          28
          0537 423540006240
          29
          30
          31
          32
          33
          34
          35
          36
          37
          38
          39
          40

```

```

; CALL ESCAPE ROUTINE

```

```

; WRKO:= OPERAND
; IF ODD ADDR THEN GOTO BZRG
; WRKO:= WRKO(12:23) CON WRKO(0:11)
; W:=12 EXT 0 CON WRKO(12:23), RTN

```

```

; LOAD INTEGER HALF-WORD (BYTE), SIGN EXTENSION, BL, OPCODE=2

```

```

BLE:      LDIM      0000,0202
          CAL       F,FALSE,ESCAP
          BL:
          0540 001000000202
          0541 071002002226
          35
          0542 071002000340
          36
          0543 071000620545
          37
          0544 032730001252
          38
          0545 432730000640
          39
          40

```

```

; CALL ESCAPE ROUTINE

```

```

; WRKO:= OPERAND
; IF ODD ADDR THEN GOTO BLRGT
; WRKO:= WRKO(12:23) CON WRKO(0:11)
; W:= 12 EXT WRK(12) CON WRKO(12:23), RTN

```

10025

;ADD INTEGER HALF-WORD (BYTE), BA, OPCODE = 22

```

01 01
02
03 0546 001000002202 BAE: LDIM 0000,2202
04 0547 071002002226 CAL F,FALSE,ESCAP
05
06 0550 071002000340 BA: CAL F,FALSE,GETOP
07 0551 023550005104 SAB CAND,M2,STAT,STAT
08 0552 071000620554 JMP T,ODD,BARGT
09 0553 032730001252 SWAP WRKO,WRKO
10 0554 032730000652 BARGT: EXT WRKO,WRKO
11 0555 033100000240 ABB ADD,WRKO,W
12 0556 071000430377 JMP T,OVFL,INOFL
13 0557 433302000004 ZBB ADDC,STAT RTN
14
15
16

```

;SUBTRACT INTEGER HALF-WORD (BYTE), BS, OPCODE = 21

```

17 0560 001000002102 BSE: LDIM 0000,2102
18 0561 071002002226 CAL F,FALSE,ESCAP
19
20
21 0562 071002000340 BS: CAL F,FALSE,GETOP
22 0563 023550005104 SAB CAND,M2,STAT,STAT
23 0564 071000620566 JMP T,ODD,BSRGT
24 0565 032730001252 SWAP WRKO,WRKO
25 0566 032730000652 BSRGT: EXT WRKO,WRKO
26 0567 033111000240 ABB SUNO,WRKO,W
27 0570 071000430377 JMP T,OVFL,INOFL
28 0571 433302000004 ZBB ADDC,STAT RTN
29
30
31

```

;LOAD EXCEPTION REGISTER, XL, OPCODE = 20

```

32 0572 001000002002 XLE: LDIM 0000,2002
33 0573 071002002226 CAL F,FALSE,ESCAP
34
35
36 0574 071002000340 XL: CAL F,FALSE,GETOP
37 0575 071000620577 JMP T,ODD,XLRGT
38 0576 032730001252 SWAP WRKO,WRKO
39 0577 001000000007 XLRGT: LDIM 0000,0007
40 0600 033540000252 AND,IMOP,WRKO,WRKO
41 0601 033550000104 RAB CAND,IMOP,STAT,STAT
42 0602 433130000244 RAB OR,WRKO,STAT RTN
43

```

; CALL GETOP, WRKO:= OPERAND

; EX(22):=EX(23):=0

; IF I/O ADDR(23) = 1 THEN GOTO BARGT

; WRKO:= WRKO(12:23) CON WRKO(0:11)

; WRKO:= 12EXT WRKO(12) CON WRKO(12:22)

; W:= W + WRKO

; IF OVERFLOW THEN GOTO INOFL

; EX(23) := CARRY, RETURN

; CALL ESCAPE ROUTINE

; CALL GETOP, WRKO:= OPERAND

; EX(22):=EX(23):=0

; IF ODD ADDR THEN GOTO BSRGT

; WRKO:= WRKO(12:23) CON WRKO(0:11)

; WRKO:= 12EXT WRKO(12) CON WRK(12:23)

; W:= W - WRKO

; IF OVERFLOW THEN GOTO INOFL

; EX(23):= CARRY

; CALL ESCAPE ROUTINE

; CALL GETOP, WRKO:= OPERAND

; IF ODD ADDR THEN GOTO XLRGT

; WRKO:=WRKO(12:23) CON WRKO(0:11)

; IMOP:= 0000007

; WRKO:= 21EXTO CON WRKO(21:23)

; STAT:= STAT(0:20) CON 3 EXT 0

; STAT:= STAT(0:20) CON WRKO(21:23), RETURN

!0026

```
;STORE INSTRUCTIONS
;*****
```

```
;SUBROUTINE STORE WORD
```

```
;STORES W(IR) IN THE ADDRESSED MEMORY LOCATION
```

```
; RETURNS WITH: I/O ADDRESS=ADDRESS OF MEMORY LOCATION
```

```
; : WRK1=SB+BASE
```

```
11 0603 0235000000573
12 0604 021511001260
13 0605 071000010613
14 0606 021520001660
15 0607 071000012251
16 0610 031330030000
17 0611 031330620013
18 0612 471000602251
19 0613 030330000000
20 0614 001000000010
21 0615 031511000160
22 0616 071000412251
23 0617 031330220007
24 0620 071000612251
25 0621 433230000003
```

```
STWD:  SAB
      SA
      JMP
      SA
      JMP
      ZBR
      ZBR
      JMP
      ZBQ
      LDIM
      RA
      JMP
      ZBR
      JMP
      ZQB

      STWD1:  ZBQ
      STWD2:  LDIM

      ADD,BASE,SB,WRK1
      SUNO,LLIM,WRK1
      F,NNEG,STWD1
      SUB,ULIM,WRK1
      F,NNEG,INTO
      OR,W,DATO
      OR,WRK1,WRTP
      T,NMADR,INTO
      OR,W
      O,10
      SUNO,IMOP,SB
      T,NNEG,INTO
      OR,SB,READP
      T,NWADR,INTO
      OR,GRX

      ;WRK1:=SB+BASE
      ;IF WRK1<LOWER LIMIT
      ;THEN GOTO STWD1
      ;IF WRK1>=UPPER LIMIT
      ;THEN GOTO INTO
      ;DATAOUT:=W(IR)
      ;I/O ADDR:=WRK1, WRITEP
      ;IF I/O ADDR<8 THEN GOTOINTO ELSE RTN
      ;Q:=W(IR)

      ;IF SB>=8
      ;THEN GOTO INTO
      ;I/O ADDR:=SB, READP
      ;IF 0=<I/O ADDR<8 THEN GOTO INTO
      ;W(I/O ADDR):=Q, RTN

      RTN
      RTN
```

10027

;STORE REGISTER, RS, OPCODE=27

01
02
03 0622 001000002704
04 0623 071002002226

RSE: LDIM 0,2704
CAL F,F,ESCAP

;CALL ESCAPE ROUTINE

05
06 0624 071002000603
07 0625 471010641742

RS: CAL F,F,STWD
WJMP T,BERR,INTB RTN

;MEM(ADDR):=W(IR)

;WAIT, IF BUSERROR THEN GOTO INTB ELSE RTN

;STORE DOUBLE REGISTER, DS, OPCODE=67

08
09
10
11
12 0626 001000006714
13 0627 071002002226

DSE: LDIM 0,6714
CAL F,F,ESCAP

;CALL ESCAPE ROUTINE

14
15 0630 071002000603
16 0631 030330000001
17 0632 071000210643
18 0633 023511004673
19 0634 021511001260
20 0635 071000010645
21 0636 071010641742
22 0637 031330030001
23 0640 031430620260
24 0641 071000602251
25 0642 471010641742

DS: CAL F,F,STWD
ZBQ OR,WPRE
JMP F,NWADR,DS1
SAB SUNO,C2,WRK1,WRK1
SA SUNO,LLIM,WRK1
JMP F,NNEG,DS2
WJMP T,BERR,INTB
ZBR OR,WPRE,DATO
ZAR OR,WRK1,WRTP
JMP T,NMADR,INTO
WJMP T,BERR,INTB RTN

;MEM(ADDR):=W(IR)
;Q:=WPRE(IR)
;IF 0=<I/O ADDR<8 THEN GOTO DS1
;WRK1:=WRK1-2
;IF WRK1<LOWER LIMIT
;THEN GOTO DS2
;IF BUSERROR THEN GOTO INTB
;DATAOUT:=WPRE(IR)
;I/O ADDR:=WRK1,WRITEP
;IF I/O ADDR<8 THEN GOTOINTO
;WAIT,
;IF BUSERROR THEN GOTO INTB ELSE RTN

26
27
28 0643 023511224567
29 0644 433230000003

DS1: SABR SUNO,C2,SB,SB,READP
ZQB OR,GRX RTN

;I/O ADDR:=SB, READP
;W(I/O ADDR):=Q, RTN

30
31 0645 023511004567
32 0646 071000000614

DS2: SAB SUNO,C2,SB,SB
JMP F,F,STWD2

;SB:=SR-2
;GOTO STWD2

10028

```

;SUBROUTINE GET OPERAND FOR WRITE MODIFICATION
;
;   FETCHES AN OPERAND FROM THE MEMORY LOCATION ADDRESSED BY SB+BASE
;   OR FROM W(SB) IF U<=SB<8.
;   RETURNS WITH WRKO=FETCHED OPERAND
;
;   GTWOP:  SADR  ADD,BASE,SB,WRK3,READP
;           JMP    T,NMADR,GTW01
;           SA     SUNO,LLJM,WRK3
;           JMP    F,NNEG,GTW02
;           SA     SUB,ULIM,WRK3
;           JMP    F,NNEG,INT0
;           WJMP   T,BERR,INTB
;           RZB    OR,DATI,WRKO      RTN
;
;           GTW02:  WJMP   F,F,GTW01
;           GTW01:  ZAR    OR,SB,READP
;           JMP    F,NWADR,GTW03
;           WJMP   F,F,INT0
;
;           GTW03:  ZBQ    OR,GRX
;           ZQB    OR,WRKO RTN
;
;   I/O ADDR:=WRK3:=SB+BASE, READP
;   IF I/O ADDR < 8 THEN GOTO GTW01
;   IF WRK3 < LOWER LIMIT
;   THEN GOTO GTW02
;   IF WRK3 <= UPPER LIMIT
;   THEN GOTO INT0
;   WAIT, IF BUSERROR THEN GOTO INTB
;   WRKO:= DATAIN, RTN
;
;   WAIT, GOTO GTW01
;   I/O ADDR:= SB, READP
;   IF 0 <= I/O ADDR < 8 THEN GOTO GTW03
;   WAIT, GOTO INT0
;
;   Q:= W(I/O ADDR)
;   WRKO:= Q, RTN

```

10029

;STORE HALF REGISTER, HS, OPCODE = 32

01
02
03 0665 001000003204
04 0666 071002002226
05

HSE: LDIM
CAL

0,3204
F,F,ESCAP

; CALL ESCAPE ROUTINE

06 0667 0710020000647
07 0670 0303300000000
08 0671 0236400006013
09 0672 071000620703
0 0673 032730001273
1 0674 0235400006252
2

HS: CAL
ZBQ
SQB
JMP
SWAP
SAB

F,F,GTWOP
OR,W
AND,M12,WRK1
T,ODD,HSRGT
WRK1,WRK1
AND,M12,WRK0,WRK0

; WRKO:= MEMORY WORD
; Q:= W
; WRK1:= 12EXT0 CON Q(12:23)
; IF I/O ADDR(23) = 1 THEN GOTO HSRGT
; WRK1:= WRK1(12:23) CON WRK1(0:11)
; WRKO:= 12EXT 0 CON WRKO(12:23)

3 0675 033130030272
4 0676 071000210702
5 0677 021500620560
6 0700 071000602251
7 0701 471010641742
8 0702 4334300000243
9

HSTOR: ABBR
JMP
SAR
JMP
WJMP
ZAR

OR,WRK1,WRKO,DATO
F,NWADR,HSREG
ADD,BASE,SB,WRTP
T,NMADR,INTO
T,BERR,INTB
OR,WRKO,GRX

; DATAOUT:= WRKO:= WRKO OR WRK1
; IF 0 =< I/O ADDR < 8 THEN GOTO HSREG
; I/O ADDR:= SB + BASE, WRTP
; IF I/O ADDR < 8 THEN GOTO INTO
; WAIT, IF BUSERROR THEN GOTO INTB ELSE RTN
; W(I/O ADDR):= WRKO, RTN

0 0703 0235500006252
1 0704 0710000000675
2
3
4
5

HSRGT: SAB
JMP

CAND,M12,WRKO,WRKO
F,F,HSTOR

; WRKO:= WRKO(0:11) CON 12EXT0
; GOTO HSTOR

6 0705 0010000003304
7 0706 071002002226
8
9

XSE: LDIM
CAL

0,3304
F,F,ESCAP

; CALL ESCAPE ROUTINE

0 0707 0710020000647
1 0710 0010000000007
2 0711 0335400000113
3 0712 071000620703
4 0713 0710000000673

XS: CAL
LDIM
RAB
JMP
JMP

F,F,GTWOP
0,7
AND,IMOP,STAT,WRK1
T,ODD,HSRGT
F,F,HSLFT

; WRKO:= MEMORY WORD
; WRK1:= 23 EXT 0 CON EX
; IF I/O ADDR(23) = 1 THEN GOTO HSRGT
; GOTO HSLFT

5 0705 0010000003304
6 0706 071002002226
7
8
9

XSE: LDIM
CAL

0,3304
F,F,ESCAP

; CALL ESCAPE ROUTINE

;STORE EXCEPTION REGISTER, XS, OPCODE = 33

0 0707 0710020000647
1 0710 0010000000007
2 0711 0335400000113
3 0712 071000620703
4 0713 0710000000673

XS: CAL
LDIM
RAB
JMP
JMP

F,F,GTWOP
0,7
AND,IMOP,STAT,WRK1
T,ODD,HSRGT
F,F,HSLFT

; WRKO:= MEMORY WORD
; WRK1:= 23 EXT 0 CON EX
; IF I/O ADDR(23) = 1 THEN GOTO HSRGT
; GOTO HSLFT

```

;EXCHANGE REGISTER AND MEMORY WORD, RX, OPCODE = 31
01
02
03 0714 001000003106 RXE: LDIN 0,3106
04 0715 071002002226 CAL F,F,ESCAP ; CALL ESCAPE ROUTINE
05
06 0716 071002000647 RX: CAL F,F,GTWOP ; WRKO:= MEMORY WORD
07 0717 030330000000 ZBQ OR,W ; Q:= W(IR)
08 0720 031230030000 ZQR OR,DATO ; DATAOUT:= Q
09 0721 071000210725 JMP F,NWADR,RXREG ; IF 0 <= I/O ADDR < 8 THEN GOTO RXREG
10 0722 021500620560 SAR ADD,BASE,SB,WRTP ; I/O ADDR:= SB + BASE, WRTEP
11 0723 033430000240 ZAB OR,WRKO,W ; W(IR):= WRKO
12 0724 471010641742 WJMP T,BERR,INTR RIN ; IF BUSERROR THEN GOTO INIB ELSE RIN
13 0725 033230000003 RXREG: ZQB OR,GRX ; W(I/O ADDR):= Q
14 0726 433430000240 ZAB OR,WRKO,W ; W(IR):= WRKO, RIN

```

10031

```
;DOUBLE-WORD INSTRUCTIONS WITH MEMORY REFERENCE
;*****
```

```
;SUBROUTINE GET DOUBLE-WORD
```

```
;
;   FETCHES A DOUBLE-WORD FROM THE ADDRESSED MEMORY LOCATION
;   AND THE PRECEDING MEMORY LOCATION.
;   RETURNS WITH: WRK1 = WORD(ADDRESS)
;               WRK2 = WORD(ADDRESS-2)
```

12	0727	023500220574	GTDWD:	SABR	ADD,BASE,SB,WRK2,READP	; I/O ADDR:= WRK2:= SB + BASE, READP
13	0730	071000600746		JMP	T,NMADR,GTDW1	; IF I/O ADDR < 8 THEN GOTO GTDW1
14	0731	021511001300		SA	SUNO,LLIM,WRK2	; IF WRK2 < LOWER LIMIT
15	0732	071000010745		JMP	F,NNEG,GTDW2	; THEN GOTO GTDW2
16	0733	021520001700		SA	SUB,ULIM,WRK2	; IF WRK2 >= UPPER LIMIT
17	0734	071000010745		JMP	F,NNEG,GTDW2	; THEN GOTO GTDW2
18	0735	071010641742		WJMP	T,BERR,INTB	; WAIT, IF BUSERROR THEN GOTO INTB
19	0736	033730001413		RZB	OR,DATI,WRK1	; WRK1:= DATAIN
20	0737	023511224714		SABR	SUNO,C2,WRK2,WRK2,READP	; I/O ADDR:= WRK2:= WRK2 - 2, READP
21	0740	023511004567		SAB	SUNO,C2,SB,SB	; SB:= SB - 2
22	0741	021511001300		SA	SUNO,LLIM,WRK2	; IF WRK2 < LOWER LIMIT
23	0742	071000010755		JMP	F,NNEG,GTDW3	; THEN GOTO GTDW3
24	0743	071010641742		WJMP	T,BERR,INTB	; WAIT, IF BUSERROR THEN GOTO INTB
25	0744	433730001412		RZB	OR,DATI,WRK0	; WRK0:= DATAIN, RETURN
26						
27	0745	0710100000746	GTDW2:	WJMP	F,F,GTDW1	; WAIT, GOTO GTDW1
28	0746	031430220160	GTDW1:	ZAR	OR,SB,READP	; I/O ADDR:= SB, READP
29	0747	071000610761		JMP	T,NWADR,GTDW6	; IF 0 > I/O ADDR >= 8 THEN GOTO GTDW6
30	0750	030330000003		ZBQ	OR,GRX	; Q := W(I/O ADDR)
31	0751	033230000013		ZQB	OR,WRK1	; WRK1:= Q
32	0752	023511224567		SABR	SUNO,C2,SB,SB,READP	; I/O ADDR:= SB:= SB - 2
33	0753	030330000003	GTDW5:	ZBQ	OR,GRX	; Q:= W(I/O ADDR)
34	0754	433230000012		ZQB	OR,WRK0 RTN	; WRK0:= Q, RETURN
35						
36	0755	0710100000756	GTDW3:	WJMP	F,F,GTDW4	; WAIT, GOTO GTDW4
37	0756	031430220160	GTDW4:	ZAR	OR,SB,READP	; I/O ADDR:= SB, READP
38	0757	0710000210753		JMP	F,NWADR,GTDW5	; IF 0 <= I/O ADDR < 8 THEN GOTO GTDW5
39	0760	0710000002251		JMP	F,F,INTO	; GOTO INTO

!0032

01 0761 071000602251
02 0762 021520000160
03 0763 071000012251
04 0764 071010641742
05 0765 033730001413
06 0766 023511224567
07 0767 071000602251
08 0770 071010641742
09 0771 433730001412

GTDW6:

JMP
SA
JMP
WJMP
RZB
SABR
JMP
WJMP
RZB

T,NMADR,INTO
SUB,CPA,SB
F,NNEG,INTO
T,BERR,INTB
OR,DATI,WRK1
SUNO,C2,SB,SB,READP
T,NMADR,INTO
T,BERR,INTB
OR,DATI,WRKO

RTN

; IF I/O ADDR < 8 THEN GOTO INTO
; IF SB >= CPA
; THEN GOTO INTO
; WAIT, IF BUSERROR THEN GOTO INTO
; WRK1:=DATAIN
; I/O ADDR:=SB := SB - 2
; IF I/O ADDR < 8 THEN GOTO INTO
; WAIT, IF BUSERROR THEN GOTO INTO
; WRKO:=DATAIN, RETURN

```

10033
01
02
03 0772 001000006612
04 0773 071002002226
05
06 0774 071002000727
07 0775 033430000260
08 0776 071000601000
09 0777 433430000241
10
11 1000 030330000003
12 1001 433230000001
13
14
15
16
17 1002 001000007012
18 1003 071002002226
19
20 1004 071002000727
21 1005 023550005104
22 1006 033100000260
23 1007 030122000377
24 1010 044330000417
25 1011 071002600753
26
27 1012 033103000241
28 1013 071000430377
29 1014 433302000004
30
31

;LOAD DOUBLE REGISTER, DL, OPCODE = 66
DLE: LDIM 0,6612
CAL F,F,ESCAP
DL: CAL F,F,GTDWD
ZAB OR,WRK1,W
JMP T,NMADR,DL1
ZAB OR,WRKO,WPRES RTN
DL1: ZBQ OR,GRX
ZQB OR,WPRES RTN

;ADD INTEGER DOUBLE WORD, AA, OPCODE = 70
AAE: LDIM 0,7012
CAL F,F,ESCAP
AA: CAL F,F,GTDWD
SAB CAND,M2,STAT,STAT
ABB ADD,WRK1,W
ABQ SUBC,WRK5,WRK5
DSRB OR,WRK5,Z,MC
CAL T,NMADR,GTDW5
ABB ADDX,WRKO,WPRES
JMP T,OVFL,INOFI
ZBB ADDC,STAT RTN

;CALL ESCAPE ROUTINE
; WRK1:= WORD(ADDR), WRKO:= WORD(ADDR-2)
; W(IR):= WRK1
; IF I/O ADDR<8 THEN GOTO DL1
; WPRE(IR):= WRKO, RETURN
; WPRE(IR):=W(I/O ADDR), RTN

;CALL ESCAPE ROUTINE
; WRK1:= WORD(ADDR), WRKO:= WORD(ADDR-2)
; EX(22):= EX(23):= 0
; W(IR):= W(IR) + WRK1
; Q:=IF CARRY THEN 0 ELSE -1
; ADCND:=CARRY
; IF I/O ADDR<8 THEN
; WRKO:=W(I/O ADDR)
; WPRE(IR):= WPRE(IR)+WRKO+ADCND
; IF OVERFLOW THEN GOTO INOFI
; EX(23):= CARRY, RTN

```

```

10034
01
02
03 1015 001000007112
04 1016 071002002226
05
06 1017 071002000727
07 1020 023550005104
08 1021 033111000260
09 1022 030122000377
10 1023 044330000417
11 1024 071002600753
12
13 1025 0331113000241
14 1026 071000430377
15 1027 433302000004
16
;SUBTRACT INTEGER DOUBLE WORD, SS, OPCODE = 71

```

```

; CALL ESCAPE ROUTINE
; WRK1:= WORD(ADDR), WRK0:= WORD(ADDR-2)
; EX(22):= EX(23):= U
; W(IR):= W(IR) - WRK1
; Q:=IF CARRY THEN 0 ELSE -1
; ADCND:=CARRY
; IF I/O ADDR<8 THEN
;   WRK0:=W(I/O ADDR)
; WPRE(IR):= WPRE(IR)-WRK0-1+ADCND
; IF OVERFLOW THEN GOTO INOFL
; EX(23):= CARRY, RTN

```

```

CAL LDIM 0,7112
CAL F,F,ESCAP
CAL F,F,GTDWD
SAB CAND,M2,STAT,STAT
ABB SUNO,WRK1,W
ABQ SUBC,WRK5,WRK5
DSRB OR,WRK5,Z,MC
CAL T,NMADR,GTDW5
ABB SUNX,WRK0,WPRE
JMP T,OVFL,INOFL
ZBB ADDC,STAT RTN

```

!0035

;SHIFT INSTRUCTIONS
;*****

;SUBROUTINE SET SHIFT COUNT

```
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
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  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
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  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
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  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  

```

10036

; ARITHMETICALLY SHIFT SINGLE, AS, OPCODE = 44

```

01
02
03 1035 0010000004401 ASE: LDIM 0,4401
04 1036 071002002226 CAL F,F,ESCAP
05
06 1037 0334300000172 AS: ZAB OR,SB,WRKO
07 1040 0710000011055 JMP F,NNEG,ASR
08 1041 0710000021061 JMP F,NZ,ASR1
09 1042 071002001031 ASL: CAL F,F,SCNTL
10 1043 0235500005104 SAB CAND,M2,STAT,STAT
11 1044 2333300000000 ZBB OR,W PUSH
12
13 1045 071002451054 CAL T,NORM,ASOF1
14 1046 0333100000012 ZBB SUN,WRKO
15 1047 547330020000 SLZB OR,W,Z,NL,F,NZ LRTN
16
17 1050 071002451054 CAL T,NORM,ASOF1
18 1051 0215400004500 ASLOF: SA AND,C2,STAT
19 1052 4710000421053 JMP T,NZ,ASOF2
20 1053 4710000542253 ASOF2: JMP T,IMSK,INTA
21
22 1054 4235300004504 ASOF1: SAB OR,C2,STAT,STAT
23
24
25 1055 071002001030 ASR: CAL F,F,SCNTR
26 1056 2333300000000 ZBB OR,W PUSH
27
28 1057 0333100000012 ZBB SUN,WRKO
29 1060 5453000024000 SRZB ADD,W,SGN,NL,F,NZ
30
31 1061 4235500005104 ASR1: SAB CAND,M2,STAT,STAT

```

; CALL ESCAPE ROUTINE

```

; WRKO:= SB
; IF SB < 0 THEN GOTO ASR
; IF SB = 0 THEN GOTO ASR1
; WRKO:= IF WRKO > 48 THEN 48 ELSE WRKO
; EX(22):=EX(23):=0
; W:= W, PUSH

```

```

; IF W(0) <> W(1) THEN EX(22):=1
; WRKO:= WRKO - 1
; W:= W(1:23) CON 0
; IF WRKO = 0 THEN POP ELSE REPEAT
; IF W(0)<>W(1) THEN EX(22):=1
; IF EX(22)=1
; THEN GOTO ASOF2 ELSE RTN
; IF INT MASK=1 THEN GOTO INTA ELSE RTN

```

; EX(22):=1, RTN

```

; WRKO:= IF WRKO >= -48 THEN -WRKO ELSE 48
; W:= W, PUSH

```

```

; WRKO:= WRKO - 1
; W:= W(0) CON W(0:22)
; IF WRKO = 0 THEN POP ELSE REPEAT
; EX(22):= EX(23):= 0, RETURN

```

0037

;ARITHMETICALLY SHIFT DOUBLE, AD, OPCODE = 45

```

1 3 1062 0010000004501
2 4 1063 071002002226
3 5 1064 0235500005104
4 6 1065 0303300000000
5 7 1066 033430000172
6 8 1067 071000011101
7 9 1070 471000421071
8 0 1071 071002001031
9 1 1072 233330000001
2 3 1073 071002451054
3 4 1074 033310000012
4 5 1075 546330020001
5 6 1076 071002451054
6 7 1077 033230000000
7 8 1100 071000001051
8 9 1101 071002001030
9 0 1102 233330000001
0 1 1103 033310000012
1 2 1104 544300024001
2 3 1105 433230000000

; CALL ESCAPE ROUTINE
; EX(22):= EX(23) := 0
; Q:= W
; WRKO := SB
; IF SB < 0 THEN GOTO ADR
; IF SB <> 0 THEN GOTO ADL ELSE RTN
; WRKO:= IF WRKO =< 48 THEN WRKO ELSE 48
; WPRE:= WPRE, PUSH
; IF WPRE(0) <> WPRE(1) THEN EX(22):=1
; WRKO:= WRKO - 1
; WPRE CON Q:= WPRE(1:22) CON Q CON 0
; IF WRKO = 0 THEN POP ELSE REPEAT
; IF WPRE(0)<>WPRE(1) THEN EX(22):=1
; W:= Q
; GOTO ASLOF

; WRKO:= IF WRKO <= -48 THEN -WRKO ELSE 48
; WPRE:= WPRE, PUSH
; WRKO:= WRKO - 1
; WPRE CON Q:= WPRE(0) CON WPRE CON Q(0:22)
; IF WRKO = 0 THEN POP ELSE REPEAT
; W:= Q, RTN

```

;LOGICALLY SHIFT SINGLE, LS, OPCODE = 46

; CALL ESCAPE ROUTINE

```
WRK := SE
```

```

; IF SB < 0 THEN GOTO LSR

```

```
: IF SB <> 0 THEN GOTO LSL ELSE RTN
```

```
: WRKO:= IF WRKO =< 48 THEN WRKO ELSE 48
```

W: W, PUSH

WRKQ := WRKQ - 1

W:=W(1:23) CON 0

```
; IF WRKO = 0 THEN POP ELSE REPEAT
```

W. W. R. T. N.

```
: WRK0:= IF WRK0 >= -48 THEN -WRK0 ELSE 48
```

$$: W := W, \text{ PUSH}$$

```

: WRKO:= WRKO - 1

```

$$W := \bigcup_{i=0}^{\infty} W_i$$

```

: IF WRKO = 0 THEN POP ELSE REPEAT

```

$$w := w, \text{ RTN}$$

10039

;LOGICALLY SHIFT DOUBLE, LD, OPCODE = 47

```

01
02
03 1125 0010000004701
04 1126 071002002226
05
06 1127 0334300000172
07 1130 0710000011137
08 1131 471000421132
09 1132 071002001031
0 1133 230330000000
1
2 1134 0333100000012
3 1135 546330020001
4
5 1136 433230000000
6
7 1137 071002001030
8 1140 230330000000
9
0 1141 0333100000012
1 1142 544330020001
2
3 1143 433230000000

; CALL ESCAPE ROUTINE
; WRKO:= SB
; IF SB < 0 THEN GOTO LDR
; IF SB <> 0 THEN GOTO LDL ELSE RTN
; WRKO:= IF WRKO =< 48 THEN WRKO ELSE 48
; Q:= W, PUSH
; WRKO:= WRKO - 1
; WPRE CON Q:= WPRE(1:22) CON Q CON 0
; IF WRKO = 0 THEN POP ELSE RTN
; W:= Q, RTN
; WRKO:= IF WRKO >= -48 THEN -WRKO ELSE 42
; Q:= W, PUSH
; WRKO:= WRKO - 1
; WPRE CON Q:= 0 CON WPRE(0:23) CON Q(0:22)
; IF WRKO = 0 THEN POP ELSE REPEAT
; W:= Q, RTN

```

```

10040
01
02
03 1144 0010000004204
04 1145 071002002226
05
06 1146 031330000000
07 1147 071000021162
08 1150 001000000001
09 1151 233730000013
10
11 1152 033310000013
12 1153 031330000000
13 1154 547330450000
14
15 1155 0453300002000
16 1156 023540006273
17 1157 071002000647
18 1160 071000620703
19
20 1161 071000000673
21
22
23 1162 0010000004000
24 1163 033730000013
25 1164 071000001157
26
27

;NORMALIZE SINGLE, NS, OPCODE = 42
; CALL ESCAPE ROUTINE
; IF W=0
; THEN GOTO NS1
; WRK1:= 1, PUSH
; WRK1:=WRK1-1
; NORM:=IF W(0)<>W(1) THEN 1 ELSE 0
; SH LINK CON W:= W(0:23) CON 0
; IF SH LINK <> W(0) THEN POP ELSE REPEAT
; W:= SHLINK CON W(0:22)
; WRK1:= 12 EXT 0 CON WRK1(12:23)
; WRK0:= WORD(ADDR)
; WORD(ADDR):= WORD(0:11) CON WRK1(12:23),
; RTN
; WORD(ADDR):= WRK1(12:23) CON WORD(12:23)
; RTN
; WRK1:= 12 EXT 0 CON -2048
; GOTO NS2

```

0041

;NORMALIZE DOUBLE, ND, OPCODE = 43

1
 2
 3 1165 001000004304
 4 1166 071002002226
 5
 6 1167 031330000001
 7 1170 071000421173
 8 1171 031330000000
 9 1172 071000021162
 0 1173 030330000000
 1 1174 001000000001
 2 1175 233730000013
 3
 4 1176 0333100000013
 5 1177 031330000001
 6 1200 546330450001
 7
 8 1201 044330002001
 9 1202 032330000000
 0 1203 071000001156

NDE: LDIM
 CAL
 ND: ZB
 JMP
 ZB
 JMP
 ZBQ
 LDIM
 RZB
 ND1: ZBB
 ZB
 DLZB
 DSRB
 ZQB
 JMP

0,4304
 F,F,ESCAP
 OR,WPRE
 T,NZ,ND1
 OR,W
 F,NZ,NS1
 OR,W
 0,1
 OR,IMOP,WRK1
 SUN,WRK1
 OR,WPRE
 OR,WPRE,Z,NL,T,NORM
 OR,WPRE,LNK,NL
 OR,W
 F,F,NS3

; CALL ESCAPE ROUTINE

; 0 OR WPRE

; IF WPRE <> 0 THEN GOTO ND1

; IF WPRE CON W = 0 THEN GOTO NS1

; Q:=W

; WRK1:= 1, PUSH

; WRK1:=WRK1-1

; NORM:=IF WPRE(0)<>WPRE(1) THEN 1 ELSE 0

; SHLINK CON WPRE CON Q:= WPRE CON Q CON 0

; IF SHLINK <> WPRE(0) THEN POP ELSE REPEAT

; WPRE CON Q := SHLINK CON WPRE CON Q(0:22)

; W:= Q

; GOTO NS3

LRTN

!0042

;SKIP INSTRUCTIONS
;*****

;SKIP INSTRUCTIONS WHICH SKIPS JUMPS TO THIS CODE

SKIP: SAB ADD,C2,IC,IC ; IC:= IC + 2
SAB ADD,C2,PC,PC RTN ; PC:= PC + 2

;SKIP IF REGISTER HIGH, SH, OPCODE = 50

SHE: LDIM 0,5021
CAL F,F,ESCAP ; CALL ESCAPE ROUTINE

SH: AB SUBO,SB,W ; SB - W
JMP T,OVFL,SH1 ; IF OVERFLOW THEN GOTO SH1
JMP F,NNEG,SKIP RTN ; IF (SB-W) < 0 THEN GOTO SKIP ELSE RTN
JMP T,NNEG,SKIP RTN ; IF (SB-W) >= 0 THEN GOTO SKIP ELSE RTN

;SKIP IF REGISTER LOW, SL, OPCODE = 51

SLE: LDIM 0,5121
CAL F,F,ESCAP ; CALL ESCAPE ROUTINE

SL: AB SUNO,SB,W ; W - SB
JMP T,OVFL,SL1 ; IF OVERFLOW THEN GOTO SL1
JMP F,NNEG,SKIP RTN ; IF (W-SB) < 0 THEN GOTO SKIP ELSE RTN
JMP T,NNEG,SKIP RTN ; IF (W-SB) >= 0 THEN GOTO SKIP ELSE RTN

;SKIP IF REGISTER EQUAL, SE, OPCODE = 52

SEE: LDIM 0,5221
CAL F,F,ESCAP ; CAL ESCAPE ROUTINE

SE: AB SUBO,SB,W ; IF (SB-W) = 0
JMP F,NZ,SKIP RTN ; THEN GOTO SKIP ELSE RTN

```

10043
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

;SKIP IF REGISTER NOT EQUAL, SN, OPCODE = 53
SNE: LDIM 0,5321
      CAL F,F,ESCAP
      ; CALL ESCAPE ROUTINE
      ; IF (SB-W) <> 0
      ; THEN GOTO SKIP ELSE RTN
      RTN

SN: AB SUBO,SB,W
    JMP T,NZ,SKIP
    RTN

;SKIP IF REGISTER BITS ONE, SO, OPCODE = 54
SOE: LDIM 0,5421
      CAL F,F,ESCAP
      ; CALL ESCAPE ROUTINE
      ; Q:= SB AND W
      ; IF SB = SB AND W
      ; THEN GOTO SKIP ELSE RTN
      RTN

SO: ABQ AND,SB,W
    AQ SUBO,SB
    JMP F,NZ,SKIP
    RTN

;SKIP IF REGISTER BITS ZERO, SZ, OPCODE = 55
SZE: LDIM 0,5521
      CAL F,F,ESCAP
      ; CALL ESCAPE ROUTINE
      ; IF SB AND W = 0
      ; THEN GOTO SKIP ELSE RTN
      RTN

SZ: AB AND,SB,W
    JMP F,NZ,SKIP
    RTN

;SKIP IF NO EXCEPTIONS, SX, OPCODE = 56
SXE: LDIM 0,5621
      CAL F,F,ESCAP
      ; CALL ESCAPE ROUTINE
      ; Q:= 21 EXT 0 CON EX(21:23)
      ; IF (21 EXT 0 CON EX) = 0
      ; THEN GOTO SKIP ELSE RTN
      RTN

SX: LDIM 0,7
    RAQ AND,IMOP,STAT
    AQ AND,SB
    JMP F,NZ,SKIP
    RTN

1226 0010000005321
1227 071002002226
1230 031121000160
1231 471000421204
1232 0010000005421
1233 071002002226
1234 030140000160
1235 031021000160
1236 471000021204
1237 0010000005521
1240 071002002226
1241 031140000160
1242 471000021204
1243 0010000005621
1244 071002002226
1245 001000000007
1246 030540000100
1247 031040000160
1250 471000021204

```

10044

;SKIP IF NO WRITE PROTECTION, SP, OPCODE = 25

01	1251	0010000002521	SPE:	LDIM	0,2521				
02	1252	071002002226		CAL	F,F,ESCAP				
03	1253	0313300000007	SP:	ZB	OR,SB				
04	1254	471000411255		JMP	T,NNEG,SKPP1				
05	1255	0010000000010	SKPP1:	LDIM	0,10				
06	1256	0235000000573		SAB	ADD,BASE,SB,WRK1				
07	1257	021511001260		SA	SUNO,LLIM,WRK1				
08	1260	071000011263		JMP	F,NNEG,SKPP2				
09	1261	021511001660		SA	SUNO,ULIM,WRK1				
10	1262	471000011204		JMP	F,NNEG,SKIP				
11	1263	031511000160	SKPP2:	RA	SUNO,IMOP,SB				
12	1264	471000011204		JMP	F,NNEG,SKIP				

; CALL ESCAPE ROUTINE

```

; IF SB < 0
; THEN RTN
; IMOP:= 8
; WRK1:= SB + BASE
; IF (SB+BASE) < LOWER LIMIT
; THEN GOTO SKPP2
; IF (SB + BASE) < UPPER LIMIT
; THEN GOTO SKIP ELSE RTN

; IF SB < 8.
; THEN GOTO SKIP ELSE RTN

```

!0045

;JUMP INSTRUCTIONS
;*****

;JUMP WITH REGISTER LINK, JL, OPCODE = 15

07 1265 001000001520
08 1266 071002002226
09
10 1267 071002000340
11
12
13 1270 071000171272
14 1271 033430000120
15 1272 001000000001
16 1273 033550000165
17 1274 033550000270
18 1275 071000700260
19 1276 071000000251

JLE: LDIM 0,1520
CAL F,F,ESCAP

JL: CAL F,F,GETOP

JL1: JMP F,LINK,JL2
ZAB OR,IC,W
LDIM 0,1
JL2: RAB CAND,IMOP,SB,IC
RAR CAND,IMOP,WRK1,PC
JMP T,INTR,INTT
JMP F,F,AFTCH

; CALL ESCAPE ROUTINE
; WRKO:= NEXT INSTRUCTION
; WRK1:= PHYSICAL ADDRESS
; SB:= LOGICAL ADDRESS
; IF W-FIELD=0 THEN GOTO JL2
; W(IR):=IC
; IC:= SB(0:22) CON 0
; PC:= WRK1(0:22) CON 0
; IF INTERRUPT THEN GOTO INTT
; GOTO AFTCH (MAIN LOOP)

```

!0046
01
02
03 1277 0010000001660
04 1300 071002002226
05
06 1301 0215000003560
07 1302 071000411306
08 1303 0010000004000
09 1304 031500000160
10 1305 071000411316
11 1306 071002000340
12
13
14 1307 0237300002014
15 1310 032730001314
16 1311 0235400005714
17 1312 0113200006414
18 1313 0010000000010
19 1314 033530000104
20 1315 0710000001270
21
22 1316 0207200002400
23 1317 0112200002400
24 1320 033500000166
25 1321 0010000000040
26 1322 0335500040231
27 1323 0710000001762
28
29

; JUMP WITH INTERRUPT DISABLED, JD, OPCODE = 16

JDE:  LDIM 0,1660
      CAL  F,F,ESCAP

JD:   SA  ADD,MONT,SB
      JMP T,NNNEG,DSARL
      LDIM 0,4000
      RA  ADD,IMOP,SB
      JMP T,NNNEG,MCALL
      CAL F,F,GETOP

      DSABL:

      OR,ILIMC,WRK2
      SWAP WRK2,WRK2
      SAB  AND,M8,WRK2,WRK2
      ZBS  SUB,WRK2,ILIM
      LDIM 0,10
      RAB  OR,IMOP,STAT,STAT
      JMP  F,F,JL1

      MCALL:

      SUB,INF
      ZQS  SUB,INF
      RAB  ADD,IMOP,SB,CAUSE
      LDIM 0,40
      RABR CAND,IMOP,CNTR,CNTR,CNTR
      JMP  F,F,ICALL

; CALL ESCAPE ROUTINE
; IF SB >= -MONTOP
; THEN GOTO DSABL
; IF SB >= -4000
; THEN GOTO MCALL
; WRKO:= NEXT INSTRUCTION
; WRK1:= PHYSICAL ADDRESS
; SB:= LOGICAL ADDRESS

; ILIM:= 16 EXT 0 CON ILIMC(4:11)
; STAT(20):= 1, DISABLE
; GOTO JL1 (JUMP LINK)

; INF:= INF - 1
; CAUSE:= MON CALL NO
; AUTOLOADING:=0
; GOTO ICALL (INTERRUPT)

```

```

!0047
01      ;JUMP WITH INTERRUPT ENABLED, JE, OPCODE = 17
02
03 1324 0010000001760 JEE: LDIM 0,1760
04 1325 071002002226 CAL F,F,ESCAP
05
06 1326 071002000340 JE: CAL F,F,GETOP
07
08
09 1327 020730002000 SZQ OR,ILIMC
10 1330 020640005400 SQQ AND,M8
11 1331 011220006400 ZQS SUB,ILIM
12 1332 001000000010 LDIM 0,10
13 1333 033550000104 RAB CAND,IMOP,STAT,STAT
14 1334 0710000001270 JMP F,F,JL1

; CALL ESCAPE ROUTINE
; WRK0:=NEXT INSTRUCTION
; WRK1:=PHYSICAL ADDRESS
; SB:=LOGICAL ADDRESS
; ILIM:= 16 EXT 0 CON ILIMC(16:23)
; STAT(20):= 0, ENABLE
; GOTO JL1 (JUMP LINK)

```

!0048

;GET AND PUT GENERAL REGISTER INSTRUCTIONS
;*****

;GET GENERAL REGISTER, GG, OPCODE = 34

GGE: LDIM 0,3401
CAL F,F,ESCAP
GG: SRZA OR,SB,WRK3,Z,NL,F,F
CAL F,F,GETRG
ZQB OR,W RTN
; CALL ESCAPE ROUTINE
; WRK3:= SB/2
; Q:= REG(WRK3)
; W:= REG(SB), RTN

;PUT GENERAL REGISTER, GP, OPCODE = 57, PRIVILEGED INSTRUCTION

GPE: LDIM 0,5741
CAL F,F,ESCAP
GP: JMP F,M,MODE,INTO
ZBQ OR,W
SRZA OR,SB,WRK3,Z,NL,F,F
JMP F,F,PUTRG
; CALL ESCAPE ROUTINE
; IF NOT M-MODE THEN GOTO INTO
; Q:= W
; WRK3:= SB/2
; REG(WRK3):= Q, RTN

!0049

;INPUT-OUTPUT INSTRUCTIONS
;*****

;DATA IN, DI, OPCODE = 35, PRIVILEGED INSTRUCTION

07	1350	0010000003541	DIE:	LDIM	O,3541				
08	1351	071002002226		CAL	F,F,ESCAP				
09									; CALL ESCAPE ROUTINE
10	1352	071000102251	DI:	JMP	F,MODE,INTO				; IF NOT M-MODE THEN GOTO INTO
11	1353	031330020007		ZBR	OR,SB,READ				; I/O ADDR:= SB, READ
12	1354	031230000000		ZQ	OR				; NO-OP
13	1355	071012001366		WCAL	F,F,SETEX				; WAIT, CALL SETEX
14	1356	433730001400		RZB	OR,DATI,W				; W:= DATAIN, RTN
15									RTN

;DATA OUT, DO, OPCODE = 1, PRIVILEGED INSTRUCTION

19	1357	001000000141	DOE:	LDIM	O,0141				
20	1360	071002002226		CAL	F,F,ESCAP				; CALL ESCAPE ROUTINE
21									
22	1361	071000102251	DO:	JMP	F,MODE,INTO				; IF NOT M-MODE THEN GOTO INTO
23	1362	031330030000		ZBR	OR,W,DATO				; DATAOUT:= W
24	1363	031330420007		ZBR	OR,SB,WRT				; I/O ADDR:= SB, WRITE
25	1364	031230000000		ZQ	OR				; NO-OP
26	1365	071010001366		WJMP	F,F,SETEX				; WAIT, GOTO SETEX
27									

;SUBROUTINE SET EXCEPTION

; THE EXCEPTION BITS OF THE STATUS REGISTER,
; STAT(21:23) IS SET IN ACCORDANCE WITH THE BUSERRORS

34	1366	0010000000007	SETEX:	LDIM	O,7				
35	1367	033550000104		RAB	CAND,IMOP,STAT,STAT				; STAT(21:23) = 0,0,0
36	1370	071000271372		JMP	F,BPAR,NPAR				; IF NOT BUS PARITY THEN GOTO NPAR
37	1371	023501005104		SAB	ADDO,M2,STAT,STAT				; STAT(21):= 1
38	1372	071000251374	NPAR:	JMP	F,BTIM,NTIM				; IF NOT TIME OUT THEN GOTO NTIM
39	1373	023530004504		SAB	OR,C2,STAT,STAT				; STAT(22):= 1
40	1374	4710000661375	NTIM:	JMP	T,BNACK,NACK				; IF NACK THEN GOTO NACK ELSE RTN
41	1375	4333010000004	NACK:	ZBB	ADDO,STAT				; STAT(23):= 1, RTN

10050

-EOT

;FLOATING POINT INSTRUCTIONS
;*****

;CONVERT INTEGER TO FLOATING, CI, OPCODE = 40

CIE: LDIM 0,4001
CAL F,F,ESCAP

CI: ZBQ OR,W

ZQB OR,WRKO
ABB SUBO,WRK1,WRK1
LDIM 0,27
RAB ADD,IMOP,SB,WRK2

NORMA: CAND,M2,STAT,STAT
ZAB OR,WRK1
ZB OR,WRKO
JMP T,NZ,CINZ
ZQ OR

CIZER: T,NZ,CINZ
ZBB AND,WPRE
LDIM 0,4000
RZB OR,IMOP,W

CINZ: OR,WRKO
ZBB SUN,WRK2
NSZB OR,WRKO,NL,F,F
DLZB OR,WRKO,Z,NL,T,NORM
DSRB OR,WRKO,LNK,NL

ZQB OR,WRK1
LDIM 0,4000
RAB ADD,IMOP,WRK1,WRK1

; CIE:

; CALL ESCAPE(4001)

; CI:

; Q:=W

; WRKO:=Q

; WRK1:=0

; WRK2:=SB+23

; NORMALIZE:

; EX(22:23):=0

; Q:=WRK1

; IF WRKO=0 AND Q=0 THEN

; BEGIN

; W:=12EX10-2048

; WPRE:=0

; RETURN

; END;

; WHILE SHLINK <> WRKO(0) DO

; BEGIN

; SH LINK CON WRKO CON Q :=

; WRKO CON Q CON 1EXT0

; WRK2:=WRK2-1

; END

; WRKO CON Q:= SHLINK CON WRKO CON Q(0:22

; WRK1:=Q

; WRK1:=WRK1+ 8*4000

RTN

PUSH

LRTN

10051	ROUND:	ZAB	ADDC,WRKO,WRKO	;	ROUND:	
01 1426 033402000252	JMP	F,OVFL,ROUND1		;	IF OVERFLOW THEN	
02 1427 071000031435	SSRB	ADD,WRKO,SGN,NL		;	BEGIN	
03 1430 045300004012	SSRB	ADD,WRK1,LNK,NL		;	WRKO CON WRK1 :=	
04 1431 045300002013	LDIM	4000,0		;	1EXTWRKO(0) CON WRKO CON WRK1(0:22);	
05 1432 001100000000	RAB	EXOR,IMOP,WRKO,WRKO		;	WRKO(0):=1 EXOR WRKO(0);	
06 1433 033560000252	ZBB	ADD0,WRK2		;	WRK2:=WRK2+1;	
07 1434 033301000014				;	END;	
08	ROUND1:			;	WRKO:=WRKO+CARRY	
09 1435 2205500006260	SAQ	CAND,M12,WRK1		;	Q:=WRK1(0:11) CON 12EXT0	
10 1436 033310000014	ZBR	SUN,WRK2		;	WHILE SHLINK<>WRKO(0) DO	
11 1437 043330000012	NSZB	OR,WRKO,NL,F,F		;	BEGIN	
12				;	WRK2:=WRK2-1	
13				;	SHLINK CON WRKO CON Q :=	
14				;	WRKO CON Q CON 1EXT0	
15 1440 546330450012	DLZB	OR,WRKO,Z,NL,T,NORM		;		
16				;	END	

!0052

01 1441 0443300002012	DSRB	OR,WRK0,LNK,NL	;	WRK0:= SH LINK CON WRK0 CON Q(0:22)
02 1442 033430000241	ZAB	OR,WRK0,WPRES	;	WPRES:=WRK0
03 1443 0010000004002	LDIM	0,4002	;	WRK2:=WRK2+8,4002
04 1444 0335000000314	RAB	ADD,IMOP,WRK2,WRK2	;	
05 1445 0010000004000	LDIM	0,4000	;	WRK2:=WRK2 EXOR 8,4000
06 1446 0335600000314	RAB	EXOR,IMOP,WRK2,WRK2	;	
07 1447 0235400006312	SAB	AND,M12,WRK2,WRK0	;	WRK0:=12EXT0 CON WRK2(12:23)
08 1450 0215500006300	SA	CAND,M12,WRK2	;	IF WRK2(0:11) CON 12EXT0 <> THEN
09 1451 071000021456	JMP	F,NZ,CIENT	;	BEGIN
10 1452 0710000411454	JMP	T,NNEG,CIOFL	;	IF WRK2>0 THEN
11			;	EX(22:23):=2
12			;	ELSE
13			;	EX(22:23):=1
14 1453 0333100000004	ZBB	SUN,STAT	;	
15 1454 0235000004504	CIOFL:	ADD,C2,STAT,STAT	;	IF FLOATING POINT MASK THEN
16 1455 071009551457	JMP	T,FPMSK,INTF	;	GOTO INTF
17 1456 4330000000240	CIENT:	ADD,WRK0,W	;	END
18			;	W:=WRK0+Q
19			;	INTF:
20 1457 0010000000004	LDIM	0,4	;	CAUSE:=4
21 1460 0537300000006	RZB	OR,IMOP,CAUSE	;	W:=WRK0+Q
22 1461 0330000000240	AQB	ADD,WRK0,W	;	
23 1462 0710000002254	JMP	F,F,EXCEP	;	GOTO EXCEPTION
24			;	
25			;	
26			;	
27			;	
28 1463 0010000006501	CFE:	0,6501	;	CFE:
29 1464 071002002226	CAL	F,F,ESCAP	;	CALL ESCAPE(6501)
30 1465 0235500005104	SAB	CAND,M2,STAT,STAT	;	CF:
31 1466 0303300000000	ZBQ	OR,W	;	EX(22:23):=0
32			;	Q:=W
33 1467 0332300000013	ZQB	OR,WRK1	;	WRK1:=Q
34 1470 0327300000673	EXT	WRK1,WRK1	;	WRK1:= 12 EXT WRK1(12) CON WRK1(12:23)
35 1471 0331000000173	ABB	ADD,SB,WRK1	;	WRK1:=SB+WRK1
36 1472 0710000411474	JMP	T,NNEG,CFNZR	;	IF WRK1<0 THEN
37 1473 4333400000000	CFZER:	AND,W	;	BEGIN
			;	

CONVERT FLOATING TO INTEGER, CF, OPCODE = 65

28 1463 0010000006501	CFE:	0,6501	;	CFE:
29 1464 071002002226	CAL	F,F,ESCAP	;	CALL ESCAPE(6501)
30 1465 0235500005104	SAB	CAND,M2,STAT,STAT	;	CF:
31 1466 0303300000000	ZBQ	OR,W	;	EX(22:23):=0
32			;	Q:=W
33 1467 0332300000013	ZQB	OR,WRK1	;	WRK1:=Q
34 1470 0327300000673	EXT	WRK1,WRK1	;	WRK1:= 12 EXT WRK1(12) CON WRK1(12:23)
35 1471 0331000000173	ABB	ADD,SB,WRK1	;	WRK1:=SB+WRK1
36 1472 0710000411474	JMP	T,NNEG,CFNZR	;	IF WRK1<0 THEN
37 1473 4333400000000	CFZER:	AND,W	;	BEGIN
			;	

RTN

```

10053
01 1474 001000000027 CFNZR: LDIM 0,27
02
03
04 1475 033511000273 SUNO,IMOP,WRK1,WRK1
05 1476 071000011503 F,NNEG,CFSHI
06 1477 071000021503 F,NZ,CFSHI
07 1500 031330000001 OR,WPRE
08
09 1501 071000021473 F,NZ,CFZER
10 1502 071000001517 F,F,CFOVF
11
12 1503 0236500006012 CAND,M12,WRKO
13 1504 0303300000001 OR,WPRE
14 1505 0332300000000 OR,W
15 1506 0303300000012 OR,WRKO
16 1507 0313300000013 OR,WRK1
17 1510 271000021513 F,NZ,CFO
18 1511 0333010000013 ADDO,WRK1
19 1512 544300024000 ADD,W,SGN,NL,F,NZ
20 1513 0312000000000 ZQ
21 1514 471000011515 F,NNEG,CF1
22 1515 0333010000000 ADDO,W
23 1516 4710000431517 T,OVFL,CFOVF
24 1517 0235000004504 ADD,C2,STAT,STAT
25 1520 4710000542253 T,IMSK,INTA
26
27
28
29
30
31 1521 0010000006212 FME: LDIM
32 1522 071002002226 F,F,ESCAP CAL
33
34 1523 0710020000727 FM: CAL
35
36 1524 0327300000674 EXT
37 1525 0303300000000 ZBQ
38 1526 0236500006017 SQB
39 1527 0332300000015 ZQB
40 1530 0327300000735 EXT
41 1531 0331010000334 ARB
;MULTIPLY FLOATING, FM, OPCODE =62
FME: LDIM 0,6212
F,F,ESCAP
FM: F,F,GTDDWD
WRK1,WRK2
OR,W
CAND,M12,WRK5
OR,WRK3
WRK3,WRK3
ADDO,WRK3,WRK2

```

10054	01 1532	030330000001	ZBQ	OR, WPRE	Q:=WPRE	
	02 1533	0332300000016	ZQB	OR, WRK4	WRK4:=Q	
	03 1534	023550006273	SAB	CAND, M12, WRK1, WRK1	WRK1:=WRK1(0:11) CON 12EXT0	
	04 1535	023540006255	SAB	AND, M12, WRK0, WRK3	WRK3:=12EXT0 CON WRK0(12:23)	
	05 1536	033100000275	ABB	ADD, WRK1, WRK3	WRK3:=WRK3+WRK1	
	06 1537	032730001335	SWAP	WRK3, WRK3	WRK3:=WRK3(12:23) CON WRK3(0:11)	
	07 1540	030330000015	ZBQ	OR, WRK3	Q:=WRK3	
	08 1541	023550006252	SAB	CAND, M12, WRK0, WRK0	WRK0:=WRK0(0:11) CON 12EXT0	
	09 1542	032730001253	SWAP	WRK0, WRK1	WRK1:=WRK0(12:23) CON WRK0(0:11)	
	10 1543	033121000252	ABB	SUB0, WRK0, WRK0	WRK0:=0	
	11 1544	001000000042	LDIM	0, 42	CARRY:=0; ADDCOND:=Q(23);	
	12 1545	033730000015	RZB	OR, IMOP, WRK3	WRK1 CON Q:=1EXT0 CON WRK1 CON Q(0:22)	
	13				FOR COUNT:=35 STEP -1 UNTIL 0 DO	
	14				BEGIN	
	15 1546	0443300000413	DSRB	OR, WRK1, Z, MC	IF ADDCOND=1 THEN	
	16 1547	2531000000373	MULN	ADD, WRK5, WRK1, NL, F, F	BEGIN	
	17 1550	055102004352	MULS	ADDC, WRK4, WRK0, SGN, NL, F, F	WRK1:=WRK1+WRK5	
	18 1551	0443300002413	DRZB	OR, WRK1, LNK, MC, F, F	WRK0:=WRK0+WRK4+C	
	19 1552	033310000015	ZBB	SUN, WRK3	END	
	20 1553	553100020373	MULN	ADD, WRK5, WRK1, NL, F, NZ	WRK0:=1EXTWRK0(0) CON WRK0(0:22)	
	21 1554	055102004352	MULS	ADDC, WRK4, WRK0, SGN, NL, F, F	ADDCOND:=Q(23)	
	22 1555	0443300002413	DRZB	OR, WRK1, LNK, MC, F, F	WRK1 CON Q:=SHLINK CON WRK1 CON Q(0:22)	
	23				END	
	24				IF ADDCOND=1 THEN	
	25				BEGIN	
	26 1556	0531110000373	MULN	SUN0, WRK5, WRK1, NL, F, F	WRK1:=WRK1-WRK5	
	27 1557	055112004352	MULS	SUNC, WRK4, WRK0, SGN, NL, F, F	WRK0:=WRK0-WRK4-1+C	
	28 1560	0443300002013	DRZB	OR, WRK1, LNK, NL, F, F	END;	
	29				WRK0:=1EXTWRK0(0) CON WRK0(0:22);	
	30				WRK1 CON Q:=SHLINK CON WRK1 CON Q(0:22)	
	31 1561	071000001405	JMP	F, F, NORMA	GOTO NORMALIZE	

Address	OpCode	OpName	OpData	OpComment
0055				
01				
02				
03	1562	LDIM	0,6012	FAE: CALL ESCAPE(6012)
04	1563	CAL	F,F,ESCAP	FA: Q:=3 C,ADD
05	1564	SZQ	OR,M2	GOTO FSALI
06	1565	JMP	F,F,FSALI	
07				
08				
09				
10				
11	1566	LDIM	0,6112	FSE: CALL ESCAPE(6112)
12	1567	CAL	F,F,ESCAP	Q:=2 C, SUB
13	1570	SZQ	OR,C2	
14				
15	1571	DSRB	OR,WRK5,Z,MC	FSALI:
16	1572	CAL	F,F,GTOWD	ADDCONDITION:=Q(23)
17	1573	EXT	WRK1,WRK2	GET DOUBLEWORD(WRK0,WRK1)
18				
19	1574	SAB	CAND,M12,WRK1,WRK5	WRK2:=12EXT WRK1(12) CON WRK1(12:23)
20	1575	ZBQ	OR,W	WRK5:=WRK1(0:11) CON 12EXTO
21	1576	ZQB	OR,WRK1	Q:=W
22	1577	EXT	WRK1,WRK3	WRK1:=Q
23	1600	SQB	CAND,M12,WRK1	WRK3:=12EXT WRK1(12) CON WRK1(12:23)
24	1601	ZAB	OR,WRK0,WRK4	WRK1:=Q(0:11) CON 12EXTO
25	1602	ZBQ	OR,WPRE	WRK4:=WRK0
26	1603	ZQB	OR,WRK0	Q:=WPRE
27	1604	ABB	SUNO,WRK2,WRK3	WRK0:=Q
28	1605	JMP	F,NNEG,FSASM	WRK3:=WRK3-WRK2
29	1606	JMP	F,NZ,FSAEQ	IF WRK3>0 THEN
30	1607	ABB	ADD,WRK3,WRK2	BEGIN
31	1610	LDIM	0,46	WRK2:=WRK3+WRK2
32	1611	RA	SUNO,IMOP,WRK3	IF WRK3>= 38 THEN
33	1612	JMP	F,NNEG,FSASH	RETURN
34	1613	ZAQ	OR,WRK5	ELSE
35	1614	ZBB	SUN,WRK3	REGIN
36	1615	DRZB	ADD,WRK4,SGN,NL,F,NZ	Q:=WRK5
37				FOR WRK3:=WRK3 STEP -1 UNTIL 0 DO
38				WRK0 CON Q :=
39	1616	ZQB	OR,WRK5	WRK4(0) CON WRK4 CON Q(0:22)
40	1617	JMP	F,F,FSAEQ	WRK5:=Q
41				END
42				END ELSE
43				IF WRK3<0 THEN
44				BEGIN

ADDRESS	INSTR	OPERANDS	CODE	COMMENT
01	LDIM	RA	0010C00000045	
02	RA		031500000320	
03	JMP		071000011640	
04	ZBQ		230330000013	
05	ZBB		033301000015	
06	DRZB		544300024012	
07				
08	ZQB		033230000013	
09				
10				
11	DIVN		0631030000373	
12	DIVN		0631020000352	
13	JMP		071000031405	
14	SSRB		045300004012	
15	SSRB		045300002013	
16	LDIM		001100000000	
17	RAB		033560000252	
18	ZBB		033301000014	
19	JMP		071000001405	
20				
21	ZBB		033340000013	
22	ZBB		033340000012	
23	JMP		071000001627	
24				

0057

; FLOATING DIVIDE, FD, OPCODE = 64

```

1 1643 0010000006412 FDE: LDIM 0,6412
2 1644 071002002226 F, F, ESCAP
3
4
5 1645 0235500005104 FD: CAND, M2, STAT, STAT
6 1646 071002000727 F, F, GTDWD
7 1647 0327300000674 WRK1, WRK2
8 1650 0235500006277 CAND, M12, WRK1, WRK5
9 1651 0334300000256 OR, WRK0, WRK4
1 1652 071000421655 T, NZ, FDO
2 1653 023500004504 ADD, C2, STAT, STAT
3 1654 471000551457 T, FPMSK, INTF
4 RTN
5 1655 0313300000001 FDO: OR, WPRE
6 1656 071000021413 JMP F, NZ, CIZER
7 1657 0303300000000 ZBQ OR, W
8 1660 0236500006000 SQB CAND, M12, W
9 1661 0332300000013 ZQB OR, WRK1
0 1662 0327300000673 EXT WRK1, WRK1
1 1663 0331210000274 ABB SUB0, WRK1, WRK2
2 1664 0433300001416 NSZB OR, WRK4, DVS, F, F
3 1665 0433300001001 NSZB OR, WPRE, DC, F, F
4 1666 0302400000000 ZQQ AND
5 1667 0010000000027 LDIM 0,27
6 1670 0337000000015 RZB ADD, IMOP, WRK3
7 1671 2661030004360 DIVD ADDX, WRK5, W, ADC, NL, F, F
8 1672 0671020003341 DIVS ADDC, WRK4, WPRE, LNK, DC, F, F
;
; FDE:
; CALL ESCAPE(6412)
;
; EX(22:23):=0
; GET DOUBLEWORD(WRK0, WRK1)
; WRK2:=12EXT WRK1(12) CON WRK1(12:23)
; WRK5:=WRK1(0:11) CON 12EXT0
; WRK4:=WRK0
; IF WRK4=0 THEN
; EX(22:23):=2
; IF FLOATING POINT MASK THEN GOTO INTF
; ELSE RETURN
; IF WPRE=0 THEN
; GOTO CIZER
; Q:=W
; W:=Q(0:11) CON 12EXT0
; WRK1:=Q
; WRK1:=12 EXT WRK1(12) CON WRK1(12:23)
; WRK2:=WRK1-WRK2
; DIVSIGN:=WRK4(0)
; ADC:=WPRE(0) EXOR --DIVSIGN
; Q:=0
; FOR WRK3:=24 STEP -1 UNTIL 0 DO
; BEGIN
; IF ADC=1 THEN
; BEGIN

```

10058

01 1673 033310000015
02 1674 566103024360
03
04
05
06
07
08
09

10 1675 067102003341
11 1676 001100000000
12 1677 033230000012
13 1700 0335600000252
14 1701 071000451706
15 1702 047330004012
16 1703 067103000360
17 1704 067102003341
18 1705 033310000014
19

FD1:

20 1706 020730006000
21 1707 001000000013
22 1710 033700000015
23 1711 266103004360
24 1712 067102003341
25 1713 033310000015
26 1714 566103024360
27 1715 067102003341
28 1716 033230000013
29 1717 032730001273
30 1720 033303000013
31 1721 071000001426
32

ZBB
DIVD
SUN,WRK3
ADDX,WRK5,W,ADC,NL,F,NZ LRTN

DIVS
LDIM
ZQB
RAB
JMP
SSLB
DIVS
DIVS
ZBB
ADDC,WRK4,WPRES,LNK,DC,F,F
4000,0
OR,WRKO
EXOR,IMOP,WRKO,WRKO
T,NORM,FD1
OR,WRKO,ADC,NL
ADDX,WRK5,W,Z,NL,F,F
ADDC,WRK4,WPRES,LNK,DC,F,F
SUN,WRK2

SZQ
LDIM
RZB
DIVD
DIVS
ZBB
DIVD
DIVS
ZQB
SWAP
ZBB
JMP
OR,M12
O,13
ADD,IMOP,WRK3
ADDX,WRK5,W,ADC,NL,F,F
ADDC,WRK4,WPRES,LNK,DC,F,F
SUN,WRK3
ADDX,WRK5,W,ADC,NL,F,NZ LRTN
ADDC,WRK4,WPRES,LNK,DC,F,F
OR,WRK1
WRK1,WRK1
ADDX,WRK1
F,F,ROUND

W:=W-WRK5
WPRE:=WPRE-WRK4-1+CARRY
END ELSE
BEGIN
W:=W+WRK5
WPRE:=WPRE+WRK4+CARRY
END
SHLINK CON WPRE CON W CON Q:=
WPRE CON W CON Q CON 1EXTADC
ADC:=WPRE(0) EXOR -.DIVSIGN
END
WRKO:=Q
WRKO:=WRKO EXOR 4000,0
IF WRKO NOT NORMALISED THEN
BEGIN
SH LINK CON WRKO:=WRKO CON ADC
'DIVIDE'
WRK2:=WRK2-1
END
Q:=8'7777
FOR WRK3:=12 STEP -1 UNTIL 0 DO
'DIVIDE'

WRK1:=Q
WRK1:=WRK1(0:11) CON WRK1(12:23)
WRK1:=WRK1+ADC
GOTO ROUND

```

00059
01
02
03
04
05 1722 001000000012
06 1723 033730000006
07 1724 071000001744
08
09 1725 001000000014
10 1726 030730000000
11 1727 020621002400
12 1730 011220002400
13
14 1731 001000000014
15 1732 033730000007
16 1733 023500002567
17 1734 071000001742
18
19 1735 023730002407
20 1736 071000001742
21
22 1737 023511000714
23
24
25 1740 033430000307
26 1741 033430000265
27
28 1742 001000000010
29 1743 033730000006
30 1744 001173770010
31 1745 033540000104
32 1746 071002671752
33 1747 071002661754
34 1750 071000651756
35 1751 071000001757

;BUS INTERRUPTS
;*****
FINT:
LDIM 0,12
RZB OR,IMOP,CAUSE
JMP F,F,INTBF
INTB4:
LDIM 0,14
RZQ OR,IMOP
SQQ SUBO,INF
ZQS SUB,INF
INTB0:
LDIM 0,14
RZB OR,IMOP,SB
SAB ADD,INF,SB,SB
JMP F,F,INTB
INTB1:
SZB OR,INF,SB
JMP F,F,INTB
INTB2:
SAB SUNO,BASE,WRK2,WRK2
INTB3:
ZAB OR,WRK2,SB
ZAB OR,WRK1,IC
INTB:
LDIM 0,10
RZB OR,IMOP,CAUSE
INTBF:
LDIM 7777,10
RAB AND,IMOP,STAT,STAT
CAL T,BPAR,PARIT
CAL T,BNACK,REJECT
JMP T,BTIM,TIMUT
JMP F,F,TSTIN
; FINT: C. BUS ERROR DURING FETCH
; CAUSE:=10;
; GOTO INTBF
; INTB4:
; INF:=INF-12;
; INTB0:
; SB:=12+INF
; GOTO INTB
; INTB1:
; SB:=INF C. GET EXC/ESC ADDRESS
; GOTO INTB
; INTB2:
; DUMPADDR:=DUMPADDR-BASE
; C. FROM FETCH FIRST INSTRUCTION
; INTB3:
; SB:=DUMPADDR; IC:=OLDIC
; C. FROM RE AND RI
; INTB:
; CAUSE:=8;
; CLEAR EX BUT BIT 20
; IF BUSPARITY THEN PARITY
; IF BUS NACK THEN REJECT
; IF BUS TIMEOUT THEN GOTO TIMEOUT
; GOTO TESTINT

```

LDIM	PARIT:	0,4	RTN	PARITY:	RETURN
RAB		ADD,IMOP,STAT,STAT		EX(21):=1	
	REJCT:			REJECT:	
LDIM		0,1	RTN	EX(23):=1	RETURN
RAB		ADD,IMOP,STAT,STAT		TIMEOUT:	
	TIMUT:			EX(22):=1	
SAB		ADD,C2,STAT,STAT		TESTINT:	
	TSTIN:			IF ILIMC<>0 THEN	
SZN		OR,ILIMC		GOTO ICALL	
JMP		T,NZ,ICALL		ELSE GOTO SYSTEMFAULT	
JMP		F,F,SYFLT			

!0061

;EXTERNAL INTERRUPTS AND MONITOR CALL
;*****

ICALL:

01			
02			
03			
04			
05	1762	071010001763	
06	1763	020730022400	
07	1764	023600004416	
08	1765	020730003000	
09	1766	011220001400	
10	1767	001000000010	
11	1770	030730000000	
12	1771	011220001000	
13	1772	011120000652	
14	1773	071010640051	
15			
16	1774	033730001417	
17	1775	031330020016	
18	1776	001000000005	
19	1777	033511000353	
20	2000	001000000016	
21	2001	033511000356	
22	2002	001000000001	
23	2003	033530000356	
24	2004	011320002416	
25	2005	033340000015	
26	2006	071010642053	
27	2007	033730001414	
28			

WJMP	
SZQR	
SQB	
SZQ	
ZQS	
LDIM	
RZQ	
ZQS	
ABS	
WJMP	
RZB	
ZBR	
LDIM	
RAB	
LDIM	
RAB	
LDIM	
RAB	
ZBS	
ZBB	
WJMP	
RZB	

F,F,+.3	
OR,INF,READ	
ADD,C2,WRK4	
OR,SIZE	
SUB,ULIM	
O,10	
OR,IMOP	
SUB,LLIM	
SUB,WRK0,WRK0,BASE	
T,BERR,SYFLT	
OR,DATI,WRK5	
OR,WRK4,READ	
O,5	
SUNO,IMOP,WRK4,WRK1	
O,16	
SUNO,IMOP,WRK4,WRK4	
O,1	
OR,IMOP,WRK4,WRK4	
SUB,WRK4,INF	
AND,WRK3	
T,BERR,DMPFL	
OR,DATI,WRK2	

```

; ICALL: C. EXT. INTERRUPT AND MON CALL
; WAIT BUSS READY
; Q:=READ:=INF C. READ NEW STATUS
; WRK4:=INF+2 C. DUMPADDRESS
; Q:=SIZE
; ULIM:=Q
;
; LLIM:=8
; BASE:=0
; WAIT, IF IOERROR THEN
; GOTO SYSTEMFAULT
; WRK5:=DATAIN
; READ DUMPADDRESS
; WRK1:=INF-3
; C. NEW IC
;
; WRK4:=INF-12 (0:22) CON 1EXT1
; INF:=WRK4
; WRK3:=0
; WAIT, IF IOERROR THEN
; GOTO DUMPFAULT
; WRK2:=DUMPADDRESS

```

10062

01

02

03 2010 071010642053

04 2011 001000002730

05 2012 031500070320

06 2013 071003000000

07 2014 031230030000

08 2015 031330420014

09 2016 023500004714

10 2017 033401000335

11 2020 001000000010

12 2021 031520000320

13 2022 071000412010

14 2023 071010642053

15 2024 031330020013

16 2025 020540006360

17 2026 011220002000

18 2027 011220006400

19 2030 020550016360

20 2031 071010642053

21 2032 033730001405

22 2033 033730221410

23 2034 023540006104

24 2035 033000000104

25 2036 001000020000

26 2037 031730060000

27 2040 033430000142

ISAVR:

WJMP

LDIM

RAR

CAX

ZQR

ZBR

SAB

ZAB

LDIM

RA

JMP

WJMP

ZBR

SAQ

ZQS

ZQS

SAQR

WJMP

RZB

RZBR

SAB

AQB

LDIM

RZR

ZAB

T,BERR,DMPFL

O,GRTAB/3

ADD,IMOP,WRK3,MIX

F,F

OR,DATO

OR,WRK2,WRT

ADD,C2,WRK2,WRK2

ADD,WRK3,WRK3

O,10

SUB,IMOP,WRK3

T,NNEG,ISAVR

T,BERR,DMPFL

OR,WRK1,READ

AND,M12,WRK5

SUB,ILIMC

SUB,ILIM

CAND,M12,WRK5,CPUST

T,BERR,DMPFL

OR,DATI,IC

OR,DATI,PC,READP

AND,M12,STAT,STAT

ADD,STAT,STAT

2,0

OR,IMOP,IR

OR,CAUSE,X

ISAVR:

FOR WRK3:=0 STEP 1 UNTIL 8 DO

BEGIN

WAIT, IF IOERROR THEN

GOTO DUMPFPAULT

Q:=GETREG(GETTAB+WRK3);

DATAOUT:=Q;

WRT:=WRK2 C. DUMP REGISTER

WRK2:=WRK2+2;

END;

WAIT, IF IOERROR THEN GOTO DUMPFPAULT

READ:=WRK1 C. READ NEW IC

Q:=WRK5(12:23)

ILIMC:=Q

ILIM:=Q

CPUST:=Q:=WRK5(0:11) CON 12EXT0

WAIT, IF IOERROR THEN GOTO DUMPFPAULT

IC:=DATAIN

PC:=DATAIN , READP

STAT:=Q+STAT(12:23);

IR(X-FIELD):=2;

W(IR(XFIELD)):=CAUSE C. W2

10063

```

01
02 2041 0010000007000
03 2042 031540000100
04 2043 071000022047
05 2044 023550006104
06 2045 001000000004
07 2046 033730000002
08 2047 001000010000
09 2050 031730060000
10 2051 033430000302
11 2052 071000000246
12
13
14 2053 0010000001000
15 2054 033500000104
16 2055 0010000007000
17 2056 031540000100
18 2057 071000020051
19 2060 0710000001757

```

```

LDIM
RA
JMP
SAB
LDIM
RZB
LDIM
RZR
ZAB
JMP

DMPFL:
LDIM
RAB
LDIM
RA
JMP
JMP

```

```

0,7000
AND,IMOP,STAT
F,NZ,DMPK
CAND,M12,STAT,STAT
0,4
OR,IMOP,X
1,0
OR,IMOP,IR
OR,WRK2,X
F,F,MLOP1

DMPK:
W1:=WRK2 C. TOP DUMP AREA;
GOTO MLOP1

DMPFAULT:
INCREASE DUMPERRORCOUNT
IF DUMPERROR OVERRUN THEN
GOTO SYSTEMFAULT
ELSE
GOTO TESTINT

```

```

;
; IF DUMPERRORCOUNT <> 0 THEN
; BEGIN
; EX:=0;
; W2:=4
; END;
; DUMPK:
; W1:=WRK2 C. TOP DUMP AREA;
; GOTO MLOP1
;
; DMPFAULT:
; INCREASE DUMPERRORCOUNT
; IF DUMPERROR OVERRUN THEN
; GOTO SYSTEMFAULT
; ELSE
; GOTO TESTINT
;

```

```
; RETURN INTERRUPT, RI, OPCODE = 14
```

Address	Instruction	Comment
04 2061	0010000001472	RIE:
05 2062	071002002226	RI:
06		
07 2063	071000102251	
08		
09 2064	0010000000016	
10 2065	0307300000000	
11 2066	020600022400	
12 2067	0334300000172	
13 2070	0334300000133	
14 2071	071010641731	
15 2072	0337300001414	
16 2073	0333400000015	
17 2074	0010000000010	
18 2075	0337300000017	
19 2076	071002002147	
20 2077	0331000000254	
21 2100	0010000000020	
22 2101	0335110000314	
23 2102	0010000000015	
24 2103	0337300000017	
25 2104	071002002147	
26 2105	0010000000014	
27 2106	0307300000000	
28 2107	0206000002400	
29 2110	0112200002400	
30		
31 2111	0010000007760	
32 2112	033550000104	
33 2113	0237300002015	
34 2114	0010000000010	
35 2115	031540000100	
36 2116	071000022120	
37 2117	0327300001335	
38 2120	0235400006335	
39 2121	0113200006415	
40 2122	031330010004	

10065

01	02	2123	023500220530	JUMP:	SABR	ADD,BASE,IC,PC,READP	;	JUMP:	PC:=IC+BASE , READP	;
02	03	2124	001000000001		LDIM	0,1	;			
03	04	2125	033550000125		RAB	CAND,IMOP,IC,IC	;		IC:=IC(0:22) CON 1EXT0	;
04	05	2126	071000602140		JMP	T,NMADR,JUMP1	;		IF PC<8 THEN	;
05	06	2127	021511001200				;		GOTO JUMP1;	;
06	07	2128	071000012137		SA	SUNO,LLIM,PC	;		IF PC<LLIM THEN	;
07	08	2129	021520001600		JMP	F,NNEG,JUMPX	;		GOTO JUMPX;	;
08	09	2130	071000012137		SA	SUB,ULIM,PC	;		IF PC>=ULIM THEN	;
09	10	2131	071000012251		JMP	F,NNEG,INT0	;		GOTO INT0;	;
10	11	2132	071010641737		WJMP	T,BERR,INTB2	;		WAIT, IF IOERROR THEN GOTO INTB2;	;
11	12	2133	071000700260		JMP	T,INTR,INIT	;		IF INTERRUPT THEN	;
12	13	2134	033730001412				;		GOTO INTT;	;
13	14	2135	071000000251		RZB	OR,DATI,WRKO	;		WRKO:=DATAIN;	;
14	15	2136	071000000251		JMP	F,F,AFTCH	;		GOTO AFTER FEICH	;
15	16	2137	071010002140	JUMPX:			;		JUMPX:	;
16	17	2138	033430220130		WJMP	F,F,+3	;		WAIT, IOREADY;	;
17	18	2139	031420000120	JUMP1:	ZABR	OR,IC,PC,READP	;		PC:=IC , READP;	;
18	19	2140	071000412251		ZA	SUB,IC	;		IF IC<0 THEN	;
19	20	2141	021511000120		JMP	T,NNEG,INT0	;		GOTO INT0	;
20	21	2142	071000412251		SA	SUNO,CPA,IC	;		IF IC>=CPA THEN	;
21	22	2143	071000412251		JMP	T,NNEG,INT0	;		GOTO INT0	;
22	23	2144	071000700260		JMP	T,INTR,INIT	;		IF INTERRUPT THEN GOTO INTT;	;
23	24	2145	071000000246		JMP	F,F,MLOP1	;		GOTO MLOP1	;

10066

01	2147	031330020014	GETRE:	ZBR	OR,WRK2,READ	; PROCEDURE GETREG(WRK2,WRK3)
02	2150	032730001336		SWAP	WRK3,WRK4	; GETREGREP;
03	2151	031430060340		ZAR	OR,WRK4,IR	; READ(WRK2)
04	2152	023500004714		SAB	ADD,C2,WRK2,WRK2	; IR(8:11):=WRK3(20:23)
05	2153	001000002774		LDIM	O,PRTAB/3	; C. FOR INDIRECT ADDRESSING OF W(0:3)
06	2154	031500070320		RAR	ADD,IMOP,WRK3,MIX	; WRK2:=WRK2+2;
07	2155	033301000015		ZBB	ADDO,WRK3	; MIX:=WRK3+PUTREG TABLE
08	2156	071010641740		WJMP	T,BERR,INTB3	; WRK3:=WRK3+1;
09						; WAIT, IF IOERROR THEN
10						; GOTO INTB3
11	2157	030730001400		RZQ	OR,DATI	; Q:=DATAIN
12	2160	071003000000		CAX	F,F	; PUTREG(MIX,Q);
13						; IF WRK3<REGTOP THEN
14	2161	031121000375		AB	SUBO,WRK5,WRK3	; GOTO GETREGREP
15	2162	471000422147		JMP	T,NZ,GETRE	; ELSE
16						; RETURN

RTN
;

!0067

; RETURN FROM ESCAPE, RE, OPCODE = 26

01

02

03

04 2163 001000002632

05 2164 071002002226

06 2165 033430000174

07

08 2166 033430000133

09 2167 033340000015

10 2170 023500000714

11

12 2171 031330220014

13 2172 071000602250

14 2173 021511001300

15 2174 071000012217

16 2175 021520001700

17 2176 071000012250

18 2177 023500004714

19 2200 001000002774

20 2201 0315000070320

21 2202 032730001336

22 2203 031430060340

23 2204 071010641737

24

25 2205 030730001400

26 2206 071003000000

27 2207 033401000335

28 2210 001000000010

29 2211 031511000320

30 2212 071000012171

31 2213 071000502111

32 2214 001100000000

33 2215 033550000104

34 2216 071000002111

35

36

37 2217 071010002220

38 2220 023511220714

39 2221 071000602250

40 2222 021511000300

41 2223 071000412250

42 2224 023500000714

43 2225 071000002177

```

; REE:
; CALL ESCAPE(2632)
; RE:
; WRK2:=SB C. DUMP ADDRESS
; WRK1:=IC C. OLDIC
; WRK3:=0
; WRK2:=WRK2+BASE;
; RREP:
; READP:=WRK2
; IF IO-ADDR<8 THEN GOTO INTRE;
; IF IO-ADDRESS<LLIM THEN
; GOTO TEST;
; IF IO-ADDRESS>=ULIM THEN
; GOTO INTRE;
; WRK2:=WRK2+2;
;
; MIX:=PRTAB+WRK3;
; IR(X-FIELD):=
; WRK3;
; WAIT, IF BUSERROR THEN
; GOTO INTB2
; Q:=DATAIN;
; PUTREG(MIX,Q);
; WRK3:=WRK3+1;
; IF WRK3<8 THEN
; GOTO RREP;
; IF MONMODE THEN
; CLEAR MONMODE IN STATUS
; GOTO REGSET;
;
; TST:
; WAIT, IO-READY;
; DUMPADDR:=DUMPADDR+BASE, READP;
; IF WRK2<8 THEN GOTO INTRE;
; IF WRK2>=CPA THEN
; GOTO INTRE;
; WRK2:=WRK2+BASE;
; GOTO WAIT;

```

```

; RETURN FROM ESCAPE, RE, OPCODE = 26

REE: LDIM 0,2632
CAL F,F,ESCAP
ZAB OR,SB,WRK2

ZAB OR,IC,WRK1
ZBB AND,WRK3
SAB ADD,BASE,WRK2,WRK2

RREP: OR,WRK2,READP
T,NMADR,INTRE
SUNO,LLIM,WRK2
F,NNEG,TST
SUB,ULIM,WRK2
F,NNEG,INTRE
ADD,C2,WRK2,WRK2
O,PRTAB/3
ADD,IMOP,WRK3,MIX
WRK3,WRK4
OR,WRK4,IR
T,BERR,INTB2

OR,DATI
F,F
ADD,WRK3,WRK3
O,10
SUNO,IMOP,WRK3
F,NNEG,RREP
T,MMODE,RGSET
4000,0
CAND,IMOP,STAT,STAT
F,F,RGSET

F,F,++3
SUNO,BASE,WRK2,WRK2,READP
T,NMADR,INTRE
SUNO,CPA,WRK2
T,NNEG,INTRE
ADD,BASE,WRK2,WRK2
F,F,WAIT

```

```

REE: LDIM 0,2632
CAL F,F,ESCAP
ZAB OR,SB,WRK2

ZAB OR,IC,WRK1
ZBB AND,WRK3
SAB ADD,BASE,WRK2,WRK2

RREP: OR,WRK2,READP
T,NMADR,INTRE
SUNO,LLIM,WRK2
F,NNEG,TST
SUB,ULIM,WRK2
F,NNEG,INTRE
ADD,C2,WRK2,WRK2
O,PRTAB/3
ADD,IMOP,WRK3,MIX
WRK3,WRK4
OR,WRK4,IR
T,BERR,INTB2

OR,DATI
F,F
ADD,WRK3,WRK3
O,10
SUNO,IMOP,WRK3
F,NNEG,RREP
T,MMODE,RGSET
4000,0
CAND,IMOP,STAT,STAT
F,F,RGSET

F,F,++3
SUNO,BASE,WRK2,WRK2,READP
T,NMADR,INTRE
SUNO,CPA,WRK2
T,NNEG,INTRE
ADD,BASE,WRK2,WRK2
F,F,WAIT

```

10068

```

01 ; PROCEDURE ESCAPE ; IMOP=CAUSE
02 ;*****
03 ;*****
04 ;*****
05 ESCAP:
06 2226 0337300000006
07 2227 071000132232
08 2230 001010000000
09 2231 433550010104
10
11
12 2232 0327300011114
13 2233 033140000154
14 2234 001000000077
15 2235 033540000314
16 2236 471000422237
17 2237 001000000006
18 2240 030730000000
19 2241 023600022415
20 2242 001010000000
21 2243 033530000104
22 2244 001100000000
23 2245 033730000017
24 2246 023511004525
25 2247 071000002263
26
27
28 2250 033430000265
29
30 2251 033340000006
31 2252 071000002254
32
33
34
35
36 2253 023730004406

```

; PROCEDURE ESCAPE ; IMOP=CAUSE

;*****

ESCAP:

RZB OR,IMOP,CAUSE
JMP F,AFESC,ESC1
LDIM 0400,0
RABR CAND,IMOP,STAT,STAT,CPUST RTN

ESC1:

SWAP STAT,WRK2
ABB AND,CAUSE,WRK2
LDIM 0,77
RAB AND,IMOP,WRK2,WRK2
JMP T,NZ,ESC2

ESC2:

LDIM 0,6
RZQ OR,IMOP
SQBR ADD,INF,WRK3,READ
LDIM 0400,0
RAB OR,IMOP,STAT,STAT
LDIM 4000,0
RZB OR,IMOP,WRK5
SAB SUNO,C2,IC,IC
JMP F,F,ESC3

INTRE:

ZAB OR,WRK1,IC

INTO:

ZBB AND,CAUSE
JMP F,F,EXCEP

INTA:

INTS:

INT1:

SZB OR,C2,CAUSE

; ESCAPE:

; CAUSE:=IMOP;
; IF AFTER ESCAPE THEN
; BEGIN
; CLEAR AFTER ESCAPE;
; CPUST:=STATUS,RETURN;
; END;
; WRK2:=STAT(12:23) CON STAT(0:11);
; WRK2:=CAUSE AND WRK2;
; WRK2:=WRK2 AND 8'77;
; IF WRK2 =0 THEN
; RETURN
; C. INSTRUCTION TYPE NOT INCLUDED
; WRK3:=INF+6, READ
; C. READ ESCAPE ADDRESS
; STAT(3):=1 C. AFTER ESCAPE;
; MASK:=WRK5:=1B0 C. MON MODE;
; IC:=IC-2 C. PREPARE FOR REPEAT
; GOTO ESC3

; INTRE: C. EXCEPTION RE

; IC:=OLDIC C. WRK1

; INTO:

; CAUSE:=0
; GOTO EXCEPTION;

; INT1: C. INTEGER EXCEPTION

; CAUSE:=1*2

[illegible]


```

!0071
01 ;ESCAPE ON UNASSIGNED OPERATION CODES
02 ;*****
03 ;*****
04 2357 001000000040 F00E: LDIM 0,0040 ;OPCODE=00
05 2360 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
06
07 2361 001000003640 F36E: LDIM 0,3640 ;OPCODE=36
08 2362 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
09
10 2363 001000003740 F37E: LDIM 0,3740 ;OPCODE=37
11 2364 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
12
13 2365 001000006340 F63E: LDIM 0,6340 ;OPCODE=63
14 2366 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
15
16 2367 001000007240 F72E: LDIM 0,7240 ;OPCODE=72
17 2370 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
18
19 2371 001000007340 F73E: LDIM 0,7340 ;OPCODE=73
20 2372 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
21
22 2373 001000007440 F74E: LDIM 0,7440 ;OPCODE=74
23 2374 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
24
25 2375 001000007540 F75E: LDIM 0,7540 ;OPCODE=75
26 2376 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
27
28 2377 001000007640 F76E: LDIM 0,7640 ;OPCODE=76
29 2400 071000002402 JMP F,F,ESUNA ;GOTO ESUNA
30
31 2401 001000007740 F77E: LDIM 0,7740 ;OPCODE=77
32
33 2402 071002002226 ESUNA: CAL F,F,ESCAP ;CALL ESCAPE ROUTINE
34 2403 071000002251 JMP F,F,INT0 ;GOTO INT0

```

.EOT

;*****

; CPU 810 TCP ROUTINE

;*****

```

10 2404 071002002455      TCPIN:  CAL      F,FALSE      GCHAR
11 2405 001000000103      LDIM      O,"C
12 2406 031521000240      RA        SUBO      IMOP,WRKO
13 2407 071000422413      JMP       T,NZ      TCP1
14 2410 001000000001      LDIM      O,1
15 2411 033730000012      RZB      OR      IMOP,WRKO
16 2412 071000002414      JMP       F,FALSE      ,+6
17 2413 071002002426      CAL      F,FALSE      FORMAT
18 2414 233430000255      ZAB      OR      WRKO,WRK3      PUSH
19 2415 071002002455      TCP2:  CAL      F,FALSE      GCHAR
20 2416 001000000040      LDIM      O,40
21 2417 031521000240      RA        SUBO      IMOP,WRKO
22 2420 071000022432      JMP       F,NZ      ,EMUL
23 2421 001000000015      LDIM      O,15
24 2422 031521000240      RA        SUBO      IMOP,WRKO
25 2423 071000022432      JMP       F,NZ      EMUL
26 2424 071002002426      CAL      F,FALSE      FORMAT
27 2425 533161400255      ABBR      31      WRKO,WRK3,40      LRTN

28
29
30 2426 001000000005      FORMAT: LDIM      O,5
31 2427 033500000252      RAB      ADD      IMOP,WRKO,WRKO
32 2430 001000000017      LDIM      O,17
33 2431 433540000252      RAB      AND      IMOP,WRKO,WRKO      RTN
34
35 2432 3010000002435      EMUL:  LDIM      O,CASEB/3
36 2433 0335000070335      RABR      ADD      IMOP,WRK3,WRK3,MIX
37 2434 071001000000      JMX      F,FALSE      POP

```

```

;WRKO:=NEXTCHAR
;IF CHAR = 'C' THEN
;WRKO:=1
;ELSE
;WRKO:=(WRKO+5)&17;
;WRK3:=WRKO
;WHILE CHAR<>'SP' AND CHAR<>'CR' D
;BEGIN
;IF CHAR ='SP' THEN
;GOTO EMUL
;IF CHAR ='CR' THEN GOTO EMUL;
;WRK3:=WRK3 EXOR WRKO;
;END;
;WRKO:=WRKO+5;
;WREKO:=WRKO AND17, RETURN.
;GOTO THE PROPER
;INTERPRETER ROUTINE.
;MIX:=CASEBASE+WRK1, JMP @MIX;

```


10074

```

;TYPE ROUTINE.
;
;THE ROUTINE TYPE THE 7 BIT CHAR IN WRK1.
;CR (<15>) IS TYPED AS CR LF .(<15><12>).
;
TYPE:      JMP      F,TPACK,,
           LDIM     0,377
           RAB      AND      IMOP,WRK1,WRK1
           ZAR      OR       WRK1,TPOUT
           LDIM     0,2
           RABR     OR       IMOP,CNTR,CNTR,CNTR0
           JMP      T,FALSE
           RABR     CAND     IMOP,CNTR,CNTR,CNTR0
           LDIM     0,15
           RA       SUBO
           JMP      F,NZ      ,.+3
           LDIM     0,12
           RZB      OR
           JMP      F,FALSE
           RTN

```

;OCTAL NUMBER INPUT ROUTINE.

```

;
;THE ROUTINE READ A OCTAL NUMBER FROM TCP TO WRK2.
;THE NUMBER IS TERMINATED BY 'CR' OR ':'.
;THE ROUTINE TEST FOR OVERFLOW.
;AUX REG: WRK0,1,2

```

```

GTNUM:    ZAB      AND      WRK2,WRK2
GTNM1:    CAL      F,FALSE      GCHAR
           LDIM     0,15
           RA       SUBO
           JMP      T,NZ      ,.+3      RTN
           LDIM     0,"
           RA       SUBO
           JMP      T,NZ      ,.+3      RTN
           LDIM     0,7
           RAB      AND      IMOP,WRK0,WRK0
           SSLA     ADD      WRK2,WRK2,Z,NL
           SSLB     OR       WRK2,Z,NL
           ABB      OR       WRK0,WRK2
           JMP      F,FALSE      GTNM1

```

```

;IF -,TPOUTACK THEN WAIT;
;MASK OUT BIT (16:23)
;TPOUT REG:=CHAR
;SET TPOUT<READY
;NO<OP.
;CLEAR TPOUT<RDY.
;TEST FOR CR
;IF NON<ZERO THEN RETURN
;WRK1:=LF.
;GOTO TYPE;

```

```

;WRK2:=0
;WRK0:=NEXTCHAR
;TEST FOR 'CR'
;IF CR THEN RETURN;
;TEST FOR ':'
;IF ':' THEN RETURN;
;WRK0:=WRK0&7;
;WRK2:=WRK2 SHIFT 2;
;WRK2:=WRK2 SHIFT 1;
;WRK2:=WRK2 OR WRK0
;REPEAT.

```

2473 071000322473
2474 001000000377
2475 033540000273
2476 031430050260
2477 001000000002
2500 033530040231
2501 071000402501
2502 033550040231
2503 001000000015
2504 031521000260
2505 471000022506
2506 001000000012
2507 033730000013
2510 0710000002473

2511 033440000314
2512 071002002455
2513 001000000015
2514 031521000240
2515 471000422516
2516 001000000072
2517 031521000240
2520 471000422521
2521 001000000007
2522 033540000252
2523 047100000314
2524 047330000014
2525 033130000254
2526 0710000002512

```

10075
01 2527 0010336000015          TCPR:  LDIM          "16.,15      CONT
02 2530 071002002535          CAL          F,FALSE      TPTXT
03 2531 0710000000051          JMP          F,F          IDLE
04
05
06
07 2532 0010000000015          CRLF:  LDIM          0,15      IMOP,WRK1
08 2533 0337300000013          RZB          OR
09 2534 0710000002473          JMP          F,FALSE      TYPE
10
11
12
13
14
15 2535 0307300000000          TPTXT:  RZQ          OR          IMOP
16 2536 0010000000003          LDIM          0,3
17 2537 0337300000012          RZB          OR          IMOP,WRK0
18 2540 0010000000010          LDIM          0,8.
19 2541 0337300000014          RZB          OR          IMOP,WRK2
20 2542 0463300000013          DSLB          OR          WRK1,Z,NL
21 2543 033410000314          ZAB          SUN          WRK2,WRK2
22 2544 071000422542          JMP          T,NZ          TPTX2
23 2545 071002002473          CAL          F,FALSE      TYPE
24 2546 033410000252          ZAB          SUN          WRK0,WRK0
25 2547 471000422540          JMP          T,NZ          TPTX1      RTN
26
27
28
29
30
31
32
33
34 2550 071002002555          TPNUM:  CAL          F,F,TPOCT
35 2551 071000002532          JMP          F,FALSE      CRLF
36

```

```

;COMMAND ERROR.
;TYPE '<15>', RETURN.
;RETURN TO IDLELOOP

```

```

;ROUTINE TO TYPE CR+LF.
;TYPE <15><12>, RETURN.

```

```

;Q:=IMOP.
;WRK0:=3, COUNTER.
;WRK2:=8., COUNTER.
;(WRK1,Q):=(WRK1,Q) SHIFT 1;
;WRK2:=WRK2-1;
;IF NON ZERO THE REPEAT;
;TYPE THE CHAR
;WRK0:=WRK0-1
;IF ZERO THEN RETURN ELSE REPEAT;

```

```

;TYPE TEXT.
;THE ROUTINE TYPE THE 3 CHARACTORS IN IMOP REG.
;AUX REG: WRK0,1,2 AND Q.

```

```

;TYPE BINARY NUMBER.

```

```

;THE ROUTINE TYPE THE BINARY NUMBER IN Q - REG AS 8
;OCTAL DIGITS, TERMINATED BY CRLF.
;AUX REG: WRK0,1 AND Q.

```

```

;TYPE THE CONTENTS OF Q AS 8 OCTAL DIGITS
;TYPE CRLF, RETURN;

```

```

10076
01 ;TYPE ADDRESS
02 ;
03 ; THE ROUTINE TYPES THE BINARY NUMBER IN THE Q-REGISTER
04 ; AS 8 OCTAL DIGITS, TERMINATED BY : SP SP
05 ; AUX REG: WRK0, WRK1, WRK2, Q, IMOP
06 ;TYPE (Q) AS 8 OCTAL DIGITS
07 2552 071002002555 TYPAD: CAL F,F,TPOCT
08 2553 001032420040 LDIM 1642,0040
09 2554 071000002535 JMP F,F,TPTXT
10 ;TYPE <:><SP><SP> ON TCP, RTN
11
12 ;TYPE OCTAL NUMBER SUBROUTINE
13 ;
14 ; THE ROUTINE TYPES THE CONTENTS
15 ; OF THE Q-REGISTER AS 8 OCTAL DIGITS
16
17 2555 001000000010 TPOCT: LDIM 0,10
18 2556 033730000012 RZB OR,IMOP,WRK0
19 2557 033440000273 TPOC1: ZAB AND,WRK1,WRK1
20 2560 046330000013 DSLB OR,WRK1,Z,NL
21 2561 046330000013 DSLB OR,WRK1,Z,NL
22 2562 046330000013 DSLB OR,WRK1,Z,NL
23 2563 001000000060 LDIM 0,60
24 2564 033500000273 RAB ADD,IMOP,WRK1,WRK1
25 2565 071002002473 CAL F,F,TYPE
26 2566 033410000252 ZAB SUN,WRK0,WRK0
27 2567 471000422557 JMP T,NZ,TPOC1 RTN
28
29 ;TYPE <SP><SP> SUBROUTINE
30 ;
31 ; AUX. REG: WRK0, WRK1, WRK2, Q, IMOP
32
33 SPSP: LDIM 1002,0
34 2570 001020020000 JMP F,F,TPTXT
35 2571 071000002535 ;IMOP:=<SP><SP>
;TYPE <SP><SP>, RETURN

```

```

!0077
01 ;EXAMINE REGISTER
02 ;
03 ; DISPLAYS THE CONTENTS OF THE ADDRESSED REGISTER AS 8
04 ; OCTAL DIGITS ON THE TCP DISPLAY
05
06 2572 071002002511 XR: CAL F,F,GTNUM ;WRK2:=ADDRESS OF REGISTER
07 2573 001100000000 LDIM 4000,0
08 2574 033550000315 RAB CAND,IMOP,WRK2,WRK3 ;WRK3:=WRK2 & 37777777
09 2575 071002002570 CAL F,F,SPSP ;TYPE <SP><SP>
10 2576 071002002712 CAL F,F,GETRG ;Q:=CONTENTS OF REGISTER
11 2577 071002002550 CAL F,F,TPNUM ;DISPLAY Q ON TCP
12 2600 071000000051 JMP F,F,IDLE ;GOTO IDLE LOOP
13
14
15
16
17
18
19 ;LOAD REGISTER
20 ; THE ADDRESSED REGISTER IS LOADED WITH THE OCTAL NUMBER
21 ; TYPED ON THE TCP
22
23 LR: CAL F,F,GTNUM ;WRK2:=ADDRESS OF REGISTER
24 2601 071002002511 ZAB OR,WRK2,WRK3 ;WRK3:=WRK2
25 2602 033430000315 CAL F,F,SPSP ;TYPE <SP><SP>
26 2603 071002002570 CAL F,F,GTNUM ;WRK2:=NEW CONTENTS OF REGISTER
27 2604 071002002511 ZAQ OR,WRK2 ;Q:=WRK2
28 2605 030430000300 CAL F,F,PUTRG ;REGISTER(WRK3):=Q
29 2606 071002002757 JMP F,F,IDLE ;GOTO IDLE LOOP
30
31 ;RUN
32 ; STARTS MICROINSTRUCTION EXECUTION FROM THE ADDRESSED
33 ; CONTROL STORE LOCATION
34
35 R: CAL F,F,GTNUM ;WRK2:=ADDRESS
36 2610 071002002511 ZAR OR,WRK2,MIX ;MIX:=ADDRESS
37 2611 031430070300 JMX F,F ;GOTO (MIX)
38 2612 071001000000

```

10078

```

01 ;EXECUTE SUBROUTINE, XS
02
03 ;
04 ; EXECUTES THE ADDRESSED SUBROUTINE
05 ; AND RETURNS TO IDLE LOOP
06
07 2613 071002002511 ;XSUBR: CAL F,F,GTNUM ;WRK2:=ADDRESS
08 2614 031430070300 ZAR OR,WRK2,MIX ;MIX:=ADDRESS
09 2615 071003000000 CAX F,F ;CALL (MIX)
10 2616 071000000051 JMP F,F,IDLE ;GOTO IDLE LOOP
11
12 ;LOAD MEMORY
13
14 ; LOADS THE ADDRESSED MEMORY LOCATION WITH THE OCTAL NUMBER
15 ; TYPED ON THE TCP
16 ; THE ADDRESS IS SAVED IN WRK5
17
18 2617 071002002511 XLM: CAL F,F,GTNUM ;WRK2:=ADDRESS
19 2620 033430000317 ZAB OR,WRK2,WRK5 ;WRK5:=ADDRESS
20 2621 071002002570 CAL F,F,SPSP ;TYPE <SP><SP>
21 2622 0710000002627 JMP F,F,LDMEM ;GOTO LDMEM
22
23 ;LOAD NEXT MEMORY
24
25 ; LOADS THE MEMORY LOCATION ADDRESSED BY WRK5+2
26 ; WITH THE OCTAL NUMBER TYPED ON THE TCP
27 ; THE ADDRESS IS SAVED IN WRK5
28
29 2623 001000000002 LN: LDM 0,2 ;WRK5:=WRK5+2
30 2624 033500000377 RAB ADD,IMOP,WRK5,WRK5 ;Q:=WRK5
31 2625 030430000360 ZAQ OR,WRK5 ;DISPLAY ADDRESS
32 2626 071002002552 CAL F,F,TYPAD ;WRK2:=DATA
33 2627 071002002511 LDMEM: CAL F,F,GTNUM ;DATA OUT:=DATA
34 2630 031430030300 ZAR OR,WRK2,DATO ;I/O ADDR:=WRK5, WRITE
35 2631 031430420360 ZAR OR,WRK5,WRT ;IF BUS ERROR THEN GOTO TCPER
36 2632 071010642527 WJMP T,BERR,TCPER ;GOTO IDLE LOOP
37 2633 0710000000051 JMP F,F,IDLE

```

```

0079
01 ;EXAMINE MEMORY
02
03 ;
04 ; DISPLAYS THE CONTENTS OF THE ADDRESSED MEMORY LOCATION
05 ; NOTE THAT ADDRESSES 0,2,4,6 REFERS TO MEMORY LOCATIONS 0,2,4,6
06 ; THE ADDRESS IS SAVED IN WRK5
07
08 XM: 2634 071002002511 F,F,GTNUM ;WRK2:=ADDRESS
09 2635 033430000317 OR,WRK2,WRK5 ;WRK5:=ADDRESS
10 2636 071002002653 F,F,XMEM ;DISPLAY ADDRESS AND CONTENTS
11 2637 071000000051 F,F,IDLE ;GOTO IDLE LOOP
12
13
14 ;EXAMINE NEXT
15
16 ;
17 ; DISPLAYS THE CONTENTS OF A NUMBER OF CONSECUTIVE MEMORY LOCATIONS
18 ; STARTING AT THE LOCATION ADDRESSED BY (WRK5)+2.
19 ; THE LAST ADDRESS IS SAVED IN WRK5.
20
21 XM: 2640 071002002511 F,F,GTNUM ;WRK2:=NO. OF LOCATIONS
22 2641 001000000002 LDIM 0,2
23 2642 033730000015 OR,IMOP,WRK3 ;WRK3:=+2
24 2643 033430000316 OR,WRK2,WRK4 ;WRK4:=WRK2
25 2644 071000020051 F,NZ,IDLE ;IF WRK2=0 THEN GOTO IDLE
26 2645 033100000337 XNEXT: ADD,WRK3,WRK5 ;WRK5:=WRK5+WRK3
27 2646 071002002653 F,F,XMEM ;DISPLAY ADDRESS AND CONTENTS
28 2647 033310000016 SUN,WRK4 ;WRK4:=WRK4-1
29 2648 071000020051 F,NZ,IDLE ;IF WRK4=0 THEN GOTO IDLE
30 2649 0710000312404 F,NTPIN,TCPIN ;IF TCP INP. RDY. THEN GOTO TCPIN
31 2650 071000002645 F,F,XNEXT ;GOTO XNEXT

```

```

!0080
01 ;DISPLAY MEMORY ADDRESS AND CONTENTS SUBROUTINE
02 ;
03 ; DISPLAYS THE ADDRESS AND THE CONTENTS OF THE MEMORY
04 ; LOCATION ADDRESSED BY WRK5
05
06 XMEM: ZAGR OR,WRK5,READ ;I/O ADDR:=Q:=WRK5, READ
07 CAL F,F,TYPAD ;DISPLAY ADDRESS
08 WJMP T,BERR,TCPER ;IF BUS ERROR THEN GOTO TCMR
09 RZQ OR,DATI ;Q:=DATA IN
10 JMP F,F,TPNUM ;DISPLAY CONTENTS OF MEM. LOCATION, RTN
11
12
13 ;SINGLE INSTRUCTION
14 ;
15 ; EXECUTES THE INSTRUCTION ADDRESSED BY PC AND RETURNS TO IDLE LOOP.
16 ; THE CONTENTS OF W0, W1, W2, W3, STAT, IC, CAUSE, AND SB
17 ; IS DISPLAYED
18
19 S: LDIM 0,20
20 PABR OR,IMOP,CNTR,CNTR,CNTR ;SINGLE INSTR:=1
21
22 ;CONTINUE
23
24 ; STARTS INSTRUCTION EXECUTION IN THE LOCATION ADDRESSED BY PC
25
26 C: WJMP F,F,MLOOP ;IF I/O READY THEN GOTO MAIN LOOP
27 2662 0710100000243

```

!0081

;SINGLE INSTRUCTION INTERRUPT

01
02
03 2663 001000000120
04 2664 033550040231
05 2665 001000002702
06 2666 033730000016
07 2667 233340000015
08 2670 031100070355
09 2671 071003000000
10 2672 071002002535
11 2673 071002002570
12 2674 071002002712
13 2675 071002002550
14 2676 033301000015
15 2677 001000000007
16 2700 031521000320
17 2701 571000010051

SINGL: LDIM
RABR
LDIM
RZB
ZBB
ABR
CAX
CAL
CAL
CAL
CAL
ZBB
LDIM
RA
JMP

0,120
CAND,IMOP,CNTR,CNTR,CNTR0
O,RGTX1/3
OR,IMOP,WRK4
AND,WRK3
ADD,WRK4,WRK3,MIX
F,F
F,F,TPXT
F,F,SPSP
F,F,GETRG
F,F,TPNUM
ADD0,WRK3
O,7
SUB0,IMOP,WRK3
F,NNNEG,IDLE

LRTN

;IF WRK3>7
;THEN GOTO IDLE ELSE DISPLAY NEXT

;RUN:=SINGLE INSTR:=0
;WRK4:=RGTX1
;WRK3:=0, PUSH
;MIX:=WRK3+WRK4
;IMOP:=REG. NAME
;TYPE <REG NAME><:>
;TYPE <SP><SP>
;Q:=CONTENTS OF REG
;TYPE CONTENTS OF REG
;WRK3:=WRK3+1

20 2702 401051630072
21 2703 401051630472
22 2704 401051631072
23 2705 401051631472
24 2706 401050652072
25 2707 401042241472
26 2710 401040651472
27 2711 401050641072

RGTX1: LDIM
LDIM
LDIM
LDIM
LDIM
LDIM
LDIM
LDIM

RIN
RIN
RIN
RIN
RIN
RIN
RIN
RIN

;IMOP:= <W0:>
;IMOP:= <W1:>
;IMOP:= <W2:>
;IMOP:= <W3:>
;IMOP:= <ST:>
;IMOP:= <IC:>
;IMOP:= <CS:>
;IMOP:= <SB:>

ADDRESS	INSTR	OPERAND	COMMENT
0000	GET	REGISTER SUBROUTINE	
0001			
0002			
0003			
0004			
0005			
0006	GETRG	LDIM 0,62	
0007		SUNO,IMOP,WRK3	
0008		F,NZ,GTRIC	
0009		LDIM 0,60	
0010		SUNO,IMOP,WRK3	
0011		F,NZ,GSP16	
0012		SUN,IMOP,WRK3	
0013		F,NZ,GSP17	
0014		LDIM 0,20	
0015		SUN,IMOP,WRK3	
0016		T,NNEG,DEFLT	
0017		O,GRTAB/3	
0018		ADD,IMOP,WRK3,MIX	
0019		F,F	
0020			
0021	GRTAB	ZAG 0,WO	
0022		ZAG 1,W1	
0023		ZAG 2,W2	
0024		ZAG 3,W3	
0025		ZAG 4,STAT	
0026		ZAG 5,IC	
0027		ZAG 6,CAUSE	
0028		ZAG 7,SB	
0029			
0030		SZQ 10,CPA	
0031		SZQ 11,BASE	
0032		SZQ 12,LLIM	
0033		SZQ 13,ULIM	
0034		SZQ 14,ILIMC	
0035		SZQ 15,INF	
0036		SZQ 16,PC	
0037		SZQ 17,SIZE	
0038		SZQ 20,MONT	
0039			

!0083						
01	2751	420730007000	GSP16:	SZQ	OR,SP16	RTN
02	2752	420730007400	GSP17:	SZQ	OR,SP17	RTN
03						
04	2753	020730004000	GTRTC:	SZQ	OR,RTC	
05	2754	001000177777		LDIM	17,7777	
06	2755	430640000000		RQQ	AND,IMOP	RTN
07						
08	2756	430120000252	DEFLT:	ABQ	SUB,WRKO,WRKO	RTN

;ADDR=60, Q:=SP16
 ;ADDR=61, Q:=SP17
 ;ADDR=62, Q:=RTC
 ;Q:=8 EXT 0 CON RTC(8:23)
 ;UNKNOWN ADDR, Q:=-1

10084

;PUT REGISTER SUBROUTINE

; THE SUBROUTINE LOADS THE REGISTER ADDRESSED BY WRK3
; WITH THE CONTENTS OF THE Q-REGISTER

```

06 2757 0010000000057 PUTRG: LDIM 0,57
07 2760 031511000320 RA SUNO,IMOP,WRK3
08 2761 071000023017 JMP F,NZ,PUT57
09 2762 031510000320 RA SUN,IMOP,WRK3
10 2763 071000023020 JMP F,NZ,PUT60
11 2764 0010000000022 LDIM 0,22
12 2765 031510000320 RA SUN,IMOP,WRK3
13 2766 471000012767 JMP F,NEG,+.3 RTN
14 2767 032730001334 SWAP WRK3,WRK2
15 2770 031430060300 ZAR OR,WRK2,IR
16
17 2771 0010000002774 LDIM 0,PRTAB/3
18 2772 031500070320 RAR ADD,IMOP,WRK3,MIX
19 2773 071001000000 JMX F,F

```

```

21 2774 4332300000002 PRTAB: ZQB OR,X RTN
22 2775 4332300000002 ZQB OR,X RTN
23 2776 4332300000002 ZQB OR,X RTN
24 2777 4332300000002 ZQB OR,X RTN
25 3000 4332300000004 ZQB OR,STAT RTN
26 3001 4332300000005 ZQB OR,IC RTN
27 3002 4332300000006 ZQB OR,CAUSE RTN
28 3003 4332300000007 ZQB OR,SB RTN

```

```

30 3004 4112200000000 SUB,CPA RTN
31 3005 4112200000400 SUB,BASE RTN
32 3006 4112200001000 SUB,LLIM RTN
33 3007 4112200001400 SUB,ULIM RTN
34 3010 4112200002000 SUB,ILIMC RTN
35 3011 4112200002400 SUB,INF RTN
36 3012 4332300000010 OR,PC RTN
37 3013 4112200003000 SUB,SIZE RTN
38 3014 4112200003400 SUB,MONT RTN
39 3015 4112200007000 SUB,SP16 RTN
40 3016 4112200007400 SUB,SP17 RTN

```

```

;WRK3-57
;IF WRK3=57 THEN GOTO PUT57
;IF WRK3=60
;THEN GOTO PUT60

;WRK3-22-1
;IF WRK3>22 THEN RETURN
;WRK2:=WRK3(12:23) CON WRK3(0:11)
;PREPARE INDIRECT ADDRESSING
;OF W(0:3) VIA X-FIELD OF IR

;MIX:=WRK3+PUT REG TABLE BASE
;GOTO (MIX)

```

```

;ADDR= 0, W0:=Q
; 1, W1:=Q
; 2, W2:=Q
; 3, W3:=Q
; 4, STAT:=Q
; 5, IC:=Q
; 6, CAUSE:=Q
; 7, SB:=Q

```

```

;ADDR=10, CPA:=Q
; 11, BASE:=Q
; 12, LLIM:=Q
; 13, ULIM:=Q
; 14, ILIMC:=Q
; 15, INF:=Q
; 16, PC:=Q
; 17, SIZE:=Q
; 20, MONT:=Q
; 21, SP16:=Q
; 22, SP17:=Q

```

ADDRESS	INSTR	OP	RTN	REMARKS
10085				
01 3017	431230100000	PUT57:	ZQR	
02			OR,INTRG	
03 3020	0710000002550	PUT60:	JMP	
04			F,F,TPNUM	
				;CLEAR INTERRUPT BIT(Q)
				;DISPLAY Q ON TCP

10086

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

;

;

;

;

CPU 810 MICRO TEST - 781204/FK

3021 001020000000

3022 071000003026

3023 001000100000

3024 071000743026

3025 001021000000

3026 033730010004

3027 071002003103

3030 001000003032

3031 071000003274

3032 071002003103

3033 071002003240

3034 001000003036

3035 071000003630

3036 071000153021

3037 071000343023

3040 071000000066

3041 001020000000

3042 033730010004

3043 071002003103

3044 001000003041

3045 071000003274

TO:

LDIM

JMP

AENTRY:

LDIM

JMP

1000,0

F,FALSE

100,0

T,SHORT

TO.1

TO.1:

LDIM

RZBR

CAL

LDIM

JMP

CAL

CAL

LDIM

JMP

JMP

JMP

1100,0

OR

F,FALSE

0,TO.2/3

F,FALSE

F,FALSE

F,F

0,TO.3/3

F,FALSE

F,FPMSK

F,SHORT

F,FALSE

IMOP,STAT,CPUST

INITA

CPUST

INITA

SIZEMEM

MEMTEST

TO

AENTRY

AUTO1

TO.3:

T1:

LDIM

RZBR

CAL

LDIM

JMP

1000,0

OR

F,FALSE

0,T1/3

F,FALSE

IMOP,STAT,CPUST

INITA

CPUST

;RUN ALL

;AUTOLOAD ENTRY

;IF SHORT=0 THEN FULL TEST ELSE JMP TO.1

;(SHORT IS A SWITCH.)

;AULOL&FULL TEST FLAG

;CLEAR INTERRUPTS, INIT REGISTERS

;IMOP:= RETURN ADDRESS.

;GOTO CPU TEST

;CLEAR INTERRUPTS, INIT REGISTERS

;CLEAR AND SIZE MEM, SIZE->W3

;IMOP:= RETURN ADDRESS.

;IF TCP ENTRY THEN REPEAT TEST;

;IF -,SHORT THEN REPEAT

;GOTO AUTOLOAD DEVICE.

;RUN CPU TEST

;SET FULL MEM TEST FLAG

;CLEAR INTERRUPTS, INIT REGISTERS

;IMOP:= RETURN ADDRESS.

```

10087
01 3046 0010200000000 T2: LDIM 1000,0 ;RUN FULL MEM TEST
02 3047 0337300100004 OR ;SET FULL TEST FLAG
03 3050 071002003103 CAL IMOP,STAT,CPUST ;CLEAR INTERRUPTS, INIT REGISTERS
04 3051 071002003240 CAL INITA ;CLEAR AND SIZE MEM, SIZE-> W3.
05 3052 001000003046 LDIM SIZEMEM ;IMOP:=RETURN ADDRESS.
06 3053 071000003630 JMP MEMTEST
07
08 3054 0337400100004 T3: RZBR IMOP,STAT,CPUST ;RUN SHORT MEM TEST, CLEAR STAT-REG.
09 3055 071002003103 CAL INITA ;CLEAR INTERRUPTS, INIT REGISTERS
10 3056 071002003240 CAL SIZEMEM ;CLEAR AND SIZE MEM, SIZE-> W3.
11 3057 001000003054 LDIM MEMTEST ;IMOP:= RETURN ADDRESS.
12 3060 071000003630 JMP
13
14 3061 071002003103 T4: CAL INITA ;CLEAR INTERRUPTS, INIT REGISTERS.
15 3062 071000002404 JMP TCPIN ;GOTO TCP INPUT ROUTINE
16
17 3063 071002003240 T5: CAL SIZEMEM ;SIZE AND CLEAR MEMORY.
18 3064 071000002404 JMP TCPIN
19
20 3065 071002002511 T.O: CAL GTNUM ;WRK2:=NUMBER
21 3066 023530016504 SABR 15,STAT,STAT,CPUST ;START TEST IN WRK2,LOOP MODE
22 3067 031430070300 ZAR WRK2,MIX
23 3070 071001000000 JMX F,FALSE ;JMP TO LOOP
24
25 3071 071002002511 TX: CAL GTNUM ;START TEST IN WRK2
26 3072 001000003075 LDIM O,TCASE/3 ;GOTO TCASE+C(WRK2)
27 3073 031500070300 RAR ADD
28 3074 071001000000 JMX IMOP,WRK2,MIX
29 3075 071000003021 JMP T0 ;T 0
30 3076 071000003041 JMP T1 ;T 1
31 3077 071000003046 JMP T2 ;T 2
32 3100 071000003054 JMP T3 ;T 3
33 3101 071000003061 JMP T4 ;T 4
34 3102 071000003063 JMP T5 ;SIZE AND CLEAR MEMORY.
35

```

10088

01 3103 0710C20000233
 02 3104 0010C00000040
 03 3105 033730040011
 04 3106 0010100000000
 05 3107 0307300000000
 06 3110 011220005000
 07 3111 001122525252
 08 3112 0307300000000
 09 3113 011220005400
 10 3114 030120000252
 11 3115 011210006000
 12 3116 0010020000000
 13 3117 0307300000000
 14 3120 011220006400
 15 3121 001000000010
 16 3122 0307300000000
 17 3123 011220007000
 18 3124 001000000002
 19 3125 0307300000000
 20 3126 011220004400
 21 3127 4334400000377

INITA:
 INIT:

CAL LDIM
 RZBR LDIM
 RZQ LDIM
 ZQS LDIM
 LDIM RZQ
 RZQ ZQS
 ABQ ZQS
 LDIM RZQ
 RZQ ZQS
 LDIM RZQ
 RZQ ZQS
 LDIM RZQ
 RZQ ZQS
 ZAB

F,F,CINTR
 0,40
 OR IMOP,CNTR,CNTR0
 400,0
 OR IMOP
 SUB 12
 5252,5252
 OR IMOP
 SUB 13
 SUB WRK0,WRK0
 SUB 14
 200,0
 OR IMOP
 SUB 15
 SUB 0,10
 OR IMOP
 SUB SP16
 0,2
 OR IMOP
 SUB C2
 AND WRK5,WRK5

;CLEAR ALL INTERRUPTS
 ;SUBROUTINE TO INITIALIZE REG
 ;DIG.OUT REG INITIALIZED, AUTOL-LAMP SET.
 ;Q:=IMOP
 ;SP12:=ERROR FLAG
 ;SP13:=DATA PATTERN,SHORT MEM TEST
 ;Q:=-1, WRK0 DUMMY.
 ;SP14:=1,TEST SYNC FLAG
 ;SP15:=LOOP FLAG
 ;SP16:=8., START PC
 ;Q:= 2
 ;C2:=2
 ;WRK5:= 0, RETURN

RTN

5 3130 023550046231
 6 3131 3710C0143133
 7 3132 5710C0703135
 8 3133 4710C0703134

LOOP:
 LOOP1:

SABR
 JMP
 JMP
 JMP

CAND
 F,IMSK
 T,INTR
 T,INTR
 14,CNTR,CNTR,CNTR0
 LOOP1 POP
 TINTR+3 LRTN
 TINTR RTN

;CLEAR TEST CYNC
 ;IF -,LOOP THEN GOTO LOOP1 ELSE POP
 ;IF INTERRUPT THEN GOTO INTERRUPT ELSE LOOP
 ;IF INTERRUPT THEN GOTO INTERRUPT ELSE SUB-R

```

10089
01 3134 023550016504
02 3135 033730102013
03 3136 001000000377
04 3137 033540000273
05 3140 001000000005
06 3141 031521000260
07 3142 071000423145
08 3143 471000543144
09 3144 571000403144
10 3145 071002003213
11 3146 071000312404
12 3147 071000330062
13 3150 033120000252
14 3151 023550015104
15
16
17
18
19
20 3152 0710000553217
21 3153 471000133154
22 3154 071002003213
23 3155 001041251122
24 3156 071002002535
25 3157 071002002532
26 3160 020730000000
27 3161 071002002555
28 3162 071002002532
29 3163 020730000400
30 3164 071002002555
31 3165 071002002532
32 3166 020730001000
33 3167 071002002555
34 3170 071002002532
35 3171 030430000200
36 3172 071002002555
37 3173 071002002532
38

;INTERRUPT ROUTINE, CLEAR LOOP FLAG
;WRK1:=INTERRUPT LEVEL, CLEAR INTERRUPT.
;WRK1:=WRK18377
;TEST FOR RTC INTERRUPT
;IF -,RTC THEN GOTO TINT1;
;IF LOOPMODE THEN CONT ELSE RTN;
;LOOP_RETURN
;SAVE WRKO,1,2 AND Q IN SP 0-3;
;IF TCPIN THEN GOTO TCP ROUTINE
;IF AUTOL THEN GOTO AUTOENTRY
;ILLEGAL INTERRUPT OCCURED, WRKO:=-1
;CLEAR ERROR FLAG
;GOTO MAIN ERROR ROUTINE.

15,STAT,STAT,CPUST
ILEV,WRK1,INTRG
IMOP,WRK1,WRK1
IMOP,WRK1
TINT1
T,NZ
T,IMSK
T,F
F,F
F,NTPIN
F,RSTRT
SUB
CAND
12,STAT,STAT,CPUST
RTN
LRTN
+3
SAVE
TCPIN
OCPAL
WRKO,WRKO
CAND

;MAIN ERROR ROUTINE
ERRAUTO
+3
SAVE
TPTXT
CRLF
TPOCT
CRLF
TPOCT
CRLF
TPOCT
CRLF
TPOCT
CRLF
TPOCT
CRLF
PC
;
;
;WRK0
;WRK1
;WRK2
;PC
;

```

[illegible]

```

0091
11 3230 001173777776 TBERR: LDIM
12 3231 033730000012 RZB
13 3232 033440000273 ZAB
14 3233 071000653152 JMP T,BTIM
15 3234 033401000273 ZAB ADDO WRK1,WRK1
16 3235 071000663152 JMP T,BNACK
17 3236 033401000273 ZAB ADDO WRK1,WRK1
18 3237 071000673152 JMP T,BPAR
19
0
1 3240 001000000010 SIZEM: LDIM
2 3241 033730000010 RZB
3 3242 223550015104 SETP1: SABR
4 3243 023530046231 SABR
5 3244 031440030000 ZAR
6 3245 033430620210 ZABR
7 3246 033440000252 ZAB
8 3247 071012643230 WCAL
9 3250 071002003130 CAL
0 3251 323711004410 SZB
1 3252 223720004412 SZB
2 3253 021500034600 SAR
3 3254 023500424610 SABR
4 3255 071000013264 JMP F,NNEG
5 3256 071010643264 WJMP T,BERR
6 3257 031430020200 ZAR
7 3260 031230000000 ZQ
8 3261 071010643264 WJMP
9 3262 031521001600 RA
0 3263 571000423264 JMP
1
2
3 3264 331120060356 ADREND: ABR
4 3265 023511004602 SAB
5 3266 033430000213 ZAB
6 3267 031440030000 ZAR
7 3270 023511624673 ADRE1: SABR
8 3271 471000203272 JMP F,NMADR
9 3272 071010643230 WJMP T,BERR
0 3273 071000003270 JMP F,F

;BUSERROR
;WRK0:=-2
;WRK1:=0
;IF BUSTIMEOUT THEN 0 ELSE
;IF BUSNACK THEN 1 ELSE
;IF BUSPARITY THEN 2;

7777,7776
OR IMOP,WRK0
AND WRK1,WRK1
T,BTIM
ADDO WRK1,WRK1
T,BNACK
ADDO WRK1,WRK1
T,BPAR
TERROR

0,8.
OR IMOP,PC
CAND 12,STAT,STAT,CPUST
OR 14,CNTR,CNTR,CNTR
AND 0,DATO
OR PC,PC,WRTP
AND WRK0,WRK0
T,BERR
F,F
SUNO,C2,PC
SUB C2,WRK0
ADD C2,PC,DATO
ADD C2,PC,PC,WRT
F,NNEG
T,BERR
OR PC,READ
OR
T,BERR
SUBO DATI,PC
T,NZ
ADREND LRTN

;SUBROUTINE TO CLEAR AND SIZE MEM, SIZE -> W
;PC:=8.
PUSH: CLEAR ERROR FLAG
;SET TEST SYNC
;DATO:= 0;
;WRITE WITH PROTECT.
;WRK0:=0, STATUS.
;WAIT, IF BUSERROR THEN GOTO TBERR.

;PC:=-2
;WRK0:= 1, STATUS
;DATAOUT:=PC+2
;WRITE ADDR IN MEM(ADDR)
;IF PC < 0 THEN GOTO ADREND;
;IF BUSERROR THEN GOTO ADREND;
;READ THE ADDRESS
;NO-OP
;IF BUSERROR THEN GOTO ADREND;
;IF WRITE ADDR<>READ ADDR THEN GOTO ADREND E

;IR:=-1, X_FIELD -> W3.
;W3:=PC-2, LAST MEM LOC.

;DATAOUT:= 0;
;WRITE A ZERO
;IF NOT MEM ADDR THEN RTN;
;IF BUSERROR THEN GOTO TBERR;
;REPEAT.

```

10092

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

CPUTEST:

03 3274 0337300000016
04 3275 0010000023420
05 3276 0337300000015
06 3277 0114200007720
07 3300 0010000000007
08 3301 071002152535

RZB
LDIM
RZB
ZAS
LDIM
CAL

OR
2,3420
OR
SUB
0,7
F,FPMSK

;START OF CPU TEST

IMOP,WRK4
IMOP,WRK3
WRK3,SP17

;WRK4:=RETURN ADDRESS

;SP17:=10000. CPU TEST PASS COUNTER.

TPTXT

;IF TCP ON THEN RING THE BELL AT TCP;

B10:

;TEST OF IMOP.

;-----

;THE IMMEDIATE OPERAND REGISTER AND THE ALU'S
;ABILITY TO SHIFT LEFT ARE TESTED.
;THE IMOP REG IS LOADED WITH A SHIFTING 'ZERO'
;WRK2 IS LOADED WITH THE EXPECTED DATA.

B20:

3302 0237300007012
3303 001000003317

SZB
LDIM

OR
0,IMBASE/3-1

SP16,WRK0
IMOP,PC

;WRK0:=10
;IMOP:=POINTER TO THE INSTR TO LOAD IMOP

B30:

3304 0337300000010
3305 0331200000314
3306 033301070010
3307 223550015104
3310 023530046231
3311 0710030000000
3312 0337300000013
3313 031521000300
3314 071002423152
3315 071002003130

RZB
ABR
ZBBR
SABR
SABR
CAX
RZB
RA
CAL
CAL

OR
SUB
ADDO
CAND
OR
F,F
OR
SURO
T,NZ
F,F

WRK2,WRK2
PC,MIX
12,STAT,STAT,CPUST
14,CNTR,CNTR,CNTR0

;PC:= DO.
;WRK2:=1, ALL ONES, START DATA PATTERN.
;MIX:=NEW ADDR, PC:= PC+1;
;SETUP, CLEAR ERROR FLAG
;SET TEST SYNC
;IMOP:=DATA PATT.
;WRK1:=IMOP

TERROR
LOOP

;IF IMOP <> WRK2 THEN ERROR;

3316 0473300000014
3317 071000003306

SSLB
JMP

OR
F,F

WRK2,Z,NL

;WRK2:= WRK2 SHIFT 1;
;REPEAT WHIT A NEW DATA PATTERN.

B30


```

;
; TEST OF THE ALU REGISTER AND SCRATCHPAD FILES.
;-----
;DATA PATTERN: A ONE IS SHIFTED THROUGH BOTH REGISTER FILES.
;THE ALU REG STACK IS READ BOTH FROM THE A- AND THE B FILES.
;NOTE: WRKO AND STAT (SEE TEST OF COND GROUP 1) IS NOT TESTED.
;
;ERROR INFORMATION:
;   WRKO:    100-234      ;SEE THE REGBASE TABLE
;   WRK1:    THE SELECTED REG'S CONTENTS
;   WRK2:    EXPECTED DATA PATTERN
;
;LDIM        0,100
RZB          OR
ZAS         SUB
;
;LDIM        0,1
RZB          OR
LDIM        0,REGRASE/3-100
RAR          ADD
SABR        12,STAT,STAT,CPUST
SABR        14,CNTR,CNTR,CNTRC
LDIM        4001,0004
RZR          OR
RZR          IMOP,IR
CAX          IMOP,WRTF
ZQB         WRK1
AB          WRK2,WRK1
CAL         REGERR
CAL         INIT
CAL         LOOP
ADD         WRK2,WRK2
T,NZ       B110
JMP         WRKO
ZBB         WRKO
ZBB         WRKO
JMP         R100
;
REGERR: CAL F,F
JMP F,F

```

REGBASE:

3800

[illegible]

03	3402	430430000300	ZAQ
04	3403	031230000000	ZQ
05	3404	033430000300	ZAB
06	3405	430430000000	ZAQ
07	3406	033430000300	ZAB
08	3407	430330000000	ZBQ
09	3410	033430000302	ZAB
10	3411	430430000020	ZAQ
11	3412	033430000302	ZAB
12	3413	430330000002	ZBQ
13	3414	033430000303	ZAB
14	3415	430430000040	ZAQ
15	3416	033430000303	ZAB
16	3417	430330000003	ZBQ
17	3420	033430000301	ZAB
18	3421	430430000060	ZAQ
19	3422	033430000301	ZAB
20	3423	430330000001	ZBQ
21	3424	033430000305	ZAB
22	3425	430430000120	ZAQ
23	3426	033430000305	ZAB
24	3427	430330000005	ZBQ
25	3430	033430000306	ZAB
26	3431	430430000140	ZAQ
27	3432	033430000306	ZAB
28	3433	430330000006	ZBQ
29	3434	033430000307	ZAB
30	3435	430430000160	ZAQ
31	3436	033430000307	ZAB
32	3437	430330000007	ZBQ
33	3440	033430000310	ZAB
34	3441	430430000200	ZAQ
35	3442	033430000310	ZAB
36	3443	430330000010	ZBQ
37	3444	033430000311	ZAB
38	3445	430430000220	ZAQ

[illegible]

```

WRK2
WRK2,W
W0
WRK2,W
W0
WRK2,X
W1
WRK2,X
X
WRK2,GRX
W2
WRK2,GRX
GRX
WRK2,WPRE
W3
WRK2,WPRE
WPRE
WRK2,IC
IC
WRK2,IC
IC
WRK2,CAUSE
CAUSE
WRK2,CAUSE
CAUSE
WRK2,SB
SB
WRK2,SB
SB
WRK2,PC
PC
WRK2,PC
PC
WRK2,CNTR
CNTR

```

RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN RTN

```

;100
;(NOP)
;102
;104
;106
;110
;112
;114
;116
;120
;122
;124
;126
;130
;132
;134
;136
;140
;142

```

10096	01	3446	033430000311	ZAB	OR	WRK2,CNTR	RTN	;144
	02	3447	430330000011	ZBQ	OR	CNTR		
	03	3450	033430000313	ZAB	OR	WRK2,WRK1		;146
	04	3451	430430000260	ZAQ	OR	WRK1		
	05	3452	033430000313	ZAB	OR	WRK2,WRK1		;150
	06	3453	430330000013	ZBQ	OR	WRK1		
	07	3454	033430000314	ZAB	OR	WRK2,14		;152
	08	3455	430430000300	ZAQ	OR	14		
	09	3456	033430000314	ZAB	OR	WRK2,14		;154
	10	3457	430330000014	ZBQ	OR	14		
	11	3460	033430000315	ZAB	OR	WRK2,15		;156
	12	3461	430430000320	ZAQ	OR	15		
	13	3462	033430000315	ZAB	OR	WRK2,15		;160
	14	3463	430330000015	ZBQ	OR	15		
	15	3464	033430000316	ZAB	OR	WRK2,16		;162
	16	3465	430430000340	ZAQ	OR	16		
	17	3466	033430000316	ZAB	OR	WRK2,16		;164
	18	3467	430330000016	ZBQ	OR	16		
	19	3470	033430000317	ZAB	OR	WRK2,17		;164
	20	3471	430430000360	ZAQ	OR	17		
	21	3472	033430000317	ZAB	OR	WRK2,17		;170
	22	3473	430330000017	ZBQ	OR	17		
	23	3474	323730003416	SZB	OR	MONT,WRK4		
	24	3475	071000003375	JMP	F,F	H120		
	25	3476	323730007405	SZB	OR	SP17,IC		;RESTORE RETURN ADDRESS IN WRK4
	26	3477	071000003375	JMP	F,F	B120		;SELECT A NEW REG.
	27	3500	011420000300	ZAS	SUB			;SAVE RETURN ADDR IN IC
	28	3501	420730000000	SZQ	OR	WRK2,0		;RETURN
	29	3502	011420000700	ZAS	SUB	0		;176
	30	3503	420730000400	SZQ	OR	WRK2,1		
	31	3504	011420001300	ZAS	SUB	1		;200
	32	3505	420730001000	SZQ	OR	WRK2,2		;202
	33	3506	011420001700	ZAS	SUB	2		
	34	3507	420730001400	SZQ	OR	WRK2,3		;204
	35	3510	011420002300	ZAS	SUB	3		
	36	3511	420730002000	SZQ	OR	WRK2,4		;206
	37	3512	011420002700	ZAS	SUB	4		
	38	3513	420730002400	SZQ	OR	WRK2,5		;210
	39	3514	011420003300	ZAS	SUB	5		
	40	3515	420730003000	SZQ	OR	WRK2,6		;212
	41	3516	011420003700	ZAS	SUB	6		
	42	3517	420730003400	SZQ	OR	WRK2,7		;214
					OR	7		

ADDRESS	INIT	POP	RETURN	ADDRESS
01 3520 011420004300	WRK2,10			:216
02 3521 420730004000	10	RTN		
03 3522 011420004700	WRK2,11			:220
04 3523 420730004400	11	RTN		
05 3524 011420005300	WRK2,12			:222
06 3525 420730005000	12	RTN		
07 3526 011420005700	WRK2,13			:224
08 3527 420730005400	13	RTN		
09 3530 011420006300	WRK2,14			:226
10 3531 420730006000	14	RTN		
11 3532 011420006700	WRK2,15			:230
12 3533 420730006400	15	RTN		
13 3534 011420007300	WRK2,16			:232
14 3535 420730007000	16	RTN		
15 3536 011420007700	WRK2,17			:234
16 3537 420730007400	17	RTN		
17 3540 311420007520	IC,SP17	POP		
18 3541 0710C2003104				
19				

0098

```

01 ;TEST OF EXTERNT INTERRUPT.
02 ;-----
03 ;THE TEST CHECK THAT ILEV 0-7 NOT SET INTERRUPT.
04 ;IT IS VERIFIED THAT ILEVEL 8-15 SET THE PROPER INTERRUPT
05 ;LEVEL.THE TEST ALSO CHECK THAT THE PROPER ILEVEL IS ABLE TO
06 ;CLEAR THE INTERRUPT.
07 ;
08 ;
09 3542 001100000000 LDIM 4000,0000
10 0 3543 033730000010 B200: RZB OR IMOP,PC
11
12
13
14
15
16
17
18
19
20
21
22
23
24 3544 0331200000314 ARB WRK2,WRK2
25 3545 033401030314 ZABR WRK2,WRK2,DATO
26 3546 223550015104 SABR 12,STAT,STAT,CPUST PUSH;SETUP, CLEAR ERROR FLAG.
27 3547 023530046231 SABR 14,CNTR,CNTR,CNTRO
28 3550 031430420200 ZAR PC,WRT
29 3551 0010000000301 LDIM 0,301
30 3552 0337300000012 RZB IMOP,WRK0
31 3553 071012663230 WCAL TBERR
32 3554 071002703623 CAL CHINT
33 3555 0710C2003130 CAL LOOP
34
35 3556 3215200007300 SA SP16,WRK2 POP
36 3557 0710C0423545 JMP T,NZ B250
37

```

```

;IMOP:=CPU ADDRESS. (SWITCHES IN POS XXX ALL
;
;THE LOOP CHECK THAT EXTERNT INTERRUPT
;LEVEL 0-7 NOT IS ABLE TO SET THE
;INTERRUPT FLAG.
;WRK2:=-1, ILEVEL COUNTER
;DATO:=0,1,...7,WRK2:=WRK2+1;
;SETUP;SETUP, CLEAR ERROR FLAG.
;SET TEST SYNC
;SIMULATE EXTERNT INTERRUPT
;
;WRK0:= 301, STATUS
;IF BNACK THEN GOTO TBERR
;IF ILLEGALT INTERRUPT THEN GOTO ERROR;
;RESULT:=8.-WRK2-1
;IF WRK2 <> 7 THEN GOTO B250;

```

0099

B260:

```
01 3560 033401030314
02 3561 223550015104
03 3562 023530046231
04 3563 031430420200
05 3564 001000000310
06 3565 033730000012
07 3566 071012653152
08 3567 071002643230
09 3570 033301000012
10 3571 001000000005
11 3572 031730100000
12 3573 030230000000
13 3574 071002303152
14 3575 033301000012
15 3576 001000000005
16 3577 031730100000
17 3600 001000000377
18 3601 033730002013
19 3602 033540000273
20 3603 031121000274
21 3604 071002423152
22 3605 033301000012
23 3606 031430100300
24 3607 030230000000
25 3610 071002703623
26 3611 071002003130
27 3612 001000000017
28 3613 031521000300
29 3614 071000423560
30 3615 023720007414
31 3616 071000423621
32 3617 031430070340
33 3620 071001000000
34 3621 011420007700
35 3622 071000003302

ZABR ADDO
SABR CAND
SABR OR
ZAR OR
LDIM 0,310
RZB OR
WCAL T,BTIM
CAL T,BERR
ZBB ADDO
LDIM 0,5
RZR OR
ZQQ OR
CAL F,INTR
ZBB ADDO
LDIM 0,5
RZR OR
LDIM 0,377
RZB OR
RAB AND
AB SUBO
CAL T,NZ
ZBB ADDO
ZAR OR
ZQQ OR
CAL T,INTR
CAL F,F

LDIM 0,15.
RA SUBO
JMP T,NZ

;PASS ADMINI.
;-----
SZB SUB
JMP T,NZ
ZAR OR
JMX F,F
ZAS SUB
JMP F,F

B400:
SP17,WRK2
WRK4,MIX
WRK2,SP17
B10

;TEST THAT ILEVEL 8-15 CAUSE INTERRUPT.
;THE ILEV-REG IS ALSO TESTED, AND IT IS CHECKED
;THE PROPER LEVEL CLEAR THE INTERRUPT;
;DATE:=8,9,...15
;PUSH;SETUP, CLEAR ERROR FLAG;
;SET TEST SYNC
;SIMULATE EXTERNT INTEKRUPT, LEVEL IN WRK2
;WRKO:= 310, STATUS.
;WAIT, IF BUSTIMEOUT THEN ERROR;
;IF HUSERKOR THE GOTO BUSERROR;
;WRKO:= 311
;CLEAR RTC INTERRUPT.
;NO-OPERATION
;IF --,INTERRUPT THEN ERROR;
;WRKO:=312, STATUS.
;CLEAR RTC INTERRUPT.
;NO-OPERATION/LDIM:= 377
;WRK1:=INTERRUPT LEVEL
;WRK1:= WRK1 EXTRACT 8
;IF OCCURED<>EXPECTED INTERRUPT LEVEL THEN ERROR
;WRKO:= 313, STATUS
;CLEAR INTERRUPT
;NO-OPERATION.
;IF ILLEGALT INTERRUPT THEN ERROR;

;IF WRK2 <> 15. THEN GOTO B260;

;IF PASS COUNTER = 0 THEN TERMINATE CPU TEST
;MIX:=RETURN ADDRESS.
;RETURN
;SP17:=SP17-1;
;START A NEW PASS
```

```

10100
01
02
03
04 3623 0337300002013
05 3624 001000000377
06 3625 0335400000273
07 3626 0215200007260
08 3627 4710000013152
09
10

;
; SUBROUTINER
;
CHINT: RZB OR
LDIM 0,377
RAB AND
SA SUB
JMP F,NNEG

ILEV,WRK1
IMOP,WRK1,WRK1
SP16,WRK1
TERROR RTN
;

;IF ILEVEL < 8. THEN RTN ELSE
;GOTO ERROR;
;

```

```

010101
01
02
03 3630 031730070000 A01: RZR LDIM
04 3631 001000000007 LDIM
05 3632 071002152535 CAL
06 3633 023730004412 SZB
07 3634 020730004400 SZQ
08
09 3635 001133777777 A02: LDIM
10 3636 033540010104 RABR
11 3637 023730007010 SZB
12
13 3640 00100000000060 A03: LDIM
14 3641 0337300000006 RZB
15 3642 033430000367 ZAB
16 3643 071000523646 JMP
17 3644 023730004406 SZB
18 3645 023710005417 SZR
19 3646 071002523752 A04: CAL
20 3647 071002123757 CAL

```

10102

; MEMORY LOOP.

;REGISTERS:

```
;
;WRK0 - STATUS
;WRK1 - GOOD DATA (OLD OR NEW_DATA)
;WRK2 - BAD DATA (DATI)
;WRK3 - NEWDATA
;WRK4 - AUX.
;WRK5 - "OLDOLDATA"
;SE - OLDDATA
;W3 - LAST MEM LOC.
;CAUSE- DATA PATTERN POINTER.
;PC - MEM ADDRESS
;Q - CURRENT INCREMENT.
```

; SETUP FOR MEM LOOP:

```
19 3650 223550015104
20 3651 023530046231
21 3652 011420003600
22 3653 033430220210
23 3654 071000603716
24 3655 031121000070
25 3656 071000013716
26 3657 031430030320
27 3660 071012643230
28 3661 033430620210
29 3662 033730001414
30 3663 031521001560
31 3664 071002423766
32 3665 033000000210
33 3666 071012643230
34 3667 031021220200
35 3670 071002003130
36
```

```
A10:  SABL  CAND  12,STAT,STAT,CPUST  PUSH;CLEAR ERROR FLAG, SETUP
      SABL  OR   14,CNTR,CNTR,CNTR0  ;SET TEST SYNC
      ZAS  SUB  PC,MONT  ;SAVE CURRENT PC START
      ZABR OR   PC,PC,READP  ;READ PROTECT, DATA:=OLDDATA.
      JMP  T,NMADR  FINIS  ;IF PC<8. THEN GOTO FINIS.
      AP   SUBO  W3,PC      ;IF PC > MEMSIZE THEN GOTO FINIS;
      JMP  F,NNEG  FINIS
      ZAR  OR   WRK3,DATO  ;DATAOUT REG:=NEWDATA
      WCAL T,BERR  ;WAIT IF BUSERROR THEN GOTO TBERR;
      ZABR OR   PC,PC,WRTP  ;WRITE WITH PROTECT,NEWDATA
      RZB  OR   DATI,WRK2  ;WRK2:=READ_OLDDATA
      RA   SUBO  DATI,SB   ;RES:=MEM(PC)-OLDDATA
      CAL  T,NZ  ODERR  ;IF RES<> 0 THEN JSR OLD_DATA_ERROR
      AQB  ADD  PC,PC      ;PC:=PC+Q.
      WCAL T,BERR  ;WAIT, IF BUSERROR THEN GOTO TBERR;
      AGR  SUBO  PC,READP  ;ADDR:=PC-Q, READ PROTECT.
      CAL  F,F  LOOP
```

```
PUSH;CLEAR ERROR FLAG, SETUP
;SET TEST SYNC
;SAVE CURRENT PC START
;READ PROTECT, DATA:=OLDDATA.
;IF PC<8. THEN GOTO FINIS.
;IF PC > MEMSIZE THEN GOTO FINIS;
```

```
;DATAOUT REG:=NEWDATA
;WAIT IF BUSERROR THEN GOTO TBERR;
;WRITE WITH PROTECT,NEWDATA
;WRK2:=READ_OLDDATA
;RES:=MEM(PC)-OLDDATA
;IF RES<> 0 THEN JSR OLD_DATA_ERROR
;PC:=PC+Q.
;WAIT, IF BUSERROR THEN GOTO TBERR;
;ADDR:=PC-Q, READ PROTECT.
```

```

10103
01
02
03 3671 223550015104
04 3672 023530046231
05 3673 071012643230
06 3674 033730001414
07 3675 033430220210
08 3676 031121000315
09 3677 0710C2423763
10 3700 0710C0603716
11 3701 031121000070
12 3702 0710C0013716
13 3703 071012643230
14 3704 033430620210
15 3705 033730001414
16 3706 031121000307
17 3707 0710C2423766
18 3710 071012643230
19 3711 033430220210
20 3712 023730004412
21 3713 0710C2003130
22
23 3714 3330C00000210
24 3715 0710C00003671
25

; MAIN MEMORY LOOP
A20:
SABR
SABR
WCAL
RZB
ZABR
AB
CAL
JMP
AB
JMP
WCAL
ZABR
RZB
AB
CAL
WCAL
ZABR
SZB
CAL
AQB
JMP
ADD
F,FALSE
POP
A20
PC,PC
PC:=PC+Q, POP;
;REPEAT.

12,STAT,STAT,CPUST PUSH;CLEAR ERROR FLAG, SETUP
14,CNTR,CNTR,CNTRP ;SET TEST SYNC
;WAIT, IF BUS ERROR THEN GOTO TBERR;
;WRK2:=READ_NEWDATA
;READ OLD DATA;
;RES:=MEM(PC-Q)-NEWDATA
;IF RES <> 0 THEN JSR NDERR;
;IF PC<8. THEN GOTO FINIS,
;RES:=SIZE-PC
;IF RES < 0 THEN GOTO FINIS;
;WAIT, IF BUSERROR THEN GOTO TBERR;
;WRITE WITH PROTECT, NEWDATA;
;WRK2:=MEM(PC),OLDDATA
;RES:=READ_DATA-OLDDATA;
;IF RES <> 0 THEN JSR ODERR;
;WAIT,IF BUSEKR THEN GOTO TBERR;
;READ NEWDATA, WITH PROTECT.
;SET STATUS, 2.

```

10104	FINIS:	WJMP	T,F	OR	T,EMODE	DOWN	POP
01 3716 071010403716							
02 3717 023730003416		SZB	OR				
03 3720 071000513726		JMP					
04 3721 023500004750		SAB	ADD				
05 3722 023511007216		SAB	SUNO				
06 3723 031021000340		AG	SUBO				
07 3724 071000413732	FINI1:	JMP	T,NNEG				
08 3725 071000003650		JMP	F,F				
09							
10 3726 0235110004750	DOWN:	SAB	SUNO				
11 3727 011420003460		ZAS	SUB				
12 3730 023521003616		SAB	SUBO				
13 3731 071000003723		JMP	F,F				
14 3732 333430000327	FINI2:	ZAB	OR				
15 3733 033430000070		ZAB	OR				
16 3734 071000513736		JMP	T,EMODE				
17 3735 023730007010		SZB	OR				
18 3736 033410000146	A25:	ZAB	SUN				
19 3737 071000423646		JMP	T,NZ				
20 3740 001040000000		LDIM	2000,0				
21 3741 033560010104		RABR	EXOR				
22 3742 030221000000		ZQQ	SUBO				
23 3743 071000113746		JMP	F,EMODE				
24 3744 033430000070		ZAB	OR				
25 3745 071000003640		JMP	F,FALSE				
26							
27 3746 0463300000007	A30:	DSL B	OR				
28 3747 031021000060		AG	SUBO				
29 3750 071000413635		JMP	T,NNEG				
30 3751 071001000000		JMX	F,FALSE				
31							

!0105

EXTDATA: ;

01 ;
02 ;
03 SSLB
04 LDIM
05 RA
06 JMP
07 ZAB
08
09
10
11 3752 0473330000015
12 3753 0010000000030
13 3754 031521000140
14 3755 471000013756
15 3756 433401000335

GENERATE NEWDATA, FULL MEM TEST.

WRK3:=NEW-DATA.
OR WRK3,Z,NL
O,24.
SUBO IMOP,CAUSE
F,NNEG ,.+3
ADDO WRK3,WRK3
RTN
RTN

;WRK3:=WRK3 SHIFT 1;

;RES:=24.-CAUSE

;IF POS THEN JMP ,+1 ELSE RETN(CAUSE<=24.);

;WRK3:=WRK3+1, RETURN.

SDATA: ;

01 ;
02 ;
03 SZB
04 ZA
05 JMP
06 ZAB
07
08
09
10
11 3757 0237300005415
12 3760 031410000140
13 3761 471000023762
14 3762 433420000335

GENERATE NEWDATA, SHORT MEM TEST,

WRK3:=ALT. PATTERN.

OR 13,WRK3

SUN CAUSE

F,NZ

SUB

;WRK3:=PATTERN.

;RES:=CAUSE-1; CAUSE = 2 ! 1;

;IF CAUSE <> 1 THEN RTN;

;COMPLEMENT PATTERN, RETURN;

RTN

RTN

NDERR: ;

01 AQB
02 ZAB
03 JMP
04 ZAB
05
06
07
08
09
10
11 3763 033021000210
12 3764 033430000333
13 3765 071000003152
14 3766 033430000173
15 3767 071000003152

SUBO PC,PC

OR WRK3,WRK1

F,F

OR

F,F

;PC:=PC-Q

TERROR

SB,WRK1

TERROR

!0106
01
02

000010 .DO 2048.-(./3)
-END

*** END OF LISTING ***

```

0001 .MAIN DOMUS MACRO ASSEMBLER REV 01.06
01 000000 .LOC 0
02 ;ADDRESS TABLE
03 ;*****
04
05
06 ; ADDRESS CALCULATION
07 ; INPUT:ESCAPE MODE, F-FIELD(0,1), AFTER AM, REL, X-FIELD(0,1)
08
09 00000 006100 (DIR/3)B11
10 00001 006160 (INDEX/3)B11
11 00002 006160 (INDEX/3)B11
12 00003 006160 (INDEX/3)B11
13 00004 006120 (REL/3)B11
14 00005 006220 (RELX/3)B11
15 00006 006220 (RELX/3)B11
16 00007 006220 (RELX/3)B11
17
18 00010 006300 (AMD/3)B11
19 00011 006400 (AMX/3)B11
20 00012 006400 (AMX/3)B11
21 00013 006400 (AMX/3)B11
22 00014 006260 (AMR/3)B11
23 00015 006360 (AMRX/3)B11
24 00016 006360 (AMRX/3)B11
25 00017 006360 (AMRX/3)B11
26
27 00020 006100 (DIR/3)B11
28 00021 006160 (INDEX/3)B11
29 00022 006160 (INDEX/3)B11
30 00023 006160 (INDEX/3)B11
31 00024 006120 (REL/3)B11
32 00025 006220 (RELX/3)B11
33 00026 006220 (RELX/3)B11
34 00027 006220 (RELX/3)B11
35
36 00030 006300 (AMD/3)B11
37 00031 006400 (AMX/3)B11
38 00032 006400 (AMX/3)B11
39 00033 006400 (AMX/3)B11
40 00034 006260 (AMR/3)B11
41 00035 006360 (AMRX/3)B11
42 00036 006360 (AMRX/3)B11
43 00037 006360 (AMRX/3)B11

```

10002 -MAIN
01 00040 006100 (DIR/3)B11
02 00041 006160 (INDEX/3)B11
03 00042 006160 (INDEX/3)B11
04 00043 006160 (INDEX/3)B11
05 00044 006120 (REL/3)B11
06 00045 006220 (RELX/3)B11
07 00046 006220 (RELX/3)B11
08 00047 006220 (RELX/3)B11
09
10 00050 006300 (AMD/3)B11
11 00051 006400 (AMX/3)B11
12 00052 006400 (AMX/3)B11
13 00053 006400 (AMX/3)B11
14 00054 006260 (AMR/3)B11
15 00055 006360 (AMRX/3)B11
16 00056 006360 (AMRX/3)B11
17 00057 006360 (AMRX/3)B11
18
19 00060 006100 (DIR/3)B11
20 00061 006160 (INDEX/3)B11
21 00062 006160 (INDEX/3)B11
22 00063 006160 (INDEX/3)B11
23 00064 006120 (REL/3)B11
24 00065 006220 (RELX/3)B11
25 00066 006220 (RELX/3)B11
26 00067 006220 (RELX/3)B11
27
28 00070 006300 (AMD/3)B11
29 00071 006400 (AMX/3)B11
30 00072 006400 (AMX/3)B11
31 00073 006400 (AMX/3)B11
32 00074 006260 (AMR/3)B11
33 00075 006360 (AMRX/3)B11
34 00076 006360 (AMRX/3)B11
35 00077 006360 (AMRX/3)B11

10003 MAIN
01 00100 006100 (DIR/3)B11
02 00101 006160 (INDEX/3)B11
03 00102 006160 (INDEX/3)B11
04 00103 006160 (INDEX/3)B11
05 00104 006120 (REL/3)B11
06 00105 006220 (RELX/3)B11
07 00106 006220 (RELX/3)B11
08 00107 006220 (RELX/3)B11
09
10 00110 006300 (AMD/3)B11
11 00111 006400 (AMX/3)B11
12 00112 006400 (AMX/3)B11
13 00113 006400 (AMX/3)B11
14 00114 006260 (AMR/3)B11
15 00115 006360 (AMRX/3)B11
16 00116 006360 (AMRX/3)B11
17 00117 006360 (AMRX/3)B11
18
19 00120 006100 (DIR/3)B11
20 00121 006160 (INDEX/3)B11
21 00122 006160 (INDEX/3)B11
22 00123 006160 (INDEX/3)B11
23 00124 006120 (REL/3)B11
24 00125 006220 (RELX/3)B11
25 00126 006220 (RELX/3)B11
26 00127 006220 (RELX/3)B11
27
28 00130 006300 (AMD/3)B11
29 00131 006400 (AMX/3)B11
30 00132 006400 (AMX/3)B11
31 00133 006400 (AMX/3)B11
32 00134 006260 (AMR/3)B11
33 00135 006360 (AMRX/3)B11
34 00136 006360 (AMRX/3)B11
35 00137 006360 (AMRX/3)B11

!0004 .MAIN
01 00140 006100 (DIR/3)B11
02 00141 006160 (INDEX/3)B11
03 00142 006160 (INDEX/3)B11
04 00143 006160 (INDEX/3)B11
05 00144 006120 (REL/3)B11
06 00145 006220 (RELX/3)B11
07 00146 006220 (RELX/3)B11
08 00147 006220 (RELX/3)B11
09
10 00150 006300 (AMD/3)B11
11 00151 006400 (AMX/3)B11
12 00152 006400 (AMX/3)B11
13 00153 006400 (AMX/3)B11
14 00154 006260 (AMR/3)B11
15 00155 006360 (AMRX/3)B11
16 00156 006360 (AMRX/3)B11
17 00157 006360 (AMRX/3)B11
18
19 00160 006100 (DIR/3)B11
20 00161 006160 (INDEX/3)B11
21 00162 006160 (INDEX/3)B11
22 00163 006160 (INDEX/3)B11
23 00164 006120 (REL/3)B11
24 00165 006220 (RELX/3)B11
25 00166 006220 (RELX/3)B11
26 00167 006220 (RELX/3)B11
27
28 00170 006300 (AMD/3)B11
29 00171 006400 (AMX/3)B11
30 00172 006400 (AMX/3)B11
31 00173 006400 (AMX/3)B11
32 00174 006260 (AMR/3)B11
33 00175 006360 (AMRX/3)B11
34 00176 006360 (AMRX/3)B11
35 00177 006360 (AMRX/3)B11

```

!0005 -MAIN
01 ;INSTRUCTION EXECUTIONS
02 ; INPUT:ESCAPE MODE, F FIELD(0:5)
03 ;
04 ; ESCAPE MODE=0
05
06 00200 045220 (INT0/3)B11
07 00201 027420 (00/3)B11
08 00202 013040 (BL/3)B11
09 00203 012460 (HL/3)B11
10 00204 010160 (LA/3)B11
11 00205 010260 (LO/3)B11
12 00206 010360 (LX/3)B11
13 00207 010460 (WA/3)B11
14
15 00210 010640 (WS/3)B11
16 00211 007500 (AM/3)B11
17 00212 011020 (WM/3)B11
18 00213 007600 (AL/3)B11
19 00214 041460 (RI/3)B11
20 00215 025560 (JL/3)B11
21 00216 026020 (JD/3)B11
22 00217 026540 (JE/3)B11
23
24 00220 013700 (XL/3)B11
25 00221 013440 (BS/3)B11
26 00222 013200 (BA/3)B11
27 00223 012700 (BZ/3)B11
28 00224 010060 (RL/3)B11
29 00225 025260 (SP/3)B11
30 00226 043520 (RE/3)B11
31 00227 014500 (RS/3)B11
32
33 00230 011320 (WD/3)B11
34 00231 016340 (RX/3)B11
35 00232 015560 (HS/3)B11
36 00233 016160 (XS/3)B11
37 00234 026760 (GG/3)B11
38 00235 027240 (DI/3)B11
39 00236 045220 (INT0/3)B11
40 00237 045220 (INT0/3)B11

```

10006 .MAIN
01 00240 030000 (CI/3)B11
02 00241 007660 (AC/3)B11
03 00242 023140 (NS/3)B11
04 00243 023560 (ND/3)B11
05 00244 020760 (AS/3)B11
06 00245 021500 (AD/3)B11
07 00246 022200 (LS/3)B11
08 00247 022560 (LD/3)B11
09
10 00250 024200 (SH/3)B11
11 00251 024340 (SL/3)B11
12 00252 024500 (SE/3)B11
13 00253 024600 (SN/3)B11
14 00254 024700 (SO/3)B11
15 00255 025020 (SZ/3)B11
16 00256 025120 (SX/3)B11
17 00257 027100 (GP/3)B11
18
19 00260 033500 (FA/3)B11
20 00261 033600 (FS/3)B11
21 00262 032460 (FM/3)B11
22 00263 045220 (INT0/3)B11
23 00264 035120 (FD/3)B11
24 00265 031520 (CF/3)B11
25 00266 017700 (DL/3)B11
26 00267 014600 (DS/3)B11
27
28 00270 020100 (AA/3)B11
29 00271 020360 (SS/3)B11
30 00272 045220 (INT0/3)B11
31 00273 045220 (INT0/3)B11
32 00274 045220 (INT0/3)B11
33 00275 045220 (INT0/3)B11
34 00276 045220 (INT0/3)B11
35 00277 045220 (INT0/3)B11

	!0007 .MAIN	;	ESCAPE MODE=1
01			
02			
03	00300 047360	(F00E/3)B11	.
04	00301 027360	(D0E/3)B11	
05	00302 013000	(BLE/3)B11	
06	00303 012420	(HLE/3)B11	
07	00304 010120	(LAE/3)B11	
08	00305 010220	(LOE/3)B11	
09	00306 010320	(LXE/3)B11	
10	00307 010420	(WAE/3)B11	
11			
12	00310 010600	(WSE/3)B11	
13	00311 007440	(AME/3)B11	
14	00312 010760	(WME/3)B11	
15	00313 007540	(ALE/3)B11	
16	00314 041420	(RIE/3)B11	
17	00315 025520	(JLE/3)B11	
18	00316 025760	(JDE/3)B11	
19	00317 026500	(JEE/3)B11	
20			
21	00320 013640	(XLE/3)B11	
22	00321 013400	(BSE/3)B11	
23	00322 013140	(BAE/3)B11	
24	00323 012640	(RZE/3)B11	
25	00324 010020	(RLE/3)B11	
26	00325 025220	(SPE/3)B11	
27	00326 043460	(REE/3)B11	
28	00327 014440	(RSE/3)B11	
29			
30	00330 011260	(WDE/3)B11	
31	00331 016300	(RXE/3)B11	
32	00332 015520	(HSE/3)B11	
33	00333 016120	(XSE/3)B11	
34	00334 026720	(GGE/3)B11	
35	00335 027200	(DIE/3)B11	
36	00336 047420	(F36E/3)B11	
37	00337 047460	(F37E/3)B11	

```

!0008 .MAIN
01 00340 027740 (CIE/3)B11
02 00341 007620 (ACE/3)B11
03 00342 023100 (NSE/3)B11
04 00343 023520 (NDE/3)B11
05 00344 020720 (ASE/3)B11
06 00345 021440 (ADE/3)B11
07 00346 022140 (LSE/3)B11
08 00347 022520 (LDE/3)B11
09
10 00350 024140 (SHE/3)B11
11 00351 024300 (SLE/3)B11
12 00352 024440 (SEE/3)B11
13 00353 024540 (SNE/3)B11
14 00354 024640 (SOE/3)B11
15 00355 024760 (SZE/3)B11
16 00356 025060 (SXE/3)B11
17 00357 027040 (GPE/3)B11
18
19 00360 033440 (FAE/3)B11
20 00361 033540 (FSE/3)B11
21 00362 032420 (FME/3)B11
22 00363 047520 (F63E/3)B11
23 00364 035060 (FOE/3)B11
24 00365 031460 (CFE/3)B11
25 00366 017640 (DLE/3)B11
26 00367 014540 (DSE/3)B11
27
28 00370 020040 (AAE/3)B11
29 00371 020320 (SSE/3)B11
30 00372 047560 (F72E/3)B11
31 00373 047620 (F73E/3)B11
32 00374 047660 (F74E/3)B11
33 00375 047720 (F75E/3)B11
34 00376 047760 (F76E/3)B11
35 00377 050020 (F77E/3)B11
36
37
38

```

-END

0000 SOURCE LINES IN ERROR

Connector I : ELCO Plug 8026-033-000-725
 Connector II : ELCO Plug 8026-033-000-725
 Cable : 5 x 2 x 0.25 mm²
 Length: 1.5m

CONNECTOR I		WIRE	CONNECTOR II	
SIGNAL NAME	PIN		PIN	SIGNAL NAME
AUTOLOAD NC	A2		A2	AUTOLOAD NC
AUTOLOAD C	A3		A3	AUTOLOAD C
AUTOLOAD NO	A4		A4	AUTOLOAD NO
AUTOLOAD C	A5		A5	AUTOLOAD C
POWER OK LAMP -	A6		A6	POWER OK LAMP -
POWER OK LAMP +	A7		A7	POWER OK LAMP +
RUN LAMP -	B7		B7	RUN LAMP -
RUN LAMP +	B6		B6	RUN LAMP +
AUTOLOAD LAMP -	B5		B5	AUTOLOAD LAMP -
AUTOLOAD LAMP +	B3		B3	AUTOLOAD LAMP +

Replaced by Dwg. No.

due to ECN

Replaces Dwg. No.

Design Check

Dwg. Office Check

Drawn by
770414 AOB

Designed by
770413 AAJ

A/S REGNENTRALEN

Unit

Dwg. No.

R11869

CBL 803
Connecting OCP 801 to CPU 801