

; 1 T10KI.MIC[10,1141] 11:27 11-AUG-1984
; 13 KS10.MIC[10,1141] 07:10 27-JULY-1984
; 68 REVISION HISTORY
; 187 HOW TO READ THE MICROCODE
; 392 CONDITIONAL ASSEMBLY DEFINITIONS
; 438 2901 REGISTER USAGE
; 474 MICROCODE FIELDS -- LISTING FORMAT
; 522 MICROCODE FIELDS -- DATAPATH CHIP
; 674 MICROCODE FIELDS -- RAM FILE ADDRESS AND D-BUS
; 708 MICROCODE FIELDS -- PARITY GENERATION & HALF WORD CONTROL
; 731 MICROCODE FIELDS -- SPEC
; 834 MICROCODE FIELDS -- DISPATCH
; 878 MICROCODE FIELDS -- SKIP
; 929 MICROCODE FIELDS -- TIME CONTROL
; 949 MICROCODE FIELDS -- RANDOM CONTROL BITS
; 971 MICROCODE FIELDS -- NUMBER FIELD
; 1309 DISPATCH ROM DEFINITIONS
; 1355 HOW TO READ MACROS
; 1514 MACROS -- DATA PATH CHIP -- GENERAL
; 1664 MACROS -- DATA PATH CHIP -- Q
; 1699 MACROS -- DATA PATH CHIP -- MISC.
; 1720 MACROS -- STORE IN AC
; 1752 MACROS -- MICROCODE WORK SPACE
; 1779 MACROS -- MEMORY CONTROL
; 1829 MACROS -- VMA
; 1846 MACROS -- TIME CONTROL
; 1859 MACROS -- SCAD, SC, FE LOGIC
; 1942 MACROS -- DATA PATH FIELD CONTROL
; 1958 MACROS -- SHIFT PATH CONTROL
; 1971 MACROS -- SPECIAL FUNCTIONS
; 2002 MACROS -- PC FLAGS
; 2031 MACROS -- PAGE FAIL FLAGS
; 2039 MACROS -- SINGLE SKIPS
; 2064 MACROS -- SPECIAL DISPATCH MACROS
; 2098 DISPATCH ROM MACROS
; 2139 SIMPLE.MIC[10,1141] 15:31 27-JULY-1984
; 2141 POWER UP SEQUENCE
; 2223 THE INSTRUCTION LOOP -- START NEXT INSTRUCTION
; 2347 THE INSTRUCTION LOOP -- FETCH ARGUMENTS
; 2459 THE INSTRUCTION LOOP -- STORE ANSWERS
; 2543 MOVE GROUP
; 2580 EXCH
; 2595 HALFWORD GROUP
; 2762 DMOVE, DMOVN, DMOVEM, DMOVNM
; 2793 BOOLEAN GROUP
; 2948 ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO
; 3044 ROTATES AND LOGICAL SHIFTS -- LSHC
; 3079 ROTATES AND LOGICAL SHIFTS -- ASHC
; 3118 ROTATES AND LOGICAL SHIFTS -- ROTC
; 3150 TEST GROUP
; 3302 COMPARE -- CAI, CAM
; 3371 ARITHMETIC SKIPS -- AOS, SOS, SKIP
; 3421 CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ
; 3512 AC DECODE JUMPS -- JRST, JFCL
; 3602 EXTENDED ADDRESSING INSTRUCTIONS

; 3643 XCT
; 3665 STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ
; 3762 STACK INSTRUCTIONS -- ADJSP
; 3795 SUBROUTINE CALL/RETURN -- JSR, JSP, JSA, JRA
; 3847 ILLEGAL INSTRUCTIONS AND UOO'S
; 4046 ARITHMETIC -- ADD, SUB
; 4075 ARITHMETIC -- DADD, DSUB
; 4108 ARITHMETIC -- MUL, IMUL
; 4159 ARITHMETIC -- DMUL
; 4300 ARITHMETIC -- DIV, IDIV
; 4377 ARITHMETIC -- DDIV
; 4498 ARITHMETIC -- DIVIDE SUBROUTINE
; 4563 ARITHMETIC -- DOUBLE DIVIDE SUBROUTINE
; 4603 ARITHMETIC -- SUBROUTINES FOR ARITHMETIC
; 4649 BYTE GROUP -- IBP, ILDB, LDB, IDPB, DPB
; 4726 BYTE GROUP -- INCREMENT BYTE POINTER SUBROUTINE
; 4739 BYTE GROUP -- BYTE EFFECTIVE ADDRESS EVALUATOR
; 4773 BYTE GROUP -- LOAD BYTE SUBROUTINE
; 4826 BYTE GROUP -- DEPOSIT BYTE IN MEMORY
; 4914 BYTE GROUP -- ADJUST BYTE POINTER
; 5073 BLT
; 5181 UBABL - BLT BYTES TO/FROM UNIBUS FORMAT
; 5255 FLT.MIC[10,1141] 01:46 20-MAR-1981
; 5256 FLOATING POINT -- FAD, FSB
; 5301 FLOATING POINT -- FMP
; 5330 FLOATING POINT -- FDV
; 5380 FLOATING POINT -- FLTR, FSC
; 5415 FLOATING POINT -- FIX AND FIXR
; 5452 FLOATING POINT -- SINGLE PRECISION NORMALIZE
; 5519 FLOATING POINT -- ROUND ANSWER
; 5530 FLOATING POINT -- DFAD, DFSB
; 5619 FLOATING POINT -- DFMP
; 5680 FLOATING POINT -- DFDV
; 5734 FLOATING POINT -- DOUBLE PRECISION NORMALIZE
; 5844 EXTEND.MIC[10,1141] 11:35 26-JULY-1984
; 5845 EXTEND -- DISPATCH ROM ENTRIES
; 5900 EXTEND -- INSTRUCTION SET DECODING
; 5942 EXTEND -- MOVE STRING -- SETUP
; 5987 EXTEND -- MOVE STRING -- OFFSET/TRANSLATE
; 6018 EXTEND -- MOVE STRING -- MOVSRJ
; 6066 EXTEND -- MOVE STRING -- SIMPLE MOVE LOOP
; 6090 EXTEND -- COMPARE STRING
; 6151 EXTEND -- DECIMAL TO BINARY CONVERSION
; 6283 EXTEND -- BINARY TO DECIMAL CONVERSION
; 6441 EXTEND -- EDIT -- MAIN LOOP
; 6495 EXTEND -- EDIT -- DECODE OPERATE GROUP
; 6514 EXTEND -- EDIT -- STOP EDIT
; 6529 EXTEND -- EDIT -- START SIGNIFICANCE
; 6536 EXTEND -- EDIT -- EXCHANGE MARK AND DESTINATION
; 6547 EXTEND -- EDIT -- PROCESS SOURCE BYTE
; 6610 EXTEND -- EDIT -- MESSAGE BYTE
; 6633 EXTEND -- EDIT -- SKIP
; 6647 EXTEND -- EDIT -- ADVANCE PATTERN POINTER
; 6680 EXTEND SUBROUTINES -- FILL OUT DESTINATION
; 6704 EXTEND SUBROUTINES -- GET MODIFIED SOURCE BYTE

Produced on Advanced Information Services Electronic Laser Printer, PNO1155, DTN: 223-7881

```

; 6741 EXTEND SUBROUTINES -- TRANSLATE
; 6827 EXTEND SUBROUTINES -- GET UNMODIFIED SOURCE BYTE
; 6856 EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING
; 6877 EXTEND SUBROUTINES -- UPDATE DEST STRING POINTERS
; 6921 EXTEND -- PAGE FAIL CLEANUP
; 6960          INOUT.MIC[10,1141]          09:17 27-JULY-1984
; 6961 TRAPS
; 6992 IO -- INTERNAL DEVICES
; 7098 IO -- INTERNAL DEVICES -- EBR & UBR
; 7200 IO -- INTERNAL DEVICES -- KL PAGING REGISTERS
; 7241 IO -- INTERNAL DEVICES -- TIMER CONTROL
; 7272 IO -- INTERNAL DEVICES -- WRTIME & RDTIME
; 7311 IO -- INTERNAL DEVICES -- WRINT & RDINT
; 7325 IO -- INTERNAL DEVICES -- RDPI & WRPI
; 7365 IO -- INTERNAL DEVICES -- SUBROUTINES
; 7506 PRIORITY INTERRUPTS -- DISMISS SUBROUTINE
; 7521 EXTERNAL IO INSTRUCTIONS
; 7709 SMALL SUBROUTINES
; 7733 UNDEFINED IO INSTRUCTIONS
; 7814 UMOVE AND UMOVEM
; 7869 WRITE HALT STATUS BLOCK
; 7961          PAGEF.MIC[10,1141]          12:32 26-JULY-1984
; 7963 PAGE FAIL REFIL LOGIC
;      Cross Reference Index
;      DCODE Location / Line Number Index
;      UCODE Location / Line Number Index

```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 4
; T10KI.MIC[10,1141] 11:27 11-AUG-1984 T10KI.MIC[10,1141] 11:27 11-AUG-1984

; 1 .NOBIN
; 2 ;KS10 MICROCODE PARAMETER FILE
; 3
; 4 ;PARAMETER FILE DEFINITIONS FOR T10KI
; 5 ;TOPS-10 KI PAGING MICROCODE (V7.02)
; 6
; 7 .SET/UBABLT=1 ;SUPPORT UBABLT INSTRUCTIONS
; 8 .SET/KLPAGE=0 ;REMOVE KL PAGING (REQUIRED FOR UBABLT)
; 9 .SET/NONSTD=1 ;FOR NOW, CALL THIS VERSION NON-STANDARD
; 10
; 11

; 12 .BIN
; 13

```
; 14 .NOBIN
; 15 .TITLE "KS10 MICROCODE V124, 27-JUL-84"
; 16
; 17
; 18
; 19
; 20
; 21
; 22 COPYRIGHT (C) 1976,1977,1978,1979,1980,1981,1982,
; 23 1984
; 24 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
; 25
; 26 THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE
; 27 ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED
; 28 ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT
; 29 NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF,
; 30 MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO
; 31 ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND
; 32 TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO
; 33 AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
; 34 REMAIN IN DEC.
; 35
; 36 THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
; 37 CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS
; 38 A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
; 39
; 40 DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
; 41 RELIABILITY OF ITS SOFTWARE IN EQUIPMENT WHICH IS
; 42 NOT SUPPLIED BY DEC.
; 43
; 44 DESIGNED AND WRITTEN BY:
; 45 DONALD A. LEWINE
; 46 DIGITAL EQUIPMENT CORP.
; 47 MARLBORO, MASS.
; 48 MR1-2/E47 X6430
; 49
; 50 MAINTAINED AND ENHANCED BY:
; 51 DONALD D. DOSSA
; 52 DIGITAL EQUIPMENT CORP.
; 53 MARLBORO, MASS.
; 54 MR1-2/E18 DTN 231-4138
; 55
; 56 SEAN KEENAN
; 57 DIGITAL EQUIPMENT CORP.
; 58 MARLBORO, MASS.
; 59 MR1-2/E18 DTN 231-4463
; 60
; 61 TIM LITT
; 62 PK01/C-2 223-6334
; 63
; 64
; 65
; 66
; 67
```

```
; 68 .TOC "REVISION HISTORY"
; 69
; 70 ;REV WHY
; 71 ;1 START KS10 MICROCODE BASED ON SM10 MICROCODE VERSION 510
; 72 ;2 UPDATE TO KS10 VERSION 512
; 73 ;3 FIX SOME DEFAULTS
; 74 ;4 CHANGE HARDWARE TO MATCH ECO #215
; 75 ;5 START TO UPDATE IO MICROCODE
; 76 ;6 MORE WORK ON IO
; 77 ;7 MAKE INTERRUPT THE 8080 BE A PULSE.
; 78 ;10 ADD NEW RDIO AND WRIO
; 79 ;11 FIX PROBLEMS IN MUO CODE & CORRECT T-FIELDS
; 80 ;12 FIX PROBLEMS IN DDIV
; 81 ;13 FIX UP PROBLEMS IN PI
; 82 ;14 TURN ON WRITE FOR FL-EXIT
; 83 ;15 FIX UP MAP INSTRUCTION
; 84 ;16 MORE WORK ON KI-STYLE MAP
; 85 ;17 INVERT HOLD RIGHT AND HOLD LEFT BITS
; 86 ;20 FIXUP WRIO & RDIO EFFECTIVE ADDRESS CALC.
; 87 ;21 FIX EDIT 15
; 88 ;22 HAVE LSH USE FAST SHIFT HARDWARE
; 89 ;23 FIX T-FIELD VALUES FOR PRODUCTION HARDWARE
; 90 ;24 REMOVE WRITE TEST FROM IO READS & WRITES
; 91 ;25 REWRITE MUL & MULI TO BE FASTER AND SMALLER. ALSO MAKE ADJBP
; 92 ; USE NEW MULSUB
; 93 ;26 MAKE BYTES USE FAST SHIFT ECO.
; 94 ;27 MAKE SURE VMA FETCH IS CORRECT
; 95 ;30 MORE OF 25 (FORGOT FMP)
; 96 ;31 FIX SOME PROBLEMS WITH TRAPS
; 97 ;32 SPEED UP EFFECTIVE ADDRESS CALCULATION
; 98 ;33 MORE OF 32
; 99 ;34 SPEED UP ASH & ROT
; 100 ;35 FIX UP RDTIM SO THAT TIME DOES NOT GO BACKWARDS
; 101 ;36 MORE OF 35
; 102 ;37 FIX UP PROBLEMS WITH INTERRUPTS AND DOUBLE F.P.
; 103 ;40 IMPROVE LISTING FORMAT
; 104 ;41 SPEEDUP KL-MODE PAGE REFILL
; 105 ;42 FIX UP DDIV
; 106 ;43 STILL MORE DDIV STUFF
; 107 ;44 CORRECT PROBLEMS IN D.P. PARITY STUFF
; 108 ;45 CORRECT THE BLT CLEAR-CORE CASE TO INTERRUPT CORRECTLY
; 109 ;46 MORE OF 45
; 110 ;47 DO NOT ALLOW SOFTWARE INTERRUPTS IF THE PI LEVEL IS NOT
; 111 ; ACTIVE.
; 112 ;50 MAKE FDV WORK THE SAME AS THE KL10
; 113 ;51 FIX INTERRUPT IN CVTBDX. MAKE ABORT WORK LIKE SPEC.
; 114 ;52 FIX BUG IN HALT LOOP
; 115 ;53 FIX IOEA TO WORK IF NO @ OR INDEXING
; 116 ;54 EDIT 47 BROKE JEN
; 117 ;55 FIX FLAGS IN MULTIPLY. ALSO CODE BUMS
; 118 ;56 MORE CODE BUMS
; 119 ;57 CORRECT OVERFLOW TRAPS WHICH DO MUUOS TO NOT STORE
; 120 ; THE TRAP FLAGS.
; 121
```

; 122 ;60 CORRECT TRAPS SO THAT DSKEA RUNS RIGHT
; 123 ;61 MORE OF 60. NOTE: MICROCODE REQUIRES ECO #299!!
; 124 ;62 ONE MORE TRY AT EDIT 60.
; 125 ;63 CORRECT TOPS-10 STYLE PAGING SO THAT A WRITE VIOLATION SETS
; 126 ; BIT 2 IN THE PAGE FAIL WORD (ACCESS ALLOWED).
; 127 ;64 EDIT 63 BROKE HARD PAGE FAILS. (NXM, BAD DATA, AND IO NXM)
; 128 ;65 INTERRUPTS OUT OF MOVSRJ INSTRUCTIONS DO STRANGE THINGS.
; 129 ;66 IO NXM PAGE FAIL FOR MISSING UBA GIVES PC+1 IN PAGE FAIL BLOCK.
; 130 ;67 ON A BAD DATA ERROR, STORE THE BAD WORD IN AC BLOCK 7 WORD 0 AND
; 131 ; 1
; 132 ;70 FIX A BUG WHICH CAUSED INTERRUPTS OUT OF CVTBDT TO GENERATE A BAD
; 133 ; ANSWER.
; 134 ;71 CLEANUP SOME THINGS TO MAKE LIFE EASIER FOR FIELD SERVICE
; 135 ;72 LOOK FOR 1-MS TRAP ON @ PAGE POINTERS AND ABORT REFILL IF
; 136 ; SET.
; 137 ;73 CORRECT EDIT 72.
; 138 ;74 EDIT 67 GENERATES A DATA PATH PARITY ERROR BECAUSE OF THE BAD
; 139 ; DATA. CORRECT TO NOT CHECK PARITY.
; 140 ; ALSO CHANGE POP TO TIE UP BUS LESS.
; 141 ;75 EDIT 60 BROKE TRAPS. MISSING =0 AT TRAP:.
; 142 ;76 CORRECT BUG IN DFAD AND DFSB
; 143 ;77 FIX PROBLEM SEEN IN SOME (ALL BUT ENGINEERING?) MACHINES CAUSED
; 144 ; BY EDIT 76
; 145 ;100 CHANGE DFAD/DFSB TO HAVE 2 MORE GUARD BITS. THIS SHOULD PRODUCE
; 146 ; KL10 ANSWERS FOR ALL NORMALIZED INPUTS
; 147 ; ALSO FIX A BUG IN CVTBDX PAGE FAIL LOGIC.
; 148 ;101 DFDV OF 0.0 / -0.5 HANGS THE MACHINE
; 149 ;102 FIX CHOPPED FLOATING POINT INSTRUCTIONS
; 150 ;103 CORRECT DFDV ROUNDING BUG.
; 151 ;104 CORRECT PROBLEMS IN DFMP
; 152 ;105 RDTIME SOMETIMES GIVES WRONG ANSWER. CARRY BETWEEN
; 153 ; WORDS GETS LOST SOMETIME.
; 154 ;106 MOVEM (ALSO, SETZM, SETOM, ETC.) SOMETIMES DOES NOT GENERATE
; 155 ; A WRITE-TRAP IN 100% OF THE CASES THAT IT SHOULD.
; 156 ;107 PXCT 14, DOES NOT GET THE INDEX REGISTER IN THE PREVIOUS
; 157 ; CONTEXT ALL THE TIME.
; 158 ;110 FIX TYPO IN EDIT 103
; 159 ;111 63. BIT BYTES DO NOT WORK CORRECTLY. DSKDA FAILS BECAUSE OF THIS
; 160 ; PROBLEM.
; 161 ; ***** VERSION 111 WENT OUT WITH SYSTEM REV 2 *****
; 162 ;
; 163 ;112 FIX COMMENT IN TEST INSTRUCTIONS
; 164 ;113 CORRECT IOEA TO COMPUTE CORRECT ADDRESS IF JUST LOCAL INDEXING
; 165 ; IS USED.
; 166 ;114 CORRECT INTERRUPT BUG IN DMUL
; 167 ;115 CORRECT COMMENTS HALT STATUS BLOCK
; 168 ;116 CORRECT PROBLEM WHERE CST MODIFIED BIT GETS SET BY MISTAKE.
; 169 ;117 RDINT INSTRUCTION DOES NOT WORK AT ALL. IT STORES RANDOM TRASH
; 170 ; IN THE WRONG PLACE. NEED TO LOAD BR NOT AR.
; 171 ;120 FLOATING POINT OPERATIONS SOMETIMES GET THE WRONG RESULT WITH
; 172 ; INPUTS OF UNNORMALIZED ZEROS. THIS SHOULD NEVER HAPPEN WITH
; 173 ; FORTRAN OR ANY OTHER DEC LANGUAGE.
; 174 ;121 PREVENT KEEP-ALIVE CRASHES FROM OCCURRING BECAUSE THE MOVSRJ
; 175 ; INSTRUCTION CAN LOCK OUT THE 1MS TIMER INTERRUPTS FROM BEING
; 176 ; HANDLED. THIS CAUSES THE OPERATING SYSTEM TO LOSE TRACK OF THE
; 177 ; PASSAGE OF TIME.

; T10KI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 7-1
; KS10.MIC[10,1141] 07:10 27-JULY-1984 REVISION HISTORY

; 178 ;122 DFAD FOLLOWED BY A FSC OF -5 CAUSES THE FSC TO GET WRONG
; 179 ; ANSWER. HAD TO CLEAR FLAG WORD AT EXIT OF DFAD TO FIX PROBLEM
; 180 ;123 MORE CODE FOR EDIT 121. ADDED ANOTHER DISPATCH ADDRESS FOR
; 181 ; PAGE FAIL CODE AT PFD:.
; 182 ;124 ADD ASSEMBLY OPTIONS FOR NOCST AND INHIBIT CST UPDATE.
; 183 ; ADD BLTUB/BLTBU TO GET UBA STYLE BYTES SWAPPED TO/FROM ILDB FORM.
; 184 ; ADD ASSEMBLY OPTIONS FOR KI/KL PAGE. NEED THE SPACE FOR BLTUB/BU.
; 185 ;
; 186 ;

; 187 .TOC "HOW TO READ THE MICROCODE"

; 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

1.0 FIELD DEFINITIONS

THESE OCCUR AT THE BEGINNING OF THE LISTING, IN THE SOURCE FILE KS10.MIC (CONTROL AND DISPATCH RAM DEFINITIONS). THEY HAVE THE FORM:

SYMBOL/=<L:R>M,J

THE PARAMETER (J) IS MEANINGFUL WHEN "D" IS SPECIFIED AS THE DEFAULT MECHANISM, AND IN THAT CASE, GIVES THE DEFAULT VALUE OF THE FIELD IN OCTAL. WHEN "F" IS SPECIFIED AS THE DEFAULT MECHANISM, (J) IS THE NAME OF A FIELD WHICH CONTAINS THE DEFAULT VALUE FOR THIS FIELD.

THE PARAMETER (L) GIVES THE BIT POSITION OF THE LEFTMOST BIT IN THE FIELD. THE SAME METHOD IS USED AS FOR (R) BELOW.

THE PARAMETER (R) GIVES THE FIELD POSITION IN DECIMAL AS THE BIT NUMBER OF THE RIGHTMOST BIT OF THE FIELD. BITS ARE NUMBERED FROM 0 ON THE LEFT. NOTE THAT THE POSITION OF BITS IN THE MICROWORD SHOWN IN THE LISTING BEARS NO RELATION TO THE ORDERING OF BITS IN THE HARDWARE MICROWORD, WHERE FIELDS ARE OFTEN BROKEN UP AND SCATTERED.

THE PARAMETER (M) IS OPTIONAL, AND SELECTS A DEFAULT MECHANISM FOR THE FIELD. THE LEGAL VALUES OF THIS PARAMETER ARE THE CHARACTERS "D", "F", "T", "P", OR "+".

"D" MEANS (J) IS THE DEFAULT VALUE OF THE FIELD IF NO EXPLICIT VALUE IS SPECIFIED.

"F" IS USED TO CAUSE THIS FIELD TO DEFAULT TO SOME OTHER FIELD.

"T" IS USED ON THE TIME FIELD TO SPECIFY THAT THE VALUE OF THE FIELD DEPENDS ON THE TIME PARAMETERS SELECTED FOR OTHER FIELDS. "T" IS NOT USED IN THE KS10.

"P" IS USED ON THE PARITY FIELD TO SPECIFY THAT THE VALUE OF THE FIELD SHOULD DEFAULT SUCH THAT PARITY OF THE ENTIRE WORD IS ODD. "P" IS NOT USED ON THE KS10.

"+" IS USED ON THE JUMP ADDRESS FIELD TO SPECIFY THAT THE DEFAULT JUMP ADDRESS IS THE ADDRESS OF THE NEXT INSTRUCTION ASSEMBLED (NOT, IN GENERAL, THE CURRENT LOCATION +1).

IN GENERAL, A FIELD CORRESPONDS TO THE SET OF BITS WHICH PROVIDE SELECT INPUTS FOR MIXERS OR DECODERS, OR CONTROLS FOR ALU'S.

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 9
HOW TO READ THE MICROCODE

; 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 ;

2.0 VALUE DEFINITIONS

FOLLOWING A FIELD DEFINITION, SYMBOLS MAY BE CREATED IN THAT FIELD TO CORRESPOND TO VALUES OF THE FIELD. THE FORM IS:

SYMBOL=N

"N" IS, IN OCTAL, THE VALUE OF SYMBOL WHEN USED IN THE FIELD.

3.0 LABEL DEFINITIONS

A MICRO INSTRUCTION MAY BE LABELLED BY A SYMBOL FOLLOWED BY COLON PRECEDING THE MICROINSTRUCTION DEFINITION. THE ADDRESS OF THE MICROINSTRUCTION BECOMES THE VALUE OF THE SYMBOL IN THE FIELD NAMED "J". EXAMPLE:

FOO: J/FOO

THIS IS A MICROINSTRUCTION WHOSE "J" FIELD (JUMP ADDRESS) CONTAINS THE VALUE "FOO". IT ALSO DEFINES THE SYMBOL "FOO" TO BE THE ADDRESS OF ITSELF. THEREFORE, IF EXECUTED BY THE MICROPROCESSOR, IT WOULD LOOP ON ITSELF.

4.0 COMMENTS

A SEMICOLON ANYWHERE ON A LINE CAUSES THE REST OF THE LINE TO BE IGNORED BY THE ASSEMBLER. THIS TEXT IS AN EXAMPLE OF COMMENTS.

5.0 MICROINSTRUCTION DEFINITION

A WORD OF MICROCODE IS DEFINED BY SPECIFYING A FIELD NAME, FOLLOWED BY SLASH (/), FOLLOWED BY A VALUE. THE VALUE MAY BE A SYMBOL DEFINED FOR THAT FIELD, AN OCTAL DIGIT STRING, OR A DECIMAL DIGIT STRING (DISTINGUISHED BY THE FACT THAT IT CONTAINS "8" AND/OR "9" AND/OR IS TERMINATED BY A PERIOD). SEVERAL FIELDS MAY BE SPECIFIED IN ONE MICROINSTRUCTION BY SEPARATING FIELD/VALUE SPECIFICATIONS WITH COMMAS. EXAMPLE:

AD/ZERO,RAMADR/AC*#,ACALU/AC+N,ACN/1,DBUS/DP

6.0 CONTINUATION

THE DEFINITION OF A MICROINSTRUCTION MAY CONTINUED ONTO TWO OR MORE LINES BY BREAKING IT AFTER ANY COMMA. IN OTHER WORDS, IF THE LAST NON-BLANK, NON-COMMENT CHARACTER ON A LINE IS A COMMA, THE INSTRUCTION SPECIFICATION IS CONTINUED ON

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 10
HOW TO READ THE MICROCODE

; 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 ;

THE FOLLOWING LINE. EXAMPLE:
 READ [AR], ;LOOK AT EFFECTIVE ADDRESS
 SKIP DP18, ;SEE IF RIGHT OR LEFT SHIFT
 SC_SHIFT-1, ;PUT NUMBER OF PLACE TO
 ;SHIFT IN SC
 LOAD FE, ; AND IN FE
 INST DISP ;GO DO THE SHIFT

7.0 MACROS

A MACRO IS A SYMBOL WHOSE VALUE IS ONE OR MORE FIELD/VALUE SPECIFICATIONS AND/OR MACROS. A MACRO DEFINITION IS A LINE CONTAINING THE MACRO NAME FOLLOWED BY A QUOTED STRING WHICH IS THE VALUE OF THE MACRO. EXAMPLE:

LOAD VMA "MEM/1, LDVMA/1

THE APPEARANCE OF A MACRO IN A MICROINSTRUCTION DEFINITION IS EQUIVALENT TO THE APPEARANCE OF ITS VALUE.

MACRO MAY HAVE PARAMETERS ENCLOSED IN []. FOR EXAMPLE,

[]_[] "AD/A,A/@2,DEST/AD,B/@1"

THE @1 GETS REPLACED BY WHAT IS WRITTEN IN THE FIRST SET OF [] AND @2 IS REPLACED BY WHAT IS WRITTEN IN THE SECOND SET OF []. THUS

[AR]_[ARX]

HAS THE SAME EFFECT AS SAYING

AD/A,A/ARX,DEST/AD,B/AR

SEE DESCRIPTION OF RULES FOR MACRO NAMES.

8.0 PSEUDO OPS

THE MICRO ASSEMBLER HAS 13 PSEUDO-OPERATORS:

.DCODE AND .UCODE SELECT THE RAM INTO WHICH SUBSEQUENT MICROCODE WILL BE LOADED, AND THEREFORE THE FIELD DEFINITIONS AND MACROS WHICH ARE MEANINGFUL IN SUBSEQUENT MICROCODE

.TITLE DEFINES A STRING OF TEXT TO APPEAR IN THE PAGE HEADER, AND

.TOC DEFINES AN ENTRY FOR THE TABLE OF CONTENTS AT THE BEGINNING.

.SET DEFINES THE VALUE OF A CONDITIONAL ASSEMBLY PARAMETER,

.CHANGE REDEFINES A CONDITIONAL ASSEMBLY PARAMETER,

.DEFAULT ASSIGNS A VALUE TO AN UNDEFINED PARAMETER.

.IF ENABLES ASSEMBLY IF THE VALUE OF THE PARAMETER IS NOT ZERO,

.IFNOT ENABLES ASSEMBLY IF THE PARAMETER VALUE IS ZERO, AND

.ENDIF RE-ENABLES ASSEMBLY IF SUPPRESSED BY THE PARAMETER NAMED.

.NOBIN TURNS OFF THE BINARY A GETS RID OF THE SPACE USED TO LIST IT,

.BIN TURN BINARY BACK ON AGAIN.

.WIDTH CONTROLS THE NUMBER OF BITS IN THE CRAM

; 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 ;

9.0 LOCATION CONTROL

A MICROINSTRUCTION "LABELLED" WITH A NUMBER IS ASSIGNED TO THAT ADDRESS.

THE CHARACTER "=" AT THE BEGINNING OF A LINE, FOLLOWED BY A STRING OF O'S, 1'S, AND/OR *'S, SPECIFIES A CONSTRAINT ON THE ADDRESS OF FOLLOWING MICROINSTRUCTIONS. THE NUMBER OF CHARACTERS IN THE CONSTRAINT STRING (EXCLUDING THE "=") IS THE NUMBER OF LOW-ORDER BITS CONSTRAINED IN THE ADDRESS. THE MICROASSEMBLER ATTEMPTS TO FIND AN UNUSED LOCATION WHOSE ADDRESS HAS 0 BITS IN THE POSITIONS CORRESPONDING TO O'S IN THE CONSTRAINT STRING AND 1 BITS WHERE THE CONSTRAINT HAS 1'S. ASTERISKS DENOTE "DON'T CARE" BIT POSITIONS.

IF THERE ARE ANY O'S IN THE CONSTRAINT STRING, THE CONSTRAINT IMPLIES A BLOCK OF <2**N> MICROWORDS, WHERE N IS THE NUMBER OF O'S IN THE STRING. ALL LOCATIONS IN THE BLOCK WILL HAVE 1'S IN THE ADDRESS BITS CORRESPONDING TO 1'S IN THE STRING, AND BIT POSITIONS DENOTED BY *'S WILL BE THE SAME IN ALL LOCATIONS OF THE BLOCK.

IN SUCH A CONSTRAINT BLOCK, THE DEFAULT ADDRESS PROGRESSION IS COUNTING IN THE "O" POSITIONS OF THE CONSTRAINT STRING, BUT A NEW CONSTRAINT STRING OCCURRING WITHIN A BLOCK MAY FORCE SKIPPING OVER SOME LOCATIONS OF THE BLOCK. WITHIN A BLOCK, A NEW CONSTRAINT STRING DOES NOT CHANGE THE PATTERN OF DEFAULT ADDRESS PROGRESSION, IT MERELY ADVANCES THE LOCATION COUNTER OVER THOSE LOCATIONS. THE MICROASSEMBLER WILL LATER FILL THEM IN.

A NULL CONSTRAINT STRING ("=" FOLLOWED BY ANYTHING BUT "O", "1", OR "*") SERVES TO TERMINATE A CONSTRAINT BLOCK. EXAMPLES:

=0

THIS SPECIFIES THAT THE LOW-ORDER ADDRESS BIT MUST BE ZERO-- THE MICROASSEMBLER FINDS AN EVEN-ODD PAIR OF LOCATIONS, AND PUTS THE NEXT TWO MICROINSTRUCTIONS INTO THEM.

=11

THIS SPECIFIES THAT THE TWO LOW-ORDER BITS OF THE ADDRESS MUST BOTH BE ONES. SINCE THERE ARE NO O'S IN THIS CONSTRAINT, THE ASSEMBLER FINDS ONLY ONE LOCATION MEETING THE CONSTRAINT.

=O*****

THIS SPECIFIES AN ADDRESS IN WHICH THE "40" BIT IS ZERO. DUE TO THE IMPLEMENTATION OF THIS FEATURE IN THE ASSEMBLER, THE DEFAULT ADDRESS PROGRESSION APPLIES ONLY TO THE LOW-ORDER 5 BITS, SO THIS CONSTRAINT FINDS ONE WORD IN WHICH THE "40" BIT IS ZERO, AND DOES NOT ATTEMPT TO FIND ONE IN WHICH THAT BIT IS A ONE. THIS LIMITATION HAS BEEN CHANGED WITH NEWER ASSEMBLER VERSIONS. HOWEVER NONE OF THE LOCATIONS IN THE MICROCODE REQUIRE ANYTHING BUT THE CONSTRAINT MENTIONED ABOVE.

```
; 392 .TOC "CONDITIONAL ASSEMBLY DEFINITIONS"  
; 393  
; 394 .DEFAULT/SIM=0 ;0=RUN ON REAL HARDWARE  
; 395 ;1=RUN UNDER SIMULATOR  
; 396  
; 397 .DEFAULT/FULL=1 ;0=INCLUDE ONLY BASIC INSTRUCTIONS  
; 398 ;1=INCLUDE FULL INSTRUCTION SET  
; 399  
; 400 .DEFAULT/INHCST=0 ;0=NO CODE TO INHIBIT CST UPDATE IF CSB=0  
; 401 ;1=DON'T UPDATE CST IF CSB=0  
; 402  
; 403 .DEFAULT/NOCSST=0 ;0=INCLUDE SUPPORT FOR WRITING THE CST  
; 404 ;1=COMPLETELY DESUPPORT CST (FOR TOPS-10)  
; 405  
; 406 .DEFAULT/UBABLT=0 ;0=NO UBABLT SUPPORT  
; 407 ;1=SUPPORT UBA STYLE BLT INSTRUCTIONS.  
; 408  
; 409 .DEFAULT/KIPAGE=1 ;0=DON'T SUPPORT KI PAGING  
; 410 ;1=DO  
; 411  
; 412 .DEFAULT/KLPAGE=1 ;0=DON'T SUPPORT KL PAGING  
; 413 ;1=DO  
; 414  
; 415 .IF/UBABLT ;THESE INSTRUCTIONS  
; 416 .IF/KLPAGE  
; 417 .CHANGE/KIPAGE=0  
; 418 .ENDIF/KLPAGE  
; 419 .IF/KIPAGE ;ALLOW ONLY ONE PAGING MODE  
; 420 .CHANGE/KLPAGE=0  
; 421 .ENDIF/KIPAGE  
; 422 .ENDIF/UBABLT ;OTHERWISE, MAY HAVE EITHER OR BOTH  
; 423  
; 424 .IF/NOCSST  
; 425 .CHANGE/INHCST=0  
; 426 .ENDIF/NOCSST  
; 427  
; 428 .DEFAULT/NONSTD=0 ;0=STANDARD MICROCODE  
; 429 ;1=NON-STANDARD MICROCODE  
; 430  
; 431 .WIDTH/108 ;ONLY FIELDS BETWEEN BITS 0 AND 107 EVER  
; 432 ; GET LOADED INTO THE CRAM. OTHER FIELDS  
; 433 ; ARE USED FOR DEFAULTING PROCESS.  
; 434  
; 435 .REGION/O,1377/2000,3777/1400,1777  
; 436 ;TRY AND KEEP STUFF OUT OF DROM SPACE  
; 437
```

```

; 438 .TOC "2901 REGISTER USAGE"
; 439
; 440 ;
; 441 ;0: !=====|
; 442 ;   !          MAG (ONES IN BITS 1-36, REST ZERO) !
; 443 ;1: !-----|
; 444 ;   !          PC (ADDRESS OF CURRENT INSTRUCTION + 1) !
; 445 ;2: !-----|
; 446 ;   !          HR (CURRENT INSTRUCTION) !
; 447 ;3: !-----|
; 448 ;   !          AR (TEMP -- MEM OP AT INST START) !
; 449 ;4: !-----|
; 450 ;   !          ARX (TEMP -- LOW ORDER HALF OF DOUBLE PREC) !
; 451 ;5: !-----|
; 452 ;   !          BR (TEMP) !
; 453 ;6: !-----|
; 454 ;   !          BRX (TEMP -- LOW ORDER HALF OF DOUBLE PREC BR!BRX) !
; 455 ;7: !-----|
; 456 ;   !          ONE (THE CONSTANT 1) !
; 457 ;10: !-----|
; 458 ;   !          EBR (EXEC BASE REGISTER) !
; 459 ;11: !-----|
; 460 ;   !          UBR (USER BASE REGISTER) !
; 461 ;12: !-----|
; 462 ;   !          MASK (ONES IN BITS 0-35, ZERO IN -1, -2, 36 AND 37) !
; 463 ;13: !-----|
; 464 ;   !          FLG (FLAG BITS)          !          PAGE FAIL CODE !
; 465 ;14: !-----|
; 466 ;   !          PI (PI SYSTEM STATUS REGISTER [RDPI]) !
; 467 ;15: !-----|
; 468 ;   !          XWD1 (1 IN EACH HALF WORD) !
; 469 ;16: !-----|
; 470 ;   !          TO (TEMP) !
; 471 ;17: !-----|
; 472 ;   !          T1 (TEMP) !
; 473 ;   !=====|
  
```

```
474 .TOC "MICROCODE FIELDS -- LISTING FORMAT"
475
476 ;
477 ;
478 ;
479 ; U 1561, 1500,2551,0303,0274,4463,7701,4200,0001,0001
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 ;
```

3633 1561:
3634 SUB: [AR]_AC-[AR],
3635 AD FLAGS, 3T,
3636 EXIT

! +----- # (MAGIC NUMBER)
! +----- MULTI PREC, MULTI SHIFT, CALL (4S, 2S, 1S)
! +----- FM WRITE, MEM, DIVIDE (4S, 2S, 1S)
! +----- CRY38, LOAD SC, LOAD FE (4S, 2S, 1S)
! +----- T
! +----- SKIP
! +----- DISP
! +----- SPEC
! +----- CLOCKS & PARITY (CLKR, GENR, CHKR, CLKL, GENL, CHKL)
! +----- DBM
! +----- DBUS
! +----- RAM ADDRESS
! +----- B
! +----- A
! +----- DEST
! +----- RSRC
! +----- LSRC]
! +----- ALU] - AD
! +----- J

LOCATION OF THIS MICRO WORD

```
; 522 .TOC "MICROCODE FIELDS -- DATAPATH CHIP"  
; 523  
; 524 J/= <0:11>+ ;CRA1  
; 525 ;NEXT MICROCODE ADDRESS  
; 526  
; 527 ;ALU FUNCTIONS  
; 528  
; 529 ;NOTE: THE AD FIELD IS A 2 DIGIT FIELD. THE LEFT DIGIT IS THE  
; 530 ; 2901 ALU FUNCTION. THE RIGHT DIGIT IS THE 2901 SRC CODE FOR  
; 531 ; THE LEFT HALF. NORMALLY THE RIGHT HALF SRC CODE IS THE SAME AS  
; 532 ; THE LEFT HALF.  
; 533 AD/= <12:17>D,44 ;DPE1 & DPE2  
; 534 A+Q=00  
; 535 A+B=01  
; 536 O+Q=02  
; 537 O+B=03  
; 538 O+A=04  
; 539 D+A=05  
; 540 D+Q=06  
; 541 O+D=07  
; 542 Q-A-.25=10  
; 543 B-A-.25=11  
; 544 Q-.25=12  
; 545 B-.25=13  
; 546 A-.25=14  
; 547 A-D-.25=15  
; 548 Q-D-.25=16  
; 549 -D-.25=17  
; 550 A-Q-.25=20  
; 551 A-B-.25=21  
; 552 -Q-.25=22  
; 553 -B-.25=23  
; 554 -A-.25=24  
; 555 D-A-.25=25  
; 556 D-Q-.25=26  
; 557 D-.25=27  
; 558 A.OR.Q=30  
; 559 A.OR.B=31  
; 560 Q=32  
; 561 B=33  
; 562 A=34  
; 563 D.OR.A=35  
; 564 D.OR.Q=36  
; 565 D=37  
; 566 A.AND.Q=40  
; 567 A.AND.B=41  
; 568
```


; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; KS10.MIC[10,1141] 07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 16
MICROCODE FIELDS -- DATAPATH CHIP

```
; 569 ;MORE ALU FUNCTIONS
; 570
; 571 ; ZERO=42
; 572 ; ; ZERO=43
; 573 ; ; ZERO=44
; 574 ; D.AND.A=45
; 575 ; D.AND.Q=46
; 576 ; ; ZERO=47
; 577 ; .NOT.A.AND.Q=50
; 578 ; .NOT.A.AND.B=51
; 579 ; ; Q=52
; 580 ; ; B=53
; 581 ; ; A=54
; 582 ; .NOT.D.AND.A=55
; 583 ; .NOT.D.AND.Q=56
; 584 ; ; ZERO=57
; 585 ; A.XOR.Q=60
; 586 ; A.XOR.B=61
; 587 ; ; Q=62
; 588 ; ; B=63
; 589 ; ; A=64
; 590 ; D.XOR.A=65
; 591 ; D.XOR.Q=66
; 592 ; ; D=67
; 593 ; A.EQV.Q=70
; 594 ; A.EQV.B=71
; 595 ; .NOT.Q=72
; 596 ; .NOT.B=73
; 597 ; .NOT.A=74
; 598 ; D.EQV.A=75
; 599 ; D.EQV.Q=76
; 600 ; .NOT.D=77
; 601
; 602 ;THIS FIELD IS THE RIGHTMOST 3 BITS OF THE
; 603 ; AD FIELD. IT IS USED ONLY TO DEFAULT THE RSRC
; 604 ; FIELD.
; 605 LSRC/= <15:17> ;DPE1
; 606
; 607 ;THIS IS THE SOURCE FOR THE RIGHT HALF OF THE
; 608 ; DATA PATH. IT LETS US MAKE THE RIGHT AND LEFT
; 609 ; HALF WORDS DO SLIGHTLY DIFFERENT THINGS.
; 610 RSRC/= <18:20>F,LSRC ;DPE2
; 611 ; AQ=0 ;A Q
; 612 ; AB=1 ;A B
; 613 ; OQ=2 ;O Q
; 614 ; OB=3 ;O B
; 615 ; OA=4 ;O A
; 616 ; DA=5 ;D A
; 617 ; DQ=6 ;D Q
; 618 ; DO=7 ;D O
; 619
```

```

; 620 ;DESTINATION CONTROL
; 621 ;SEE DPE1 AND DPE2 (2'S WEIGHT IS INVERTED ON DPE5)
; 622 DEST/=<21:23>D,3 ;DPE1 & DPE2
; 623     A=0 ;A REG IS CHIP OUTPUT, AD IS WRITTEN
; 624     ; INTO REG FILE
; 625     AD=1 ;REG FILE GETS AD
; 626     Q_AD=2 ;REG FILE IS NOT LOADED
; 627     PASS=3 ;AD OUTPUT IS CHIP OUTPUT
; 628     ; Q AND REG FILE LEFT ALONE
; 629     Q Q*2=4 ;ALSO REG FILE GETS AD*2
; 630     AD*2=5 ;AND Q IS LEFT ALONE
; 631     Q Q*.5=6 ;ALSO REG FILE GETS AD*.5
; 632     AD*.5=7 ;AND Q IS LEFT ALONE
; 633
; 634 ; <24:25> ;UNUSED
; 635
; 636 A/=<26:29> ;DPE1 & DPE2
; 637     MAG=0
; 638     PC=1
; 639     HR=2
; 640     AR=3
; 641     ARX=4
; 642     BR=5
; 643     BRX=6
; 644     ONE=7
; 645     EBR=10
; 646     UBR=11
; 647     MASK=12
; 648     FLG=13
; 649     PI=14
; 650     XWD1=15
; 651     TO=16
; 652     T1=17
; 653
; 654 ; <30:31> ;UNUSED
; 655
; 656 B/=<32:35>D,0 ;DPE1 & DPE2
; 657     MAG=0
; 658     PC=1
; 659     HR=2
; 660     AR=3
; 661     ARX=4
; 662     BR=5
; 663     BRX=6
; 664     ONE=7
; 665     EBR=10
; 666     UBR=11
; 667     MASK=12
; 668     FLG=13
; 669     PI=14
; 670     XWD1=15
; 671     TO=16
; 672     T1=17
; 673

```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 18
MICROCODE FIELDS -- RAM FILE ADDRESS AND D-BUS

```
; 674 .TOC "MICROCODE FIELDS -- RAM FILE ADDRESS AND D-BUS"
; 675
; 676 RAMADR/=<36:38>D,4 ;DPE6
; 677 AC#=0 ;AC NUMBER
; 678 AC*#=1 ;AC .FN. #
; 679 XR#=2 ;INDEX REGISTER
; 680 VMA=4 ;VIRTUAL MEMORY REFERENCE
; 681 RAM=6 ;VMA SUPPLIES 10-BIT RAM ADDRESS
; 682 #=7 ;ABSOLUTE RAM FILE REFERENCE
; 683
; 684 ; <39:39>
; 685
; 686 ;LEFT HALF ON DPE3 AND RIGHT HALF ON DPE4
; 687 DBUS/=<40:41>D,1 ;DPE3 & DPE4
; 688 PC FLAGS=0 ;PC FLAGS IN LEFT HALF
; 689 PI NEW=0 ;NEW PI LEVEL IN BITS 19-21
; 690 ; VMA=0 ;VMA IN BITS 27-35
; 691 DP=1 ;DATA PATH
; 692 RAM=2 ;CACHE, AC'S AND WORKSPACE
; 693 DBM=3 ;DBM MIXER
; 694
; 695 ;LEFT HALF ON DPM1 AND RIGHT HALF ON DPM2
; 696 DBM/=<42:44>D,7 ;DPM1 & DPM2
; 697 SCAD DIAG=0 ;(LH) SCAD DIAGNOSTIC
; 698 PF DISP=0 ;PAGE FAIL DISP IN BITS 18-21
; 699 APR FLAGS=0 ;APR FLAGS IN BITS 22-35
; 700 BYTES=1 ;5 COPIES OF SCAD 1-7
; 701 EXP=2 ;LH=EXPONENT, RH=TIME FRACTION
; 702 DP=3 ;DATA PATH
; 703 DP SWAP=4 ;DATA PATH SWAPPED
; 704 VMA=5 ;VMA FLAGS,,VMA
; 705 MEM=6 ;MEMORY BUFFER
; 706 #=7 ;NUMBER FIELD IN BOTH HALVES
; 707
```

```
; 708 .TOC "MICROCODE FIELDS -- PARITY GENERATION & HALF WORD CONTROL"  
; 709  
; 710 AD PARITY OK/= <108>D,0 ;**NOT STORED IN CRAM**  
; 711 ;THIS BIT IS A 1 IF THE ALU IS DOING  
; 712 ; SOMETHING WHICH DOES NOT INVALIDATE  
; 713 ; PARITY. IT DOES NOT APPEAR IN THE  
; 714 ; REAL MACHINE. WE JUST USE IT TO SET  
; 715 ; THE DEFAULT FOR GENR & GENL  
; 716  
; 717 CLKL/= <45:45>D,1 ;DPE5  
; 718 ;CLOCK THE LEFT HALF OF THE MACHINE  
; 719 GENL/= <46:46>F,AD PARITY OK ;DPE4 FROM CRM2 PARITY EN LEFT H  
; 720 ;STORE PARITY FOR 2901 LEFT  
; 721 CHKL/= <47:47> ;DPE4 FROM CRM2 PARITY CHK LEFT H  
; 722 ;CHECK LEFT HALF DBUS PARITY  
; 723  
; 724 CLKR/= <48:48>D,1 ;DPE5  
; 725 ;CLOCK THE RIGHT HALF OF THE MACHINE  
; 726 GENR/= <49:49>F,AD PARITY OK ;DPE4 FROM CRM2 PARITY EN RIGHT H  
; 727 ;STORE PARITY FOR 2901 RIGHT  
; 728 CHKR/= <50:50> ;DPE4 FROM CRM2 PARITY CHK RIGHT H  
; 729 ;CHECK RIGHT HALF DBUS PARITY  
; 730
```

; 731 .TOC "MICROCODE FIELDS -- SPEC"

; 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

THE FOLLOWING SPECIAL FUNCTION ARE DECODED ON DPE1, DPE5, AND DPMA:

```

=====
!S! EFFECT          ! CRA6 SPEC      ! CRA6 SPEC      ! CRA6 SPEC      !
!P! ON SHIFT       ! EN 40          ! EN 20          ! EN 10          !
!E! PATHS          ! E102 ON DPE5  ! E101 ON DPE5  ! E410 ON DPMA  !
!C! (SEE DPE1)    !                ! E411 ON DPMA  ! E113 ON CRA2  !
=====
!O! NORMAL        ! CRY 18 INH    ! PREVIOUS       ! #              !
=====
!1! ZERO          ! IR LOAD       ! XR LOAD        ! CLR 1 MSEC     !
=====
!2! ONES          ! <SPARE>       ! <SPARE>        ! CLR IO LATCH   !
=====
!3! ROT          ! PI LOAD       ! APR FLAGS      ! CLR IO BUSY    !
=====
!4! ASHC         ! ASH TEST      ! SET SWEEP      ! PAGE WRITE     !
=====
!5! LSHC         ! EXP TEST      ! APR EN         ! NICOND         !
=====
!6! DIV          ! PC FLAGS      ! PXCT OFF       ! PXCT EN        !
=====
!7! ROTC         ! AC BLOCKS EN ! MEM CLR        ! MEM WAIT       !
=====

```

THE DPM BOARD USES THE SPEC FIELD TO CONTROL THE DBM MIXER, AS FOLLOWS:

```

=====
! S !
! P ! ACTION WHEN DBM
! E ! SELECTS DP
! C ! GET DP BITS ! GET SCAD 1-7 !
=====
! O ! ALL ! NONE !
=====
! 1 ! 7-35 ! 0-6 !
=====
! 2 ! 0-6 AND 14-35 ! 7-13 !
=====
! 3 ! 0-13 AND 21-35 ! 14-20 !
=====
! 4 ! 0-20 AND 28-35 ! 21-27 !
=====
! 5 ! 0-27 AND 35 ! 28-34 !
=====
! 6 ! SAME AS ZERO !
=====
! 7 ! SAME AS ZERO !
=====

```

; 784 ;THE SPEC FIELD IS DEFINED AS A 6-BIT FIELD. THE TOP 3 BITS
; 785 ; ARE SPEC SEL A, SPEC SEL B, AND SPEC SEL C. THE LOW 3 BITS ARE
; 786 ; THE SELECT CODE.

; 787
; 788 SPEC/= <51:56>D,0 ;DPE1 & DPE5 & DPM1 & DPMA
; 789 #=10 ;DECODE # BITS
; 790 CLRCLK=11 ;CLEAR 1MS NICOND FLAG
; 791 CLR IO LATCH=12 ;CLEAR IO LATCH
; 792 CLR IO BUSY=13 ;CLEAR IO BUSY
; 793 LDPAGE=14 ;WRITE PAGE TABLE
; 794 NICOND=15 ;DOING NICOND DISPATCH
; 795 LDPXCT=16 ;LOAD PXCT FLAGS
; 796 WAIT=17 ;MEM WAIT
; 797 PREV=20 ;FORCE PREVIOUS CONTEXT
; 798 LOADXR=21 ;LOAD XR #, USES PXCT FIELD TO SELECT
; 799 ; CORRECT AC BLOCK
; 800 APR FLAGS=23 ;LOAD APR FLAGS
; 801 CLRCSH=24 ;CLEAR CACHE
; 802 APR EN=25 ;SET APR ENABLES
; 803 MEMCLR=27 ;CLEAR PAGE FAULT CONDITION
; 804 SWEEP=34 ;SET SWEEP
; 805 PXCT OFF=36 ;TURN OFF THE EFFECT OF PXCT
; 806 INHCRY18=40 ;INHIBIT CARRY INTO LEFT HALF
; 807 LOADIR=41 ;LOAD THE IR
; 808 LDPI=43 ;LOAD PI SYSTEM
; 809 ASHOV=44 ;TEST RESULT OF ASH
; 810 EXPTST=45 ;TEST RESULT OF FLOATING POINT
; 811 FLAGS=46 ;CHANGE PC FLAGS
; 812 LDACBLK=47 ;LOAD AC BLOCK NUMBERS
; 813 LDINST=61 ;LOAD INSTRUCTION

; 814
; 815 ;THE SPEC FIELD IS REDEFINED WHEN USED FOR BYTE MODE STUFF
; 816 BYTE/= <54:56> ;DPM1 (SPEC SEL)

; 817 BYTE1=1
; 818 BYTE2=2
; 819 BYTE3=3
; 820 BYTE4=4
; 821 BYTE5=5

; 822
; 823 ;THE SPEC FIELD IS REDEFINED WHEN USED TO CONTROL SHIFT PATHS
; 824 SHSTYLE/= <54:56> ;DPE1 (SPEC SEL)
; 825 NORM=0 ;2 40-BIT REGISTERS
; 826 ZERO=1 ;SHIFT ZERO INTO 36 BITS (ASH TOP 2901)
; 827 ONES=2 ;SHIFT IN ONES
; 828 ROT=3 ;ROTATE
; 829 ASHC=4 ;ASHC
; 830 LSHC=5 ;LSHC
; 831 DIV=6 ;SPECIAL DIVIDE
; 832 ROTC=7 ;ROTATE DOUBLE
; 833

```

; 834 .TOC "MICROCODE FIELDS -- DISPATCH"
; 835 ;
; 836 ; ! D ! CRA1 ! CRA1 ! DPEA !
; 837 ; ! I ! DISP ! DISP ! DISP !
; 838 ; ! S ! 10 ! 20 ! 40 !
; 839 ; ! P ! ! ! ! !
; 840 ;
; 841 ; ! O ! DIAG ADR ! DIAG ADR ! O !
; 842 ;
; 843 ; ! 1 ! RETURN ! RETURN ! DP 18-21 !
; 844 ;
; 845 ; ! 2 ! MULTIPLY ! J ! J !
; 846 ;
; 847 ; ! 3 ! PAGE FAIL ! AREAD ! AREAD !
; 848 ;
; 849 ; ! 4 ! NICOND ! NOT USABLE ! NORM !
; 850 ;
; 851 ; ! 5 ! BYTE ! NOT USABLE ! DP 32-35 !
; 852 ;
; 853 ; ! 6 ! EA MODE ! NOT USABLE ! DROM A !
; 854 ;
; 855 ; ! 7 ! SCAD ! NOT USABLE ! DROM B !
; 856 ;

```

```

; 857 ;NOTE: DISP EN 40 & DISP EN 10 ONLY CONTROL THE LOW 4 BITS OF THE
; 858 ; JUMP ADDRESS. DISP EN 20 ONLY CONTROLS THE HI 7 BITS. TO DO
; 859 ; SOMETHING TO ALL 11 BITS BOTH 20 & 40 OR 20 & 10 MUST BE ENABLED.
; 860 ;

```

```

; 861 DISP/= <57:62>D,70 ;CRA1 & DPEA
; 862 CONSOLE=00 ;CONSOLE DISPATCH
; 863 DROM=12 ;DROM
; 864 AREAD=13 ;AREAD
; 865 DP LEFT=31 ;DP 18-21
; 866 NORM=34 ;NORMALIZE
; 867 DP=35 ;DP 32-35
; 868 ADISP=36 ;DROM A FIELD
; 869 BDISP=37 ;DROM B FIELD
; 870 RETURN=41 ;RETURN
; 871 MUL=62 ;MULTIPLY
; 872 PAGE FAIL=63 ;PAGE FAIL
; 873 NICOND=64 ;NEXT INSTRUCTION DISPATCH
; 874 BYTE=65 ;BYTE SIZE AND POSITION
; 875 EAMODE=66 ;EFFECTIVE ADDRESS MODE
; 876 SCADO=67 ;J12 IF SCAD BIT 0 = 1
; 877

```

```

; 878 .TOC "MICROCODE FIELDS -- SKIP"
; 879 ;
; 880 ; ! S ! CRA2 ! DPEA ! DPEA !
; 881 ; ! K ! SKIP ! SKIP ! SKIP !
; 882 ; ! I ! 10 ! 20 ! 40 !
; 883 ; ! P ! ! ! ! !
; 884 ; !-----!
; 885 ; ! 0 ! 0 ! 0 ! 0 !
; 886 ; !-----!
; 887 ; ! 1 ! TRAP CYCLE ! CRY O2 ! CARRY OUT !
; 888 ; !-----!
; 889 ; ! 2 ! AD=0 ! ADL SIGN ! ADL=0 !
; 890 ; !-----!
; 891 ; ! 3 ! SC SIGN ! ADR SIGN ! ADR=0 !
; 892 ; !-----!
; 893 ; ! 4 ! EXECUTE ! USER IOT ! -USER !
; 894 ; !-----!
; 895 ; ! 5 ! -BUS IO BUSY ! JFCL SKIP ! FPD FLAG !
; 896 ; !-----!
; 897 ; ! 6 ! -CONTINUE ! CRY O1 ! AC # IS ZERO !
; 898 ; !-----!
; 899 ; ! 7 ! -1 MSEC ! TXXX ! INTERRUPT REQ !
; 900 ; !-----!

```

```

; 901
; 902 SKIP/= <63:68>D,70 ;CRA2 & DPEA
; 903 IOLGL=04 ;(.NOT.USER)!(USER IOT)!(CONSOLE EXECUTE MODE)
; 904 LLE=12 ;AD LEFT .LE. 0
; 905 CRYO=31 ;AD CRY -2
; 906 ADLEQO=32 ;ADDER LEFT = 0
; 907 ADREQO=33 ;ADDER RIGHT = 0
; 908 KERNEL=34 ;.NOT. USER
; 909 FPD=35 ;FIRST PART DONE
; 910 ACO=36 ;AC NUMBER IS ZERO
; 911 INT=37 ;INTERRUPT REQUEST
; 912 LE=42 ;(AD SIGN)!(AD.EQ.O)
; 913 CRY2=51 ;AD CRY O2
; 914 DPO=52 ;AD SIGN
; 915 DP18=53 ;AD BIT 18
; 916 IOT=54 ;USER IOT
; 917 JFCL=55 ;JFCL SKIP
; 918 CRY1=56 ;AD CRY 1
; 919 TXXX=57 ;TEST INSTRUCTION SHOULD SKIP
; 920 TRAP CYCLE=61 ;THIS INSTRUCTION IS THE RESULT OF A
; 921 ; TRAP 1, 2, OR 3
; 922 ADEQO=62 ;AD.EQ.O
; 923 SC=63 ;SC SIGN BIT
; 924 EXECUTE=64 ;CONSOLE EXECUTE MODE
; 925 -IO BUSY=65 ;.NOT. I/O LATCH
; 926 -CONTINUE=66 ;.NOT. CONTINUE
; 927 -1 MS=67 ;.NOT. 1 MS. TIMER
; 928

```


; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 24
MICROCODE FIELDS -- TIME CONTROL

```
; 929 .TOC "MICROCODE FIELDS -- TIME CONTROL"
; 930
; 931 DT/=<109:111>D,0 ;**NOT STORED IN CRAM**
; 932 ;DEFAULT TIME FIELD (USED IN MACROS)
; 933 ; CAN BE OVERRIDDEN IN MACRO CALL
; 934 2T=0
; 935 3T=1
; 936 4T=2
; 937 5T=3
; 938
; 939
; 940 T/=<69:71>F,DT ;CSL5 (E601)
; 941 ;CLOCK TICKS MINUS TWO REQUIRED TO
; 942 ; DO A MICRO INSTRUCTION
; 943 2T=0 ;TWO TICKS
; 944 3T=1 ;THREE TICKS
; 945 4T=2 ;FOUR TICKS
; 946 5T=3 ;FIVE TICKS
; 947
; 948
```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 25
MICROCODE FIELDS -- RANDOM CONTROL BITS

```
; 949 .TOC "MICROCODE FIELDS -- RANDOM CONTROL BITS"  
; 950  
; 951 CRY38/=<72> ;DPE5  
; 952 ;INJECT A CARRY INTO THE 2901 ADDER  
; 953 LOADSC/=<73> ;DPM4  
; 954 ;LOAD THE STEP COUNTER FROM THE SCAD  
; 955 LOADFE/=<74> ;DPM4  
; 956 ;LOAD THE FE REGISTER FROM THE SCAD  
; 957 FMWRITE/=<75> ;DPE5 (E302)  
; 958 ;WRITE THE RAM FILE.  
; 959 MEM/=<76> ;DPM5 (E612) & DPE5 (E205)  
; 960 ;START (OR COMPLETE) A MEMORY OR I/O CYCLE UNDER  
; 961 ; CONTROL OF THE NUMBER FIELD.  
; 962 DIVIDE/=<77> ;DPE5  
; 963 ;THIS MICROINSTRUCTION IS DOING A DIVIDE  
; 964 MULTI PREC/=<78> ;DPE5  
; 965 ;MULTIPRECISION STEP IN DIVIDE, DFAD, DFSB  
; 966 MULTI SHIFT/=<79> ;CSL5 (HAS NOTHING TO DO WITH DPE5 MULTI SHIFT)  
; 967 ;FAST SHIFT  
; 968 CALL/=<80> ;CRA2 (STACK IS ON CRA3)  
; 969 ;THIS IS A CALL  
; 970
```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 26
MICROCODE FIELDS -- NUMBER FIELD

```
; 971 .TOC "MICROCODE FIELDS -- NUMBER FIELD"
; 972
; 973 ;HERE IS THE GENERAL FIELD
; 974 #/= <90:107> ;MANY PLACES
; 975
; 976 ;# REDEFINED WHEN USED AS SCAD CONTROL:
; 977 SCAD/= <90:92> ;DPM3
; 978 A*2=0
; 979 A.OR.B=1
; 980 A-B-1=2
; 981 A-B=3
; 982 A+B=4
; 983 A.AND.B=5
; 984 A-1=6
; 985 A=7
; 986 SCADA/= <93:95> ;DPM3
; 987 SC=0
; 988 S#=1
; 989 PTR44=2 ;44 AND BIT 6 (SEE DPM3)
; 990 BYTE1=3
; 991 BYTE2=4
; 992 BYTE3=5
; 993 BYTE4=6
; 994 BYTE5=7
; 995 SCADB/= <96:97> ;DPM3
; 996 FE=0
; 997 EXP=1
; 998 SHIFT=2
; 999 SIZE=3
; 1000 S#/= <98:107> ;DPM3
; 1001
; 1002 ;# REDEFINED WHEN USED AS STATE REGISTER CONTROL:
; 1003 STATE/= <90:107> ;NOT USED BY HARDWARE
; 1004 SIMPLE=0 ;SIMPLE INSTRUCTIONS
; 1005 BLT=1 ;BLT IN PROGRESS
; 1006 MAP=400002 ;MAP IN PROGRESS
; 1007 SRC=3 ;MOVE STRING SOURCE IN PROGRESS
; 1008 DST=4 ;MOVE STRING FILL IN PROGRESS
; 1009 SRC+DST=5 ;MOVE STRING DEST IN PROGRESS
; 1010 DSTF=6 ;FILLING DEST
; 1011 CVTDB=7 ;CONVERT DEC TO BIN
; 1012 COMP-DST=10 ;COMPARE DEST
; 1013 EDIT-SRC=11 ;EDIT SOURCE
; 1014 EDIT-DST=12 ;EDIT DEST
; 1015 EDIT-S+D=13 ;BOTH SRC AND DST POINTERS
; 1016
```

```

; 1017 ;# REDEFINED WHEN USED AS WORSPACE ADDRESS
; 1018
; 1019 WORK/=<98:107> ;DPE6
; 1020 BADWO=160 ;AC BLK 7 WORD 0 (BAD DATA FROM MEMORY)
; 1021 BADW1=161 ;AC BLK 7 WORD 1 (BAD DATA FROM MEMORY)
; 1022 MUL=200 ;TEMP FOR MULTIPLY
; 1023 DIV=201 ;TEMP FOR DIVIDE
; 1024 SV.VMA=210 ;SAVE VMA
; 1025 SV.AR=211 ;SAVE AR
; 1026 SV.ARX=212 ;SAVE ARX
; 1027 SV.BR=213 ;SAVE BR
; 1028 SV.BRX=214 ;SAVE BRX
; 1029 SBR=215 ;SPT BASE REGISTER
; 1030 CBR=216 ;CST BASE ADDRESS
; 1031 CSTM=217 ;CST MASK
; 1032 PUR=220 ;PROCESS USE REGISTER
; 1033 ADJP=221 ;"P" FOR ADJBP
; 1034 ADJS=222 ;"S" FOR ADJBP
; 1035 ADJPTR=223 ;BYTE POINTER FOR ADJBP
; 1036 ADJQ1=224 ;TEMP FOR ADJBP
; 1037 ADJR2=225 ;TEMP FOR ADJBP
; 1038 ADJBPW=226 ;(BYTES/WORD) FOR ADJBP
; 1039 HSBADR=227 ;ADDRESS OF HALT STATUS BLOCK
; 1040 APR=230 ;APR ENABLES
; 1041 ;THE FOLLOWING WORDS ARE USED BY EXTEND INSTRUCTION
; 1042 EO=240 ;ORIGINAL EFFECTIVE ADDRESS
; 1043 E1=241 ;EFFECTIVE ADDRESS OF WORD AT EO
; 1044 SLEN=242 ;SOURCE LENGTH
; 1045 MSK=243 ;BYTE MASK
; 1046 FILL=244 ;FILL BYTE
; 1047 CMS=245 ;SRC BYTE IN STRING COMPARE
; 1048 FSIG=246 ;PLACE TO SAVE ARX WHILE STORING
; 1049 ; THE FLOAT CHAR
; 1050 BDH=247 ;BINARY BEING CONVERTED TO
; 1051 BDL=250 ; DECIMAL
; 1052
; 1053 ;TIMER STUFF
; 1054 TIME0=300 ;HIGH ORDER 36 BITS OF TIME
; 1055 TIME1=301 ;LOW ORDER 36 BITS OF TIME
; 1056 PERIOD=302 ;INTERRUPT PERIOD
; 1057 TTG=303 ;TIME TO GO TO NEXT INTERRUPT
; 1058

```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; KS10.MIC[10,1141] 07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 28
MICROCODE FIELDS -- NUMBER FIELD

```
; 1059 ;DDIV STUFF
; 1060     ACO=314
; 1061     AC1=315
; 1062     AC2=316
; 1063     AC3=317
; 1064     DDIV SGN=320
; 1065     DVSOR H=321
; 1066     DVSOR L=322
; 1067 ;POWERS OF TEN
; 1068     DECL0=344     ;LOW WORD
; 1069     DECHI=373     ;HIGH WORD
; 1070
; 1071     YSAVE=422     ;Y OF LAST INDIRECT POINTER
; 1072     PTA.E=423     ;ADDRESS OF EXEC PAGE MAP (NOT PROCESS TABLE)
; 1073     PTA.U=424     ;ADDRESS OF USER PAGE MAP
; 1074     TRAPPC=425    ;SAVED PC FROM TRAP CYCLE
; 1075     SV.AR1=426    ;ANOTHER PLACE TO SAVE AR
; 1076
```

```
; 1077 ;# REDEFINED WHEN USED AS PC FLAG CONTROL (ALL ON DPE9)
; 1078
; 1079 SETOV/=<90> ;DPE9
; 1080 ;SET ARITHMETIC OVERFLOW
; 1081 SETFOV/=<91> ;SET FLOATING OVERFLOW
; 1082 SETNDV/=<92> ;SET NO DIVIDE
; 1083
; 1084 -----
; 1085
; 1086 CLRFPD/=<93> ;CLEAR FIRST PART DONE
; 1087 SETFPD/=<94> ;SET FIRST PART DONE
; 1088 HOLD USER/=<95> ;WHEN THIS BIT IS SET IT:
; 1089 ; 1. PREVENTS SETTING USER IOT IN USER MODE
; 1090 ; 2. PREVENTS CLEARING USER IN USER MODE
; 1091
; 1092 -----
; 1093
; 1094 ; <96> ;SPARE
; 1095 TRAP2/=<97> ;SET TRAP 2
; 1096 TRAP1/=<98> ;SET TRAP 1
; 1097
; 1098 -----
; 1099
; 1100 LD PCU/=<99> ;LOAD PCU FROM USER
; 1101 ; <100> ;SPARE
; 1102 ; <101> ;SPARE
; 1103
; 1104 -----
; 1105
; 1106 ; <102> ;SPARE
; 1107 ; <103> ;SPARE
; 1108 JFCLFLG/=<104> ;DO A JFCL INSTRUCTION
; 1109
; 1110 -----
; 1111
; 1112 LD FLAGS/=<105> ;LOAD FLAGS FROM DP
; 1113 ; <106>
; 1114 ADFLGS/=<107> ;UPDATE CARRY FLAGS
; 1115
```

```
; 1116 ;# REDEFINED WHEN USED AS MEMORY CYCLE CONTROL
; 1117
; 1118 FORCE USER/= <90> ;FORCE USER MODE REFERENCE
; 1119 FORCE EXEC/= <91> ;FORCE EXEC MODE REFERENCE
; 1120 ; (DOES NOT WORK UNDER PXCT)
; 1121 FETCH/= <92> ;THIS IS AN INSTRUCTION FETCH
; 1122
; 1123 ;-----
; 1124
; 1125 READ CYCLE/= <93> ;SELECT A READ CYCLE
; 1126 WRITE TEST/= <94> ;PAGE FAIL IF NOT WRITTEN
; 1127 WRITE CYCLE/= <95> ;SELECT A MEMORY WRITE CYCLE
; 1128
; 1129 ;-----
; 1130
; 1131 ; <96> ;SPARE BIT
; 1132 DONT CACHE/= <97> ;DO NOT LOOK IN CACHE
; 1133 PHYSICAL/= <98> ;DO NOT INVOKE PAGING HARDWARE
; 1134
; 1135 ;-----
; 1136
; 1137 PXCT/= <99:101> ;WHICH PXCT BITS TO LOOK AT
; 1138 CURRENT=0
; 1139 E1=1
; 1140 D1=3
; 1141 BIS-SRC-EA=4
; 1142 E2=5
; 1143 BIS-DST-EA=6
; 1144 D2=7
; 1145
; 1146 ;-----
; 1147
; 1148 AREAD/= <102> ;LET DROM SELECT SYSLE TYPE AND VMA LOAD
; 1149 DP FUNC/= <103> ;IGNORE # BITS 0-11 AND USE DP 0-13 INSTEAD
; 1150 ; DP9 MEANS "FORCE PREVIOUS"
; 1151 LDVMA/= <104> ;LOAD THE VMA
; 1152
; 1153 ;-----
; 1154
; 1155 EXT ADR/= <105> ;PUT VMA BITS 14-17 ONTO BUS
; 1156 WAIT/= <106> ;START A MEMORY OR I/O CYCLE
; 1157 BWRITE/= <107> ;START A MEMORY CYCLE IF DROM ASKS FOR IT
; 1158
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 31
; KS10.MIC[10,1141] 07:10 27-JULY-1984 MICROCODE FIELDS -- NUMBER FIELD

; 1159 ;THESE BITS ARE USED ONLY TO SETUP DP FOR A DP FUNCTION

; 1160

; 1161 ; <99> ;PREVIOUS

; 1162 IO CYCLE/= <100> ;THIS IS AN I/O CYCLE

; 1163 WRU CYCLE/= <101> ;WHO ARE YOU CYCLE

; 1164

; 1165 ;-----

; 1166

; 1167 VECTOR CYCLE/= <102> ;READ INTERRUPT VECTOR

; 1168 IO BYTE/= <103> ;BYTE CYCLE

; 1169 ; <104>

; 1170

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 32
MICROCODE FIELDS -- NUMBER FIELD

```
; 1171 ;# REDEFINED WHEN USED AS PI RIGHT BITS
; 1172 PI.ZER/=<90:92> ;ZEROS
; 1173 PI.IP1/=<93> ;PI 1 IN PROG
; 1174 PI.IP2/=<94>
; 1175 PI.IP3/=<95>
; 1176 PI.IP4/=<96>
; 1177 PI.IP5/=<97>
; 1178 PI.IP6/=<98>
; 1179 PI.IP7/=<99>
; 1180 PI.ON/=<100> ;SYSTEM IS ON
; 1181 PI.CO1/=<101> ;CHAN 1 IS ON
; 1182 PI.CO2/=<102>
; 1183 I.CO3/=<103>
; 1184 I.CO4/=<104>
; 1185 I.CO5/=<105>
; 1186 I.CO6/=<106>
; 1187 I.CO7/=<107>
; 1188
; 1189 ;# REDEFINED WHEN USED AS WRPI DATA
; 1190 PI.MBZ/=<90:93> ;MUST BE ZERO
; 1191 PI.DIR/=<94> ;DROP INTERRUPT REQUESTS
; 1192 PI.CLR/=<95> ;CLEAR SYSTEM
; 1193 PI.REQ/=<96> ;REQUEST INTERRUPT
; 1194 PI.TCN/=<97> ;TURN CHANNEL ON
; 1195 PI.TCF/=<98> ;TURN CHANNEL OFF
; 1196 PI.TSF/=<99> ;TURN SYSTEM OFF
; 1197 PI.TSN/=<100> ;TURN SYSTEM ON
; 1198 PI.SC1/=<101> ;SELECT CHANNEL 1
; 1199 PI.SC2/=<102>
; 1200 PI.SC3/=<103>
; 1201 PI.SC4/=<104>
; 1202 PI.SC5/=<105>
; 1203 PI.SC6/=<106>
; 1204 PI.SC7/=<107>
; 1205
```

```

; 1206 ;# REDEFINED WHEN USED AS AC CONTROL
; 1207
; 1208
; 1209 ;THIS FIELD CONTROLS THE INPUT TO A 74LS181 ON DPE6. THE NUMBER
; 1210 ; FIELD HAS THIS FORMAT IN <98:107>:
; 1211 ;
; 1212 ;      !-----!-----!-----!-----!-----!-----!-----!-----!
; 1213 ;      !CARRY! S8 ! S4 ! S2 ! S1 ! MODE! B8 ! B4 ! B2 ! B1 !
; 1214 ;      ! IN !      FUNCTION !      DATA INPUTS !
; 1215 ;      !-----!-----!-----!-----!-----!-----!-----!-----!
; 1216 ;
; 1217
; 1218 ACALU/=<98:103>
; 1219      B=25
; 1220      AC+N=62
; 1221 ACN/=<104:107>
; 1222      ;AC NAMES FOR STRING INSTRUCTIONS
; 1223      SRCLEN=0      ;SOURCE LENGTH
; 1224      SRCP=1      ;SOURCE POINTER
; 1225      DLEN=3      ;DEST LENGTH
; 1226      DSTP=4      ;DEST POINTER
; 1227      MARK=3      ;POINTER TO MARK
; 1228      BINO=3      ;HIGH WORD OF BINARY
; 1229      BIN1=4      ;LOW WORD OF BINARY
; 1230

```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 34
MICROCODE FIELDS -- NUMBER FIELD

```
; 1231 ;# FIELD REDEFINED WHEN USE AS APRID DATA
; 1232 MICROCODE OPTIONS/=<90:98>
; 1233 ;100 - NON-STANDARD MICROCODE
; 1234 ;200 - NO CST AT ALL
; 1235 ;400 - INHIBIT CST UPDATE IS AVAILABLE
; 1236 ;040 - UBABLT INSTRUCTIONS ARE PRESENT
; 1237 ;020 - KI PAGING IS PRESENT
; 1238 ;010 - KL PAGING IS PRESENT
; 1239 MICROCODE OPTION(INHCST)/=<90>
;; 1240 .IF/INHCST
; 1241 .OPT=1
; 1242 .IFNOT/INHCST
; 1243 .OPT=0
; 1244 .ENDIF/INHCST
; 1245 MICROCODE OPTION(NOCST)/=<91>
;; 1246 .IF/NOCST
; 1247 .OPT=1
; 1248 .IFNOT/NOCST
; 1249 .OPT=0
; 1250 .ENDIF/NOCST
; 1251 MICROCODE OPTION(NONSTD)/=<92>
; 1252 .IF/NONSTD
; 1253 .OPT=1
; 1254 .IFNOT/NONSTD
; 1255 .OPT=0
; 1256 .ENDIF/NONSTD
; 1257 MICROCODE OPTION(UBABLT)/=<93>
; 1258 .IF/UBABLT
; 1259 .OPT=1
; 1260 .IFNOT/UBABLT
; 1261 .OPT=0
; 1262 .ENDIF/UBABLT
; 1263 MICROCODE OPTION(KIPAGE)/=<94>
; 1264 .IF/KIPAGE
; 1265 .OPT=1
; 1266 .IFNOT/KIPAGE
; 1267 .OPT=0
; 1268 .ENDIF/KIPAGE
; 1269 MICROCODE OPTION(KLPAGE)/=<95>
;; 1270 .IF/KLPAGE
; 1271 .OPT=1
; 1272 .IFNOT/KLPAGE
; 1273 .OPT=0
; 1274 .ENDIF/KLPAGE
; 1275
; 1276 MICROCODE VERSION/=<99:107>
; 1277 .UCV=124
; 1278
```

```
; 1279 ;# FIELD REDEFINED WHEN USED AS A HALT CODE
; 1280
; 1281 HALT/= <90:107>
; 1282 ; CODES 0 TO 77 ARE "NORMAL" HALTS
; 1283 POWER=0 ; POWER UP
; 1284 HALT=1 ; HALT INSTRUCTION
; 1285 CSL=2 ; CONSOLE HALT
; 1286 ; CODES 100 TO 777 ARE SOFTWARE ERRORS
; 1287 IOPF=100 ; I/O PAGE FAIL
; 1288 ILLII=101 ; ILLEGAL INTERRUPT INSTRUCTION
; 1289 ILLINT=102 ; BAD POINTER TO UNIBUS INTERRUPT VECTOR
; 1290 ; CODES 1000 TO 1777 ARE HARDWARE ERRORS
; 1291 BW14=1000 ; ILLEGAL BWRITE FUNCTION (BAD DROM)
; 1292 NICOND 5=1004 ; ILLEGAL NICOND DISPATCH
; 1293 MULERR=1005 ; VALUE COMPUTED FOR 10**21 WAS WRONG
; 1294 .IFNOT/FULL
; 1295 PAGEF=1777 ; PAGE FAIL IN SMALL MICROCODE
; 1296 .ENDIF/FULL
; 1297
; 1298
; 1299
; 1300 ;# FIELD REDEFINED WHEN USED AS FLG BITS
; 1301
; 1302 FLG.W/= <94> ; W BIT FROM PAGE MAP
; 1303 FLG.PI/= <95> ; PI CYCLE
; 1304 FLG.C/= <96> ; CACHE BIT FROM PAGE MAP
; 1305 FLG.SN/= <97> ; SPECIAL NEGATE IN FDV & DFDV
; 1306
; 1307 ; RIGHT HALF OF FLG USED TO RECOVER FROM PAGE FAILS
; 1308
```

```
; 1309 .TOC "DISPATCH ROM DEFINITIONS"  
; 1310  
; 1311 ;ALL ON DPEA  
; 1312  
; 1313 .DCODE  
; 1314 A/=<2:5> ;OPERAND FETCH MODE  
; 1315 READ=0 ;READ  
; 1316 WRITE=1 ;WRITE  
; 1317 DREAD=2 ;DOUBLE READ  
; 1318 DBLAC=3 ;DOUBLE AC  
; 1319 SHIFT=4 ;SIMPLE SHIFT  
; 1320 DSHIFT=5 ;DOUBLE SHIFT  
; 1321 FPI=6 ;FLOATING POINT IMMEDIATE  
; 1322 FP=7 ;FLOATING POINT  
; 1323 RD-PF=10 ;READ, THEN START PREFETCH  
; 1324 DFP=11 ;DOUBLE FLOATING POINT  
; 1325 IOT=12 ;CHECK FOR IO LEGAL THEN SAME AS I  
; 1326  
; 1327 B/=<8:11> ;STORE RESULTS AS  
; 1328 SELF=4 ;SELF  
; 1329 DBLAC=5 ;DOUBLE AC  
; 1330 DBLB=6 ;DOUBLE BOTH  
; 1331 AC=15 ;AC  
; 1332 MEM=16 ;MEMORY  
; 1333 BOTH=17 ;BOTH  
; 1334  
; 1335 ;B-FIELD WHEN USED IN FLOATING POINT OPERATIONS  
; 1336 ROUND/=<8> ;ROUND THE RESULT  
; 1337 MODE/=<9> ;SEPARATE ADD/SUB & MUL/DIV ETC.  
; 1338 FL-B/=<10:11> ;STORE RESULTS AS  
; 1339 AC=1 ;AC  
; 1340 MEM=2 ;MEMORY  
; 1341 BOTH=3 ;BOTH  
; 1342  
; 1343 J/=<12:23> ;DISPATCH ADDRESS (MUST BE 1400 TO 1777)  
; 1344  
; 1345 ACDISP/=<24> ;DISPATCH ON AC FIELD  
; 1346 I/=<25> ;IMMEDIATE DISPATCH. DISP/AREAD DOES A DISP/DROM  
; 1347 ; IF THIS BIT IS SET.  
; 1348 READ/=<26> ;START A READ AT AREAD  
; 1349 TEST/=<27> ;START A WRITE TEST AT AREAD  
; 1350 COND FUNC/=<28> ;START A MEMORY CYCLE ON BWRITE  
; 1351 VMA/=<29>D,1 ;LOAD THE VMA ON AREAD  
; 1352 WRITE/=<30> ;START A WRITE ON AREAD  
; 1353 .UCODE  
; 1354
```

; 1355 .TOC "HOW TO READ MACROS"

; 1356 ;
; 1357 ; 1.0 REGISTER TRANSFER MACROS

; 1358 ;
; 1359 ; MOST MACROS USED IN THE KS10 ARE USED TO OPERATE ON DATA IN (OR FROM/TO) 2901
; 1360 ; REGISTERS. THE NAMES OF THE 2901 REGISTERS ARE MACRO PARAMETERS AND ARE
; 1361 ; ENCLOSED IN []. A TYPICAL MACRO IS:

; 1362 ;
; 1363 ; [AR]_[AR]+[BR]

; 1364 ;
; 1365 ; THE SYMBOL _ IS PRONOUNCED "GETS". THE ABOVE MACRO WOULD BE READ "THE AR GETS
; 1366 ; THE AR PLUS THE BR".

; 1367 ;
; 1368 ; IF A MACRO DOES NOT HAVE A _ IN IT, THERE IS NO RESULT STORED. THUS, [AR]-[BR]
; 1369 ; JUST COMPARES THE AR AND THE BR AND ALLOWS FOR SKIPS ON THE VARIOUS ALU BITS.

; 1370 ;
; 1371 ;
; 1372 ;
; 1373 ; 1.1 SPECIAL SYMBOLS

; 1374 ;
; 1375 ; THERE ARE A BUNCH OF SYMBOLS USED IN THE MACROS WHICH ARE NOT 2901 REGISTERS.
; 1376 ; THEY ARE DEFINED HERE:

- ; 1377 ;
; 1378 ; 1. AC -- THE AC SELECTED BY THE CURRENT INSTRUCTION. SEE DPEA
; 1379 ;
; 1380 ; 2. AC[] -- AC+N. AC[1] IS AC+1, AC[2] IS AC+2, ETC.
; 1381 ;
; 1382 ; 3. APR -- THE APR FLAGS FROM DPMA
; 1383 ;
; 1384 ; 4. EA -- THE EFFECTIVE ADDRESS. THAT IS, 0 IN THE LEFT HALF AND THE
; 1385 ; CONTENTS OF THE HR IN THE RIGHT HALF.
; 1386 ;
; 1387 ; 5. EXP -- THE F.P. EXPONENT FROM THE SCAD. [AR]_EXP WILL TAKE THE
; 1388 ; EXPONENT OUT OF THE FE AND PUT IT BACK INTO THE NUMBER IN THE AR.
; 1389 ;
; 1390 ; 6. FE -- THE FE REGISTER
; 1391 ;
; 1392 ; 7. FLAGS -- THE PC FLAGS (FROM DPE9) IN THE LEFT HALF.
; 1393 ;
; 1394 ; 8. Q -- THE Q REGISTER
; 1395 ;
; 1396 ; 9. RAM -- THE RAM FILE, RAM ADDRESS IS IN THE VMA.
; 1397 ;
; 1398 ; 10. P -- THE P FIELD OF THE BYTE POINTER. SAME IDEA AS EXP.
; 1399 ;
; 1400 ; 11. TIME -- THE 1MS. TIMER
; 1401 ;
; 1402 ; 12. VMA -- THE VMA. WHEN READ IT INCLUDES THE VMA FLAGS
; 1403 ;
; 1404 ; 13. XR -- INDEX REGISTER
; 1405 ;

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 38
HOW TO READ MACROS

; 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 ;

14. XWD -- HALF WORD. USED TO GENERATE CONSTANTS. FOR EXAMPLE, [AR]_O XWD [40] WOULD LOAD THE CONSTANT 40 (OCTAL) INTO THE AR.
15. +SIGN AND -SIGN -- SIGN BITS USED TO SIGN SMEAR F.P. NUMBERS. FOR EXAMPLE, [AR]_+SIGN WOULD CLEAR AR BITS 0 TO 8.
16. WORK[] -- LOCATIONS IN THE WORKSPACE USED AS SCRATCH SPACE.* FOR EXAMPLE, [AR]_WORK[CSTM] WOULD LOAD THE AR WITH THE CST MASK FROM THE RAM. CSTM IS A SYMBOL DEFINED IN THE WORK FIELD.

1.2 LONG

LONG IS USED ON SHIFT OPERATIONS TO INDICATE THAT THE Q REGISTER IS ALSO SHIFTED. THIS SAYS NOTHING ABOUT HOW THE SHIFT PATHS ARE CONNECTED UP.

2.0 MEMORY MACROS

MEMORY IS INDICATED BY THE SYMBOL "MEM". WHEN WE ARE WAITING FOR DATA FROM MEMORY THE "MEM READ" MACRO IS USED. WHEN WE ARE SENDING DATA TO MEMORY, THE "MEM WRITE" MACRO IS USED. EXAMPLE,
MEM READ, ;WAIT FOR MEMORY
[AR]_MEM ;LOAD DATA INTO AR
VMA_ IS USED THE LOAD THE VMA. THUS, VMA_[PC] LOADS THE VMA FROM THE PC.

3.0 TIME CONTROL

THERE ARE 2 SETS OF MACROS USED FOR TIME CONTROL. THE FIRST, SELECTS THE RAM ADDRESS TO SPEED UP THE NEXT INSTRUCTION. THESE MACROS ARE AC, AC[], XR, VMA, WORK[]. THE SECOND, SETS THE TIME FIELD. THESE ARE 2T, 3T, 4T, AND 5T TO SELECT 2, 3, 4, OR 5 TICKS.

4.0 SCAD MACROS

THE SCAD MACROS LOOK LIKE THE 2901 MACROS EXECPT NO [] ARE REQUIRED. THERE ARE ONLY A FEW SYMBOLS USED.

1. FE -- THE FE REGISTER
2. SC -- THE SC REGISTER

- ; 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 ;
3. EXP -- THE EXPONENT FROM A F.P. NUMBER. FOR EXAMPLE FE_EXP LOADS THE FE FROM DP BITS 1-8.
 4. SHIFT -- THE SHIFT COUNT FROM SHIFT INSTRUCTIONS. THAT IS DP BITS 18 AND 28-35.
 5. S# -- THE SMALL NUMBER. THE 10 BIT MAGIC NUMBER INPUT TO THE SCADA MIXER.

5.0 CONTROL MACROS

ALL CONTROL MACROS LOOK LIKE ENGLISH COMMANDS. SOME EXAMPLES,
HOLD LEFT ;DO NOT CLOCK LEFT HALF OF DP
SET APR ENABLES ;LOAD APR ENABLES FROM DP
SET NO DIVIDE ;SET NO DIVIDE PC FLAG

6.0 SKIPS

ALL SKIPS CAUSE THE NEXT MICRO INSTRUCTION TO COME FROM THE ODD WORD OF AN EVEN/ODD PAIR. THE MACROS HAVE THE FORMAT OF SKIP COND. THEY SKIP IF CONDITION IS TRUE. SOME EXAMPLES,
SKIP AD.EQ.O ;SKIP IF ADDER OUTPUT IS ZERO
SKIP IRPT ;SKIP IF INTERRUPT IS PENDING

7.0 DISPATCH MACROS

DISPATCH MACROS CAUSE THE MACHINE TO GO TO ONE OF MANY PLACES. IN MOST CASES THEY HAVE THE WORD "DISP" IN THE NAME OF THE MACRO. FOR EXAMPLE, MUL DISP, BYTE DISP.

8.0 SUPER MACROS

THERE ARE PLACES WHERE ONE MICRO INSTRUCTION IS USED IN MANY PLACES. FOR EXAMPLE, MANY PLACES DETECT ILLEGAL OPERATIONS AND WANT TO GENERATE A TRAP TO THE MONITOR. WE COULD WRITE

J/UUO
BUT THIS WASTES A MICRO STEP DOING A USELESS JUMP. INSTEAD WE WRITE,
UUO

THIS MACRO IS THE FIRST STEP OF THE UUO ROUTINE AND JUMPS TO THE SECOND INSTRUCTION. WE WRITE THE EXPANSION OF THE UUO MACRO AS THE FIRST INSTRUCTION OF THE UUO ROUTINE SO THAT THE READER CAN SEE WHAT IT DOES. SOME EXAMPLES OF

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 40
HOW TO READ MACROS

; 1506 ;
; 1507 ;
; 1508 ;
; 1509 ;
; 1510 ;
; 1511 ;
; 1512 ;
; 1513 ;

SUPER MACROS ARE:
PAGE FAIL TRAP
DONE

HALT []

;GENERATE A PAGE FAIL TRAP
;THIS INSTRUCTION IS NOW COMPLETE
; USED WITH A SKIP OR DISP WHERE
; SOME PATHS ARE NOP'S
;JUMP TO HALT LOOP. ARGUMENT IS A
; CODE

```

; 1514 .TOC "MACROS -- DATA PATH CHIP -- GENERAL"
; 1515
; 1516 .NOT.[ ] "AD/.NOT.A,A/@1"
; 1517 [ ]+[ ] "AD/A+B,A/@1,B/@2"
; 1518 [ ]-[ ] "AD/A-B-.25,A/@1,B/@2,ADD .25"
; 1519 [ ]-# "AD/A-D-.25,DBUS/DBM,DBM/#,A/@1,ADD .25"
; 1520 [ ].AND.# "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1"
; 1521 [ ].AND.Q "AD/A.AND.Q,A/@1,DEST/PASS"
; 1522 [ ].AND.[ ] "AD/A.AND.B,A/@2,B/@1,DEST/PASS"
; 1523 [ ].AND.NOT.[ ] "AD/.NOT.A.AND.B,A/@2,B/@1,DEST/PASS"
; 1524 [ ].OR.[ ] "AD/A.OR.B,A/@2,B/@1,DEST/PASS"
; 1525 [ ].XOR.# "AD/D.XOR.A,DBUS/DBM,DBM/#,A/@1"
; 1526 [ ].XOR.[ ] "AD/A.XOR.B,A/@2,B/@1,DEST/PASS"
; 1527 [ ]-#[ ] "AD/D-A-.25,DEST/AD,A/@2,B/@1,DBUS/DBM,DBM/#,ADD .25"
; 1528 [ ]# "AD/D,DBUS/DBM,DBM/#,DEST/AD,B/@1"
; 1529 [ ]-1 "AD/-A-.25,A/ONE,DEST/AD,B/@1,ADD .25"
; 1530 [ ]-2 "AD/-A-.25,DEST/AD*2,A/ONE,B/@1,ADD .25"
; 1531 [ ]-Q "AD/-Q-.25,DEST/AD,B/@1,ADD .25"
; 1532 [ ]-Q*2 "AD/-Q-.25,DEST/AD*2,B/@1,ADD .25"
; 1533 [ ]-Q*.5 "AD/-Q-.25,DEST/AD*.5,B/@1,ADD .25"
; 1534 [ ]-[ ] "AD/-A-.25,A/@2,DEST/AD,B/@1,ADD .25"
; 1535 [ ]-[ ]-.25 "AD/-A-.25,A/@2,DEST/AD,B/@1"
; 1536 [ ]-[ ]*2 "AD/-A-.25,A/@2,DEST/AD*2,B/@1,ADD .25"
; 1537 [ ]-.NOT.AC "AD/.NOT.D,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1538 [ ]-.NOT.AC[ ] "AD/.NOT.D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,DT/3T"
; 1539 [ ]-.NOT.Q "AD/.NOT.Q,DEST/AD,B/@1"
; 1540 [ ]-.NOT.[ ] "AD/.NOT.A,A/@2,DEST/AD,B/@1"
; 1541 [ ]O "AD/ZERO,DEST/AD,B/@1"
; 1542 [ ]O*.5 LONG "AD/ZERO,DEST/Q_Q*.5,B/@1"
; 1543 [ ]O XWD [ ] "AD/47,DEST/AD,B/@1,DBM/#,DBUS/DBM,#/@2,RSRC/DA,A/MASK"
; 1544 [ ]_AC "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,AD PARITY"
; 1545 [ ]-AC "AD/-D-.25,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1546 [ ]-AC[ ] "AD/-D-.25,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,ADD .25,DT/3T"
; 1547 [ ]-AC*.5 "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD*.5,B/@1,DT/3T"
; 1548 [ ]-AC*.5 LONG "AD/D,DBUS/RAM,RAMADR/AC#,DEST/Q_Q*.5,B/@1,DT/3T"
; 1549 [ ]-AC*2 "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD*2,B/@1,DT/3T"
; 1550 [ ]-AC+1 "AD/D+A,DBUS/RAM,RAMADR/AC#,A/ONE,DEST/AD,B/@1"
; 1551 [ ]-AC+1000001 "AD/D+A,DBUS/RAM,RAMADR/AC#,A/XWD1,DEST/AD,B/@1"
; 1552 [ ]-AC+[ ] "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,DT/3T"
; 1553 [ ]-AC-1 "AD/D-A-.25,DBUS/RAM,RAMADR/AC#,A/ONE,DEST/AD,B/@1,ADD .25"
; 1554 [ ]-AC-[ ] "AD/D-A-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1555 [ ]-AC-[ ]-.25 "AD/D-A-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1556 [ ]-AC[ ]-[ ] "AD/D-A-.25,A/@3,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,ADD .25,DT/3T"
; 1557 [ ]-AC[ ]-1 "AD/D-A-.25,A/ONE,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,ADD .25,DT/3T"
; 1558 [ ]-AC[ ].AND.[ ] "AD/D.AND.A,A/@3,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,DT/3T"
; 1559 [ ]-AC.AND.MASK "AD/D.AND.A,A/MASK,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,AD PARITY"
; 1560 [ ]-AC[ ] "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,AD PARITY,DT/3T"
; 1561 [ ]-AC[ ]*2 "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD*2,B/@1,AD PARITY,DT/3T"
; 1562 [ ]-AC[ ]*.5 "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD*.5,B/@1,AD PARITY,DT/3T"
; 1563 [ ]_APR "AD/D,DBUS/DBM,DBM/APR FLAGS,DEST/AD,B/@1,DT/3T"
; 1564 [ ]_CURRENT AC [ ] "AD/D,DBUS/RAM,RAMADR/#,ACALU/B,ACN/@2,DEST/AD,B/@1,AD PARITY,DT/3T"
; 1565 [ ]_EA FROM [ ] "AD/57,RSRC/OA,A/@2,DEST/AD,B/@1"
; 1566

```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 42
MACROS -- DATA PATH CHIP -- GENERAL

```
; 1567 [ ]_EA "AD/57,RSRC/OA,A/HR,DEST/AD,B/@1"
; 1568 [ ]_EXP "AD/D,DBUS/DBM,DBM/EXP,A/@1,B/@1,DEST/A,SCAD/A+B,SCADA/S#,S#/O,SCADB/FE,HOLD RIGHT,EXP TEST"
; 1569 [ ]_FE "AD/D,DEST/AD*.5,B/@1,DBUS/DBM,DBM/DP,SCAD/A+B,SCADA/S#,S#/O,SCADB/FE,BYTE/BYTES"
; 1570 [ ]_FLAGS "AD/D.AND.A,DBUS/PC_FLAGS,A/MASK,DEST/AD,B/@1,RSRC/OQ"
; 1571 [ ]_P "AD/D,DEST/A,A/@1,B/@1,DBUS/DBM,DBM/DP,BYTE/BYTE1,SCAD/A+B,SCADA/S#,S#/O,SCADB/FE"
; 1572 [ ]_PC WITH FLAGS "AD/D,DBUS/PC_FLAGS,RSRC/OA,A/PC,DEST/AD,B/@1"
; 1573 [ ]_Q "AD/Q,DEST/AD,B/@1"
; 1574 [ ]_Q*.5 "AD/Q,DEST/AD*.5,B/@1"
; 1575 [ ]_Q*2 "AD/Q,DEST/AD*2,B/@1"
; 1576 [ ]_Q*2 LONG "AD/Q,DEST/Q_Q*2,B/@1"
; 1577 [ ]_Q+1 "AD/A+Q,A/ONE,DEST/AD,B/@1"
; 1578 [ ]_RAM "AD/D,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1,AD PARITY"
; 1579 [ ]_TIME "AD/44,RSRC/DA,A/MASK,DBUS/DBM,DBM/EXP,DEST/AD,B/@1"
; 1580 [ ]_VMA "AD/D,DEST/AD,B/@1,DBUS/DBM,DBM/VMA"
; 1581 [ ]_XR "AD/D,DBUS/RAM,RAMADR/XR#,DEST/AD,B/@1"
; 1582 [ ]_["] "AD/A,A/@2,DEST/AD,B/@1"
; 1583 [ ]_["] SWAP "AD/D,DBUS/DBM,DBM/DP_SWAP,DEST/A,A/@2,B/@1"
; 1584 [ ]_["] XWD O "AD/45,DEST/AD,B/@1,DBM/DBM,DBUS/DBM,##/@2,RSRC/DO,A/MASK"
; 1585 [ ]_["] *.5 "AD/A,A/@2,DEST/AD*.5,B/@1"
; 1586 [ ]_["] *.5 LONG "AD/A,A/@2,DEST/Q_Q*.5,B/@1"
; 1587 [ ]_["] *2 "AD/A,A/@2,DEST/AD*2,B/@1"
; 1588 [ ]_["] *2 LONG "AD/A,A/@2,DEST/Q_Q*2,B/@1"
; 1589 [ ]_["] *4 "AD/A+B,A/@2,B/@1,DEST/AD*2"
; 1590 [ ]_["] +# "AD/D+A,DBUS/DBM,DBM/##,A/@2,DEST/AD,B/@1"
; 1591 [ ]_["] +.25 "AD/O+A,A/@2,DEST/AD,B/@1,ADD .25"
; 1592 [ ]_["] +O "AD/O+A,A/@2,DEST/AD,B/@1"
; 1593 [ ]_["] +1 "AD/A+B,A/ONE,B/@1,B/@2,DEST/AD"
; 1594 [ ]_["] +1000001 "AD/D+A,A/@2,DBUS/DBM,DBM/##,##/1,DEST/AD,B/@1"
; 1595 [ ]_["] +AC "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1596 [ ]_["] +AC[ ] "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@3,DEST/AD,B/@1,DT/3T"
; 1597 [ ]_["] +Q "AD/A+Q,A/@2,DEST/AD,B/@1"
; 1598 [ ]_["] +RAM "AD/D+A,A/@2,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1"
; 1599 [ ]_["] +XR "AD/D+A,DBUS/RAM,RAMADR/XR#,A/@2,DEST/AD,B/@1,HOLD LEFT"
; 1600 [ ]_["] +["] "AD/A+B,A/@3,B/@1,B/@2,DEST/AD"
; 1601 [ ]_["] +["] +.25 "AD/A+B,A/@3,B/@1,B/@2,DEST/AD,ADD .25"
; 1602 [ ]_["] -# "AD/A-D-.25,DBUS/DBM,DBM/##,A/@2,DEST/AD,B/@1,ADD .25"
; 1603 [ ]_["] -1 "AD/B-A-.25,B/@1,A/ONE,DEST/AD,ADD .25"
; 1604 [ ]_["] -1000001 "AD/A-D-.25,A/@2,DBUS/DBM,DBM/##,##/1,DEST/AD,B/@1,ADD .25"
; 1605 [ ]_["] -AC "AD/A-D-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1606 [ ]_["] -RAM "AD/A-D-.25,A/@2,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1,ADD .25"
; 1607 [ ]_["] -["] "AD/B-A-.25,B/@1,B/@2,A/@3,DEST/AD,ADD .25"
; 1608 [ ]_["] -["] REV "AD/A-B-.25,B/@1,B/@3,A/@2,DEST/AD,ADD .25"
; 1609 [ ]_["] .AND.# "AD/D.AND.A,DBUS/DBM,DBM/##,DEST/AD,A/@2,B/@1"
; 1610 [ ]_["] .AND.# CLR LH "AD/ZERO,RSRC/DA,DBUS/DBM,DBM/##,DEST/AD,A/@2,B/@1"
; 1611 [ ]_["] .AND.# CLR RH "AD/D.AND.A,RSRC/OQ,DBUS/DBM,DBM/##,DEST/AD,A/@2,B/@1"
; 1612 [ ]_["] (AC[ ] .AND.[ ])*.5 "AD/D.AND.A,DEST/AD*.5,A/@3,B/@1,RAMADR/AC*#,DBUS/RAM,ACALU/AC+N,ACN/@2"
; 1613 [ ]_["] (Q+1)*.5 "AD/A+Q,A/ONE,DEST/AD*.5,B/@1"
; 1614 [ ]_["] (#-["])*2 "AD/D-A-.25,DEST/AD*2,A/@2,B/@1,DBUS/DBM,DBM/##,ADD .25"
; 1615 [ ]_["] (-["])*5 "AD/-A-.25,A/@2,DEST/AD*.5,B/@1,ADD .25"
; 1616 [ ]_["] (-["]-.25)*.5 LONG "AD/-A-.25,A/@2,DEST/Q_Q*.5,B/@1"
; 1617 [ ]_["] (-["]-.25)*2 LONG "AD/-A-.25,A/@2,DEST/Q_Q*2,B/@1"
; 1618
```



```
; 1664 .TOC "MACROS -- DATA PATH CHIP -- Q"
; 1665
; 1666 Q-[ ] "AD/Q-A-.25,A/@1,ADD .25"
; 1667 Q.AND.NOT.[ ] "AD/.NOT.A.AND.Q,A/@1,DEST/PASS"
; 1668 Q_[ ] "AD/A,DEST/Q_AD,A/@1"
; 1669 Q_[ ]-[ ] "AD/A-B-.25,A/@1,B/@2,DEST/Q_AD,ADD .25"
; 1670 Q_[ ]+[ ] "AD/A+B,A/@1,B/@2,DEST/Q_AD"
; 1671 Q_[ ]AND.[ ] "AD/A.AND.B,A/@1,B/@2,DEST/Q_AD"
; 1672 Q_.NOT.AC[ ] "AD/.NOT.D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,DT/3T"
; 1673 Q_-[ ] "AD/-A-.25,DEST/Q_AD,A/@1,ADD .25"
; 1674 Q_-1 "Q_-[ONE]"
; 1675 Q_-AC[ ] "AD/-D-.25,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,ADD .25,DT/3T"
; 1676 Q_-Q "AD/-Q-.25,ADD .25,DEST/Q_AD"
; 1677 Q_AC "AD/D,DBUS/RAM,RAMADR/AC#,DEST/Q_AD,CHK PARITY"
; 1678 Q_AC[ ] "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,CHK PARITY,DT/3T"
; 1679 Q_AC[ ]AND.MASK "AD/D.AND.A,A/MASK,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,CHK PARITY,DT/3T"
; 1680 Q_AC[ ]AND.[ ] "AD/D.AND.A,A/@2,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,CHK PARITY,DT/3T"
; 1681 Q_.NOT.Q "AD/.NOT.Q,DEST/Q_AD"
; 1682 Q_# "AD/D,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1683 Q_0 "AD/ZERO,DEST/Q_AD"
; 1684 Q_0 XWD [ ] "AD/47,DEST/Q_AD,DBM/#,DBUS/DBM,#/@1,RSRC/DA,A/MASK"
; 1685 Q_Q+.25 "AD/O+Q,DEST/Q_AD,ADD .25"
; 1686 Q_Q+1 "AD/A+Q,A/ONE,DEST/Q_AD"
; 1687 Q_Q-1 "AD/Q-A-.25,A/ONE,DEST/Q_AD,ADD .25"
; 1688 Q_Q+AC "AD/D+Q,DBUS/RAM,RAMADR/AC#,DEST/Q_AD"
; 1689 Q_Q*.5 "[MAG][MASK]*.5 LONG,SHSTYLE/NORM"
; 1690 Q_Q*2 "[MASK][MAG]*2 LONG,SHSTYLE/NORM"
; 1691 Q_Q.OR.# "AD/D.OR.Q,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1692 Q_Q.AND.# "AD/D.AND.Q,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1693 Q_Q.AND.[ ] "AD/A.AND.Q,A/@1,DEST/Q_AD"
; 1694 Q_Q.AND.NOT.[ ] "AD/.NOT.A.AND.Q,A/@1,DEST/Q_AD"
; 1695 Q_Q+[ ] "AD/A+Q,A/@1,DEST/Q_AD"
; 1696 Q_[ ]AND.Q "AD/A.AND.Q,A/@1,DEST/Q_AD"
; 1697 Q_[ ]OR.Q "AD/A.OR.Q,A/@1,DEST/Q_AD"
; 1698
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 45
; KS10.MIC[10,1141] 07:10 27-JULY-1984 MACROS -- DATA PATH CHIP -- MISC.

; 1699 .TOC "MACROS -- DATA PATH CHIP -- MISC."
; 1700
; 1701 CLEAR []O "AD/D.AND.A,A/@1,DBUS/DBM,DBM/#,#/377777,DEST/AD,B/@1,HOLD RIGHT"
; 1702 CLEAR ARXO "CLEAR [ARX]O"
; 1703
; 1704 ;CYCLE CHIP REGISTERS THRU AD SO WE CAN TEST BITS
; 1705 READ XR "AD/D,DBUS/RAM,RAMADR/XR#"
; 1706 READ [] "AD/B,B/@1"
; 1707 READ Q "AD/Q"
; 1708
; 1709 ;TEST BITS IN REGISTERS (SKIP IF ZERO)
; 1710 TR [] "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1,SKIP ADR.EQ.O,DT/3T"
; 1711 TL [] "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1,SKIP ADL.EQ.O,DT/3T"
; 1712
; 1713
; 1714 ;CAUSE BITS -2 AND -1 TO MATCH BIT 0.
; 1715 FIX [] SIGN "AD/D,DEST/A,A/@1,B/@1,DBUS/DP,HOLD RIGHT"
; 1716
; 1717 ;GENERATE A MASK IN Q AND ZERO A 2901 REGISTER
; 1718 GEN MSK [] "AD/ZERO,DEST/Q_Q*2,B/@1,ONES"
; 1719

; T1OKI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 46
MACROS -- STORE IN AC

```
; 1720 .TOC "MACROS -- STORE IN AC"
; 1721
; 1722 FM WRITE "FMWRITE/1"
; 1723
; 1724 AC[] [] VIA AD "AD/B,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,CHK PARITY"
; 1725 AC[] VIA AD "AD/B,DEST/PASS,B/@1,RAMADR/AC#,DBUS/DP,FM WRITE,CHK PARITY"
; 1726 AC[] [] "AD/A,DEST/A,B/@2,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1727 AC[] [] TEST "AD/D,DBUS/DP,DEST/A,B/@2,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1728 AC[] []+1 "AD/A+B,DEST/PASS,A/ONE,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1729 AC[] []*2 "AD/A+B,DEST/PASS,A/@2,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1730 AC[] "AD/A,DEST/A,B/@1,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1731 AC[] TEST "AD/D,DBUS/DP,DEST/A,B/@1,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1732 AC[]+1 "AD/A+B,DEST/PASS,A/ONE,B/@1,RAMADR/AC#,FM WRITE"
; 1733 AC[]+Q "AD/A+Q,DEST/PASS,A/@1,B/@1,RAMADR/AC#,FM WRITE"
; 1734 AC[] []+Q "AD/A+Q,DEST/PASS,A/@2,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,FM WRITE"
; 1735 AC[] []-[] "AD/A-B-.25,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1736 AC[] []+[] "AD/A+B,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1737 AC[] []+[] "AD/A+B,DEST/PASS,B/@2,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1738 AC[] [] .AND. [] "AD/A.AND.B,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1739 AC[] Q.AND. [] "AD/A.AND.Q,DEST/PASS,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1740 AC[] [] .EQV. Q "AD/A.EQV.Q,DEST/PASS,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1741 AC[] []-[] "AD/-B-.25,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1742 AC[] -[] "AD/-A-.25,DEST/PASS,A/@1,RAMADR/AC#,DBUS/DP,ADD .25,FM WRITE"
; 1743 AC[] [] .OR. [] "AD/A.OR.B,A/@1,B/@2,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1744 AC[] [] .NOT. [] "AD/.NOT.B,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1745 AC[] .NOT. [] "AD/.NOT.B,DEST/PASS,B/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1746 AC[] -Q "AD/-Q-.25,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1747 AC Q "AD/Q,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1748 AC[] O "AD/ZERO,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1749 AC[] 1 "AD/B,DEST/PASS,B/ONE,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1750 AC[] Q "AD/Q,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1751
```

```
; 1752 .TOC "MACROS -- MICROCODE WORK SPACE"
; 1753
; 1754
; 1755 WORK[]_Q "AD/Q,DEST/PASS, RAMADR/#,WORK/@1,FM WRITE"
; 1756 Q_WORK[] "AD/D,DEST/Q_AD, RAMADR/#,DBUS/RAM,WORK/@1,DT/3T"
; 1757 WORK[]_O "AD/ZERO,DEST/PASS, RAMADR/#,WORK/@1,FM WRITE"
; 1758 WORK[]_1 "AD/B,DEST/PASS, RAMADR/#,WORK/@1,B/ONE,FM WRITE"
; 1759 WORK[]_[] "AD/B,DEST/PASS, RAMADR/#,WORK/@1,B/@2,FM WRITE"
; 1760 WORK[]_[] CLR LH "AD/47,RSRC/AB,DEST/PASS, RAMADR/#,WORK/@1,B/@2,A/MASK,FM WRITE"
; 1761 WORK[]_[]-1 "AD/A-B-.25,A/@2,B/ONE,DEST/PASS, RAMADR/#,WORK/@1,FM WRITE, ADD .25"
; 1762 WORK[]_[] NOT.[] "AD/.NOT.B,DEST/PASS, RAMADR/#,WORK/@1,B/@2,FM WRITE"
; 1763 WORK[]_[] AND.[] "AD/A.AND.B,DEST/PASS, RAMADR/#,WORK/@1,A/@2,B/@3,FM WRITE"
; 1764 []_AND_NOT_WORK[] "AD/.NOT.D.AND.A,A/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1765 []_AND_WORK[] "AD/D.AND.A,A/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1766 []_[]+WORK[] "AD/D+A,A/@2,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@3,DT/3T"
; 1767 []_[] AND_WORK[] "AD/D.AND.A,A/@2,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@3,DT/3T"
; 1768 []_[] AND_NOT_WORK[] "AD/.NOT.D.AND.A,A/@2,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@3,DT/3T"
; 1769 []_[] OR_WORK[] "AD/D.OR.A,A/@2,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@3,DT/3T"
; 1770 []_WORK[] "AD/D,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1771 []_NOT_WORK[] "AD/.NOT.D,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1772 []_WORK[] "AD/-D-.25,ADD .25,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1773 []_WORK[]+1 "AD/D+A,A/ONE,DEST/AD,B/@1,DBUS/RAM, RAMADR/#,WORK/@2,DT/3T"
; 1774 Q_Q_WORK[] "AD/Q-D-.25,DEST/Q_AD,DBUS/RAM, RAMADR/#,WORK/@1,ADD .25,DT/3T"
; 1775 []_[]-WORK[] "AD/A-D-.25,DEST/AD,A/@2,B/@1,DBUS/RAM, RAMADR/#,WORK/@3,ADD .25,DT/3T"
; 1776
; 1777 RAM_[] "AD/B,DEST/PASS, RAMADR/RAM,B/@1,FM WRITE"
; 1778
```


; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 48
MACROS -- MEMORY CONTROL

```
; 1779 .TOC "MACROS -- MEMORY CONTROL"
; 1780
; 1781 MEM CYCLE "MEM/1"
; 1782
; 1783 ;THE FOLLOWING MACROS CONTROL MEMORY ADDRESS
; 1784 LOAD VMA "MEM CYCLE,LDVMA/1"
; 1785 FORCE EXEC "FORCE EXEC/1"
; 1786 VMA PHYSICAL "PHYSICAL/1,FORCE EXEC/1,FORCE USER/0,EXT ADR/1,LOAD VMA"
; 1787 VMA PHYSICAL WRITE "LOAD VMA,VMA PHYSICAL,WAIT/1,MEM/1,WRITE CYCLE/1,WRITE TEST/O"
; 1788 VMA PHYSICAL READ "LOAD VMA,VMA PHYSICAL,WAIT/1,MEM/1,READ CYCLE/1,WRITE TEST/O"
; 1789 VMA EXTENDED "EXT ADR/1"
; 1790
; 1791 PXCT EA "PXCT/E1"
; 1792 PXCT DATA "PXCT/D1"
; 1793 PXCT BLT DEST "PXCT/D1"
; 1794 PXCT BYTE PTR EA "PXCT/E2"
; 1795 PXCT BYTE DATA "PXCT/D2"
; 1796 PXCT STACK WORD "PXCT/D2"
; 1797 PXCT BLT SRC "PXCT/D2"
; 1798 PXCT EXTEND EA "PXCT/E2"
; 1799
; 1800 ;THE FOLLOWING MACROS GET MEMORY CYCLES STARTED
; 1801 WRITE TEST "WRITE TEST/1,WAIT/1"
; 1802 START READ "MEM CYCLE,READ CYCLE/1,WAIT/1"
; 1803 START WRITE "MEM CYCLE,WRITE TEST,WRITE CYCLE/1,WAIT/1"
; 1804 START NO TEST WRITE "MEM CYCLE,WRITE CYCLE/1,WAIT/1"
; 1805 FETCH "START READ,FETCH/1,PXCT/CURRENT,WAIT/1"
; 1806
; 1807 ;THE FOLLOWING MACROS COMPLETE MEMORY CYCLES
; 1808 MEM WAIT "MEM CYCLE,WAIT/1"
; 1809 MEM READ "MEM WAIT,DBUS/DBM,DBM/MEM"
; 1810 MEM WRITE "MEM WAIT,DT/3T"
; 1811 SPEC MEM READ "SPEC/WAIT,DBUS/DBM,DBM/MEM"
; 1812 SPEC MEM WRITE "SPEC/WAIT,DT/3T"
; 1813
; 1814
; 1815 ;THINGS WHICH WRITE MEMORY
; 1816 MEM_[] "AD/B,DEST/PASS,B/@1,DBUS/DP,RAMADR/VMA,CHK PARITY"
; 1817 MEM_Q "AD/Q,DBUS/DP,RAMADR/VMA"
; 1818
; 1819
; 1820 ;THINGS WHICH READ MEMORY
; 1821 []_IO DATA "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1"
; 1822 []_MEM "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1,CHK PARITY"
; 1823 []_MEM THEN FETCH "AD/D,DBUS/DBM,RAMADR/VMA,DEST/A,A/PC,B/@1,CHK PARITY, FETCH, LOAD VMA"
; 1824 []_MEM*.5 "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD*.5,B/@1,CHK PARITY"
; 1825 []_MEM.AND.MASK "AD/D.AND.A,A/MASK,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1,CHK PARITY"
; 1826 []_(MEM.AND.[])*.5 "AD/D.AND.A,A/@2,DBUS/DBM,RAMADR/VMA,DEST/AD*.5,B/@1,CHK PARITY"
; 1827 Q_MEM "AD/D,DBUS/DBM,RAMADR/VMA,DEST/Q_AD,CHK PARITY"
; 1828
```

```
; 1829 .TOC "MACROS -- VMA"  
; 1830  
; 1831 VMA_[] "AD/A,A/@1,DEST/PASS,LOAD VMA"  
; 1832 VMA_[] WITH FLAGS "AD/A,A/@1,DEST/PASS,LOAD VMA,WAIT/1, MEM/1, EXT ADR/1, DP FUNC/1, DT/3T"  
; 1833 VMA_[] .OR.[] WITH FLAGS "AD/A.OR.B,A/@1,B/@2,DEST/PASS,LOAD VMA,WAIT/1, MEM/1, EXT ADR/1, DP FUNC/1, DT/3T"  
; 1834 VMA_[]+1 "AD/A+B,A/ONE,B/@1,DEST/AD,HOLD LEFT,LOAD VMA"  
; 1835 VMA_[]-1 "AD/B-A-.25,A/ONE,B/@1,ADD .25,HOLD LEFT,LOAD VMA"  
; 1836 VMA_[]+XR "AD/D+A,DBUS/RAM,RAMADR/XR#,A/@1,LOAD VMA"  
; 1837 VMA_[]+[] "AD/A+B,DEST/PASS,A/@1,B/@2,LOAD VMA"  
; 1838  
; 1839 NEXT [] PHYSICAL WRITE "AD/A+B,A/ONE,B/@1,DEST/AD,HOLD LEFT,LOAD VMA, VMA PHYSICAL, START WRITE"  
; 1840  
; 1841 ;MACROS TO LOAD A 2901 REGISTER WITH VMA FLAG BITS  
; 1842 []_VMA FLAGS "AD/45,DEST/AD,B/@1,DBM/#,DBUS/DBM,RSRC/DO,A/MASK"  
; 1843 []_VMA IO READ "[@1]_VMA FLAGS,READ CYCLE/1,IO CYCLE/1,WRITE TEST/O, PHYSICAL/1, FORCE EXEC/1, FORCE USER/O"  
; 1844 []_VMA IO WRITE "[@1]_VMA FLAGS,WRITE CYCLE/1,IO CYCLE/1,WRITE TEST/O, PHYSICAL/1, FORCE EXEC/1, FORCE USER/O"  
; 1845
```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 50
MACROS -- TIME CONTROL

```
; 1846 .TOC      "MACROS -- TIME CONTROL"  
; 1847  
; 1848 AC        "RAMADR/AC#"  
; 1849 AC[]     "RAMADR/AC*#,ACALU/AC+N,ACN/@1"  
; 1850 XR       "RAMADR/XR#"  
; 1851 VMA      "RAMADR/VMA"  
; 1852 WORK[]   "RAMADR/#, WORK/@1"  
; 1853  
; 1854 2T       "T/2T"  
; 1855 3T       "T/3T"  
; 1856 4T       "T/4T"  
; 1857 5T       "T/5T"  
; 1858
```

```

: 1859 .TOC "MACROS -- SCAD, SC, FE LOGIC"
: 1860
: 1861 LOAD SC "LOADSC/1"
: 1862 LOAD FE "LOADFE/1"
: 1863 STEP SC "SCAD/A-1,SCADA/SC,LOAD SC,SKIP/SC"
: 1864 SHIFT "SCAD/A+B,SCADA/S#,SCADB/FE,S#/1, LOAD FE, MULTI SHIFT/1"
: 1865
: 1866 SC_SC-1 "SCAD/A-1,SCADA/SC,LOAD SC"
: 1867 SC_SHIFT "SCAD/A+B,SCADA/S#,S#/O,SCADB/SHIFT,LOAD SC"
: 1868 SC_SHIFT-1 "SCAD/A+B,SCADA/S#,S#/1777,SCADB/SHIFT,LOAD SC"
: 1869 SC_SHIFT-2 "SCAD/A+B,SCADA/S#,S#/1776,SCADB/SHIFT,LOAD SC"
: 1870 SC_-SHIFT "SCAD/A-B,SCADA/S#,S#/0000,SCADB/SHIFT,LOAD SC"
: 1871 SC_-SHIFT-1 "SCAD/A-B,SCADA/S#,SCADB/SHIFT,S#/1777,LOAD SC"
: 1872 SC_-SHIFT-2 "SCAD/A-B,SCADA/S#,SCADB/SHIFT,S#/1776,LOAD SC"
: 1873 SC_SC-EXP "SCAD/A-B,SCADA/SC,SCADB/EXP,LOAD SC"
: 1874 SC_SC-EXP-1 "SCAD/A-B-1,SCADA/SC,SCADB/EXP,LOAD SC"
: 1875 SC_SC-FE-1 "SCAD/A-B-1,SCADA/SC,SCADB/FE,LOAD SC"
: 1876 SC_SC-FE "SCAD/A-B,SCADA/SC,SCADB/FE,LOAD SC"
: 1877 SC_EXP "SCAD/A+B,SCADA/S#,S#/O,SCADB/EXP,LOAD SC"
: 1878 SC_S#-FE "SCAD/A-B,SCADA/S#,SCADB/FE,LOAD SC"
: 1879 SC_FE+S# "SCAD/A+B,SCADA/S#,SCADB/FE,LOAD SC"
: 1880 SC_FE "SCAD/A.OR.B,SCADA/S#,S#/O,SCADB/FE,LOAD SC"
: 1881 SC_S# "SCAD/A,SCADA/S#,LOAD SC"
: 1882
: 1883
: 1884 SC_36. "SC_S#,S#/36."
: 1885 SC_35. "SC_S#,S#/35."
: 1886 SC_34. "SC_S#,S#/34."
: 1887 SC_28. "SC_S#,S#/28."
: 1888 SC_27. "SC_S#,S#/27."
: 1889 SC_26. "SC_S#,S#/26."
: 1890 SC_24. "SC_S#,S#/24."
: 1891 SC_22. "SC_S#,S#/22."
: 1892 SC_20. "SC_S#,S#/20."
: 1893 SC_19. "SC_S#,S#/19."
: 1894 SC_14. "SC_S#,S#/14."
: 1895 SC_11. "SC_S#,S#/11."
: 1896 SC_9. "SC_S#,S#/9."
: 1897 SC_8. "SC_S#,S#/8."
: 1898 SC_7. "SC_S#,S#/7."
: 1899 SC_6. "SC_S#,S#/6."
: 1900 SC_5. "SC_S#,S#/5."
: 1901 SC_4. "SC_S#,S#/4."
: 1902 SC_3. "SC_S#,S#/3."
: 1903 SC_2. "SC_S#,S#/2."
: 1904 SC_1. "SC_S#,S#/1."
: 1905 SC_0. "SC_S#,S#/O."
: 1906 SC_-1. "SC_S#,S#/1777"
: 1907 SC_-2. "SC_S#,S#/1776"
: 1908

```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 52
MACROS -- SCAD, SC, FE LOGIC

```
; 1909 FE_-FE "SCAD/A-B,SCADA/S#,S#/O,SCADB/FE,LOAD FE"
; 1910 FE_-FE-1 "SCAD/A-B,SCADA/S#,S#/1777,SCADB/FE,LOAD FE"
; 1911 FE_-FE-19 "SCAD/A+B,SCADB/FE,SCADA/S#,S#/1550,LOAD FE"
; 1912 FE_-FE+S# "SCAD/A-B,SCADA/S#,SCADB/FE,LOAD FE"
; 1913 FE_-FE+SC "SCAD/A+B,SCADA/SC,SCADB/FE,LOAD FE"
; 1914 FE_-FE.AND.S# "SCAD/A.AND.B,SCADA/S#,SCADB/FE,LOAD FE"
; 1915 FE_-P "SCAD/A,SCADA/BYTE1,LOAD FE"
; 1916 FE_-S "SCAD/A+B,SCADA/S#,S#/O,SCADB/SIZE,LOAD FE"
; 1917 FE_-S+2 "SCAD/A+B,SCADA/S#,S#/20,SCADB/SIZE,LOAD FE"
; 1918 FE_-S-20 "SCAD/A-B,SCADA/S#,S#/1760,SCADB/SIZE,LOAD FE"
; 1919 FE_-S-10 "SCAD/A-B,SCADA/S#,S#/1770,SCADB/SIZE,LOAD FE"
; 1920 FE_-S# "SCAD/A,SCADA/S#,LOAD FE"
; 1921 FE_-S#-FE "SCAD/A-B,SCADA/S#,SCADB/FE,LOAD FE"
; 1922 FE_-2 "FE_S#,S#/1776"
; 1923 FE_-12. "FE_S#,S#/1764"
; 1924 FE_-O "FE_S#,S#/O"
; 1925 FE_-1 "FE_S#,S#/1777"
; 1926 FE_-FE+1 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/1,LOAD FE"
; 1927 FE_-FE+2 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/2,LOAD FE"
; 1928 FE_-FE+10 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/10,LOAD FE"
; 1929 FE_-FE-1 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/1777,LOAD FE"
; 1930 FE_-FE+4 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/4,LOAD FE"
; 1931 FE_-EXP "SCAD/A+B,SCADA/S#,S#/O,SCADB/EXP,LOAD FE"
; 1932 FE_-SC+EXP "SCAD/A+B,SCADA/SC,SCADB/EXP,LOAD FE"
; 1933 FE_-SC-EXP "SCAD/A-B,SCADA/SC,SCADB/EXP,LOAD FE"
; 1934 FE_-FE+P "SCAD/A+B,SCADA/BYTE1,SCADB/FE,LOAD FE"
; 1935 FE_-FE-200 "SCAD/A+B,SCADA/S#,S#/1600,SCADB/FE,LOAD FE"
; 1936 FE_-FE+200 "SCAD/A-B,SCADA/S#,S#/200,SCADB/FE,LOAD FE"
; 1937 FE_-FE+S# "SCAD/A+B,SCADA/S#,SCADB/FE,LOAD FE"
; 1938
; 1939
; 1940 GEN 17-FE "SCAD/A-B,SCADA/S#,S#/210,SCADB/FE"
; 1941
```

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 53
MACROS -- DATA PATH FIELD CONTROL

; 1942 .TOC "MACROS -- DATA PATH FIELD CONTROL"
; 1943
; 1944 HOLD LEFT "CLKL/O,GENL/O"
; 1945 ADL PARITY "GENL/1"
; 1946 CHK PARITY L "CHKL/1"
; 1947
; 1948 HOLD RIGHT "CLKR/O,GENR/O"
; 1949 ADR PARITY "GENR/1"
; 1950 CHK PARITY R "CHKR/1"
; 1951
; 1952 AD PARITY "AD PARITY OK/1"
; 1953 CHK PARITY "CHKL/1,CHKR/1"
; 1954 BAD PARITY "CHKL/O,CHKR/O"
; 1955
; 1956 INH CRY18 "SPEC/INHCRY18"
; 1957

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 54
MACROS -- SHIFT PATH CONTROL

```
; 1958 .TOC      "MACROS -- SHIFT PATH CONTROL"
; 1959
; 1960 ASH      "SHSTYLE/NORM" ;ASH SHIFT
; 1961 LSH      "SHSTYLE/NORM" ;LSH SHIFT (SAME HARDWARE AS ASH BUT
; 1962          ; BITS -2 AND -1 ARE PRESET TO ZERO)
; 1963 ROT      "SHSTYLE/ROT"
; 1964 LSHC     "SHSTYLE/LSHC"
; 1965 ASHC     "SHSTYLE/ASHC"
; 1966 ROTC     "SHSTYLE/ROTC"
; 1967 ONES     "SHSTYLE/ONES" ;SHIFT IN 1 BITS
; 1968 DIV      "SHSTYLE/DIV"  ;SPECIAL PATH FOR DIVIDE (LIKE ROTC BUT
; 1969          ; COMPLEMENT BIT AS IT GOES AROUND)
; 1970
```

```
; 1971 .TOC "MACROS -- SPECIAL FUNCTIONS"
; 1972
; 1973 LOAD IR "SPEC/LOADIR" ;LOAD INSTRUCTION REG FROM
; 1974 ; DBUS0-DBUS8, LOAD AC# FROM
; 1975 ; DBUS9-DBUS12
; 1976 ; UPDATE LAST-INST-PUBLIC PC FLAG
; 1977 LOAD INST "SPEC/LDINST"
; 1978 LOAD INST EA "SPEC/LOADXR,PXCT/CURRENT"
; 1979 LOAD BYTE EA "SPEC/LOADXR,PXCT/E2"
; 1980 LOAD IND EA "SPEC/LOADXR,PXCT/E1"
; 1981 LOAD SRC EA "SPEC/LOADXR,PXCT/BIS-SRC-EA"
; 1982 LOAD DST EA "SPEC/LOADXR,PXCT/BIS-DST-EA"
; 1983 ADD .25 "CRY38/1" ;GENERATE CARRY IN TO BIT 37
; 1984 CALL [] "CALL/1,J/@1" ;CALL A SUBROUTINE
; 1985 LOAD PXCT "SPEC/LDPXCT" ;LOAD PXCT FLAGS IF EXEC MODE
; 1986 TURN OFF PXCT "SPEC/PXCT OFF"
; 1987 LOAD PAGE TABLE "SPEC/LDPAGE"
; 1988 LOAD AC BLOCKS "SPEC/LDACBLK"
; 1989 SWEEP "SPEC/SWEEP,PHYSICAL/1"
; 1990 CLRCSH "SPEC/CLRCSH,PHYSICAL/1"
; 1991 LOAD PI "SPEC/LDPI"
; 1992 SET HALT "SPEC/#,#/74"
; 1993 CLEAR CONTINUE "SPEC/#,#/40"
; 1994 CLEAR EXECUTE "SPEC/#,#/20"
; 1995 CLEAR RUN "SPEC/#,#/10"
; 1996 UNHALT "SPEC/#,#/62"
; 1997 SET APR ENABLES "SPEC/APR EN"
; 1998 ABORT MEM CYCLE "DBUS/DBM,RAMADR/VMA,DBM/MEM,AD/ZERO,SPEC/MEMCLR,LOAD VMA"
; 1999 CLR IO BUSY "SPEC/CLR IO BUSY"
; 2000 CLR IO LATCH "SPEC/CLR IO LATCH"
; 2001
```


; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 56
MACROS -- PC FLAGS

```
; 2002 .TOC "MACROS -- PC FLAGS"
; 2003
; 2004 CHANGE FLAGS "SPEC/FLAGS"
; 2005
; 2006 SET AROV "CHANGE FLAGS, HOLD USER/1, SETOV/1, TRAP1/1"
; 2007 SET FOV "CHANGE FLAGS, HOLD USER/1, SETFOV/1, TRAP1/1"
; 2008 SET NO DIVIDE "CHANGE FLAGS, HOLD USER/1, SETOV/1, SETNDV/1, TRAP1/1"
; 2009 SET FL NO DIVIDE "SET NO DIVIDE, SETFOV/1"
; 2010
; 2011 ASH AROV "SPEC/ASHOV"
; 2012 SET FPD "CHANGE FLAGS, HOLD USER/1, SETFPD/1"
; 2013 CLR FPD "CHANGE FLAGS, HOLD USER/1, CLRFPD/1"
; 2014
; 2015 SET PDL OV "CHANGE FLAGS, HOLD USER/1, TRAP2/1"
; 2016 SET TRAP1 "CHANGE FLAGS, HOLD USER/1, TRAP1/1"
; 2017
; 2018 LOAD PCU "CHANGE FLAGS, LD PCU/1"
; 2019 UPDATE USER "CHANGE FLAGS, HOLD USER/1"
; 2020 LEAVE USER "CHANGE FLAGS, HOLD USER/O"
; 2021
; 2022 JFCL FLAGS "CHANGE FLAGS, HOLD USER/1, JFCLFLG/1"
; 2023
; 2024 LOAD FLAGS "CHANGE FLAGS, LD FLAGS/1"
; 2025 EXP TEST "SPEC/EXPTST"
; 2026 AD FLAGS "CHANGE FLAGS, ADFLGS/1, HOLD USER/1"
; 2027
; 2028 NO DIVIDE "SET NO DIVIDE, J/NIDISP"
; 2029 FL NO DIVIDE "SET FL NO DIVIDE, J/NIDISP"
; 2030
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 57
; KS10.MIC[10,1141] 07:10 27-JULY-1984 MACROS -- PAGE FAIL FLAGS

; 2031 .TOC "MACROS -- PAGE FAIL FLAGS"
; 2032
; 2033 STATE_[] "[FLG]_#, STATE/@1, HOLD LEFT"
; 2034 END STATE "[FLG]_O, HOLD LEFT"
; 2035
; 2036 END BLT "END STATE"
; 2037 END MAP "END STATE"
; 2038

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 58
MACROS --- SINGLE SKIPS

```
; 2039 .TOC "MACROS -- SINGLE SKIPS"
; 2040 ;SKIPS IF:
; 2041 SKIP IF ACO "SKIP/ACO" ;THE AC NUMBER IS ZERO
; 2042 SKIP DPO "SKIP/DPO" ;DP BIT 0=1
; 2043 SKIP DP18 "SKIP/DP18" ;DP BIT 18=1
; 2044 SKIP AD.EQ.O "SKIP/ADEQO,DT/3T" ;ADDER OUTPUT IS ZERO
; 2045 SKIP AD.LE.O "SKIP/LE,DT/3T" ;ADDER OUTPUT IS LESS THAN OR EQUAL
; 2046 ; TO ZERO.
; 2047 SKIP ADL.LE.O "SKIP/LLE,DT/3T" ;ADDER LEFT IS LESS THAN OR EQUAL
; 2048 ; TO ZERO.
; 2049 SKIP FPD "SKIP/FPD" ;FIRST-PART-DONE PC FLAG IS SET
; 2050 SKIP KERNEL "SKIP/KERNEL" ;USER=0
; 2051 SKIP IO LEGAL "SKIP/IOLGL" ;USER=0 OR USER IOT=1
; 2052 SKIP CRYO "SKIP/CRYO" ;ADDER BIT CRYO=1 (NOT PC FLAG BIT)
; 2053 SKIP CRY1 "SKIP/CRY1" ;ADDER BIT CRY1=1 (NOT PC FLAG BIT)
; 2054 SKIP CRY2 "SKIP/CRY2,DT/3T" ;ADDER BIT CRY2=1
; 2055 SKIP JFCL "SKIP/JFCL" ;IF JFCL SHOULD JUMP
; 2056 SKIP ADL.EQ.O "SKIP/ADLEQO" ;ALU BITS -2 TO 17 = 0
; 2057 SKIP ADR.EQ.O "SKIP/ADREQO" ;ALU BITS 18-35 = 0
; 2058 SKIP IRPT "SKIP/INT" ;INTERRUPT IS PENDING
; 2059 SKIP -1MS "SKIP/-1 MS" ;DON'T SKIP IF 1MS TIMER HAS EXPIRED.
; 2060 SKIP AC REF "SKIP/ACREF" ;VMA IS 0-17
; 2061 SKIP EXECUTE "SKIP/EXECUTE" ;CONSOLE EXECUTE
; 2062 TXXX TEST "SKIP/TXXX" ;TEST INSTRUCTION SHOULD SKIP
; 2063
```

; 2064 .TOC "MACROS -- SPECIAL DISPATCH MACROS"
; 2065
; 2066 NEXT INST "DISP/NICOND,SPEC/NICOND,J/NICOND"
; 2067 NEXT INST FETCH "DISP/NICOND,SPEC/NICOND,J/NICOND-FETCH"
; 2068 EA MODE DISP "DISP/EAMODE, RAMADR/XR#"
; 2069 AREAD "DISP/AREAD, WAIT/1, AREAD/1, MEM/1, J/O"
; 2070 B DISP "DISP/BDISP"
; 2071 BWRITE DISP "B DISP, MEM/1, BWRITE/1, WRITE CYCLE/1, J/BWRITE"
; 2072 INST DISP "DISP/DROM, J/O"
; 2073 EXIT "BWRITE DISP, SPEC/O, WRITE TEST/1"
; 2074 AD FLAGS EXIT "BWRITE DISP, WRITE TEST/O, AD FLAGS"
; 2075 FL-EXIT "WRITE CYCLE/1, WRITE TEST/1, MEM/1, BWRITE/1, B DISP, J/FL-BWRITE"
; 2076 TEST DISP "B DISP, J/TEST-TABLE"
; 2077 SKIP-COMP DISP "B DISP, J/SKIP-COMP-TABLE"
; 2078 JUMP DISP "B DISP, J/JUMP-TABLE"
; 2079 DONE "VMA [PC], LOAD VMA, FETCH, NEXT INST FETCH"
; 2080 JUMPA "[PC]_[AR], HOLD LEFT, LOAD VMA, FETCH, NEXT INST FETCH"
; 2081 UUU "[HR]_[HR].AND.#, #/777740, HOLD RIGHT, J/UUOGO"
; 2082 LUUU "[AR]_O XWD [40], J/LUUU1"
; 2083 PAGE FAIL TRAP "TL [FLG], FLG.PI/1, J/PFT"
; 2084 TAKE INTERRUPT "[FLG]_[FLG].OR.#, FLG.PI/1, HOLD RIGHT, J/PI"
; 2085 INTERRUPT TRAP "WORK[SV.AR]_[AR], J/ITRAP"
; 2086 MUL DISP "DISP/MUL"
; 2087 DIV DISP "DISP/DIV"
; 2088 BYTE DISP "DISP/BYTE, DT/3T"
; 2089 SCAD DISP "DISP/SCADO" ; SKIP (2'S WEIGHT) IS SCAD IS MINUS
; 2090 RETURN [] "DISP/RETURN, J/@1"
; 2091 PI DISP "DISP/PI"
; 2092 NORM DISP "DISP/NORM, DT/3T"
; 2093 DISMISS "TR [PI], #/077400, CALL [JEN1], DT/3T"
; 2094 CALL LOAD PI "[TO] [PI] SWAP, CALL [LDPI2]"
; 2095 HALT [] "AD/47, DEST/AD, B/T1, DBM/#, DBUS/DBM, HALT/@1, RSRC/DA, A/MASK, J/HALTED"
; 2096 CLEANUP DISP "READ [FLG], DBUS/DP, DISP/DP, 3T, J/CLEANUP"
; 2097

; T10KI.MCR[10,1141]
; KS10.MIC[10,1141]

11:45 11-AUG-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 60
DISPATCH ROM MACROS

```
; 2098 .TOC "DISPATCH ROM MACROS"
; 2099 .DCODE
; 2100
; 2101 ;"A FIELD" MACROS SAY HOW TO FETCH ARGUMENTS
; 2102
; 2103 I "I/1"
; 2104 I-PF "I/1,VMA/O,READ/1"
; 2105 R "A/READ,READ/1"
; 2106 R-PF "A/RD-PF,READ/1"
; 2107 W "A/WRITE,TEST/1"
; 2108 RW "A/READ,TEST/1,READ/1"
; 2109 IW "I/1,TEST/1" ;IMMED WHICH STORE IN E. (SETZM, ETC.)
; 2110 IR "I/1,READ/1" ;START READ A GO TO EXECUTE CODE
; 2111 DBL R "A/DREAD,READ/1" ;AR!ARX _ E!E+1
; 2112 DBL AC "A/DBLAC"
; 2113 SH "A/SHIFT,VMA/O,READ/1"
; 2114 SHC "A/DSHIFT,VMA/O,READ/1"
; 2115 FL-R "A/FP,READ/1" ;FLOATING POINT READ
; 2116 FL-RW "A/FP,READ/1,TEST/1"
; 2117 FL-I "A/FPI,READ/O" ;FLOATING POINT IMMEDIATE
; 2118 DBL FL-R "A/DFP,READ/1"
; 2119 IOT "A/IOT" ;CHECK FOR IO LEGAL
; 2120
; 2121 ;"B FIELD" MACROS SAY HOW TO STORE RESULTS
; 2122
; 2123 AC "B/AC"
; 2124 M "B/MEM,TEST/1,COND FUNC/1"
; 2125 B "B/BOTH,TEST/1,COND FUNC/1"
; 2126 S "B/SELF,TEST/1,COND FUNC/1"
; 2127 DAC "B/DBLAC"
; 2128 DBL B "B/DBLB,TEST/1,COND FUNC/1"
; 2129 FL-AC "FL-B/AC" ;FLOATING POINT
; 2130 FL-MEM "FL-B/MEM,TEST/1,COND FUNC/1" ;FLOATING POINT TO MEMORY
; 2131 FL-BOTH "FL-B/BOTH,TEST/1,COND FUNC/1" ;FLOATING POINT TO BOTH
; 2132 ROUND "ROUND/1" ;FLOATING POINT ROUNDED
; 2133
; 2134
; 2135 ;CONTROL BITS
; 2136 W TEST "TEST/1"
; 2137 AC DISP "ACDISP/1"
; 2138 .UCODE
; 2139
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 61
DISPATCH ROM MACROS

```
; 2140 .BIN
; 2141 .TOC "POWER UP SEQUENCE"
; 2142
; 2143 .UCODE
; 2144
; 2145 ;HERE IS WHERE WE FIRE THE MACHINE UP DURING POWER ON
; 2146
; 2147
; 2148 O: [MASK]_#, #/377777 ;BUILD A MASK WITH
; 2149 [MASK]_[MASK]*2 ; A ONE IN 36-BITS AND O
; 2150 [MASK]_[MASK].OR.#,#/1 ; IN BITS -2,-1,36,37
; 2151 [MAG]_[MASK]*.5 ;MAKE CONSTANT
; 2152 [XWD1]_#, #/1 ;CONSTANT WITH 1 IN EACH
; 2153 ; HALF WORD
; 2154 [ONE]_O XWD [1], ;THE CONSTANT 1
; 2155 CALL/1 ;RESET STACK (CAN NEVER RETURN
; 2156 ; TO WHERE MR LEFT US)
; 2157 3: [AR]_O XWD [376000] ;ADDRESS OF HALT STATUS
; 2158 ; BLOCK
; 2159 WORK[HSBADR]_[AR] ;SAVE FOR HALT LOOP
; 2160 [UBR]_O, ABORT MEM CYCLE ;CLEAR THE UBR AND RESET
; 2161 ; MEMORY CONTROL LOGIC
; 2162 [EBR]_O, LOAD AC BLOCKS ;CLEAR THE EBR AND FORCE
; 2163 ; PREVIOUS AND CURRENT AC
; 2164 ; BLOCKS TO ZERO
; 2165 [FLG]_O, SET APR ENABLES ;CLEAR THE STATUS FLAGS AND
; 2166 ; DISABLE ALL APR CONDITIONS
; 2167 WORK[APR]_[FLG] ;ZERO REMEMBERED ENABLES
; 2168
; 2169 WORK[TIMEO]_[FLG] ;CLEAR TIME BASE
; 2170 WORK[TIME1]_[FLG] ; ..
; 2171 .IF/FULL
; 2172 AC[BINO]_O ;COMPUTE A TABLE OF POWERS OF
; 2173 AC[BIN1]_1 ; TEN
; 2174 [AR]_O, SC 19. ;WE WANT TO GET 22 NUMBERS
; 2175 WORK[DECLO]_1 ;STARTING WITH 1
; 2176 WORK[DECHI]_O ; ..
; 2177 [HR]_#, WORK/DECLO ;ADDRESS OF LOW WORD
; 2178 [BRX]_#, WORK/DECHI ;ADDRESS OF HIGH WORD
; 2179 TENLP: [BRX]_[BRX]+1, LOAD VMA ;ADDRESS THE HIGH WORD
; 2180 =O* [ARX]_AC[BIN1], ;LOW WORD TO ARX
; 2181 CALL [DBSLOW] ;MULTIPLY BY TEN
; 2182 RAM [BR] ;SAVE HIGH WORD
; 2183 [HR]_[HR]+1, LOAD VMA ;WHERE TO STORE LOW WORD
; 2184 RAM_[ARX], STEP SC ;STORE LOW WORD AND SEE IF
; 2185 ; WE ARE DONE
; 2186 =O J/TENLP ;NOT YET--KEEP GOING
; 2187 [BR].XOR.#, 3T, SKIP ADL.EQ.O, #/330656
; 2188 ;DID WE GET THE RIGHT ANSWER
; 2189 ; IN THE TOP 18 BITS?
; 2190 =O**0 HALT [MULERR] ;NO--CPU IS BROKEN
; 2191 .ENDIF/FULL
; 2192
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 62
POWER UP SEQUENCE

```
U 0141, 3507,4221,0014,4174,4007,0700,0010,0000,0000 ; 2193 =0**1 [PI]_O, CALL [LOADPI] ;CLEAR PI STATE
; 2194 =1**1 ;CLEAR REGISTERS SO NO
; 2195 ;PARITY ERROR HAPPEN
;2196 .IFNOT/FULL
;2197 [ARX]_O ;WRITTEN WHILE COMPUTING POWERS
;2198 [BR]_O ;OF 10
;2199 [BRX]_O
; 2200 .ENDIF/FULL
U 0151, 0323,4751,1217,4374,4007,0700,0000,0000,0120 ; 2201 [T1]_O XWD [120] ;RH OF 120 CONTAINS START ADDRESS
; 2202 ; FOR SIMULATOR. FOR THE REAL
; 2203 ; MACHINE IT IS JUST DATA WITH
; 2204 ; GOOD PARITY.
; 2205 =
; 2206 ;THE CODE UNDER .IF/SIM MUST USE THE SAME ADDRESS AS THE CODE
; 2207 ; UNDER .IFNOT/SIM SO THAT MICROCODE ADDRESSES DO NOT CHANGE BETWEEN
; 2208 ; VERSIONS
;2209 .IF/SIM
;2210 VMA_[T1], START READ ;READ THE WORD
;2211 MEM_READ, [PC]_MEM, HOLD LEFT, J/START
;2212 ;GO FIRE UP SIMULATOR AT THE
;2213 ; PROGRAMS STARTING ADDRESS
; 2214 .IFNOT/SIM
; 2215 [PC]_O, ;CLEAR LH:OF PC
; 2216 LEAVE USER, ;ENTER EXEC MODE
; 2217 LOAD FLAGS ;CLEAR TRAP FLAGS
; 2218 [T1] #, HALT/POWER, ;LOAD T1 WITH POWER UP CODE
; 2219 J/PWRON ;ENTER HALT LOOP. DO NOT STORE
; 2220 ; HALT STATUS BLOCK
; 2221 .ENDIF/SIM
; 2222
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 63
THE INSTRUCTION LOOP -- START NEXT INSTRUCTION

```
; 2223 .TOC "THE INSTRUCTION LOOP -- START NEXT INSTRUCTION"
; 2224
; 2225 ;ALL INSTRUCTIONS EXCEPT JUMP'S AND UUD'S END UP HERE
; 2226 1400:
U 1400, 0110,3443,0100,4174,4156,4700,0200,0014,0012 ; 2227 DONE: DONE
U 1401, 0110,0111,0701,4170,4156,4700,0200,0014,0012 ; 2228 1401: VMA_[PC]+1, NEXT INST FETCH, FETCH
; 2229 =0
U 0372, 0110,0111,0701,4170,4156,4700,0200,0014,0012 ; 2230 SKIP: VMA_[PC]+1, NEXT INST 'FETCH, FETCH
U 0373, 0110,3443,0100,4174,4156,4700,0200,0014,0012 ; 2231 DONE
; 2232
; 2233
; 2234 ;16-WAY DISPATCH BASED ON NEXT INSTRUCTION
; 2235 =0000
; 2236 NICOND:
; 2237 =0001 [AR]_O XWD [423], ;TRAP TYPE 3
; 2238 ; GET ADDRESS OF TRAP INST
; 2239 TURN OFF PXCT, ;CLEAR PXCT
U 0101, 3377,4751,1203,4374,4367,0700,0000,0000,0423 ; 2240 J/TRAP ;PROCESS TRAP (INOUT.MIC)
; 2241 =0010 [AR]_O XWD [422], ;TRAP TYPE 2
; 2242 TURN OFF PXCT, ;CLEAR PXCT
U 0102, 3377,4751,1203,4374,4367,0700,0000,0000,0422 ; 2243 J/TRAP ;GO TRAP
; 2244 =0011 [AR]_O XWD [421], ;TRAP TYPE 1
; 2245 TURN OFF PXCT, ;TURN OF PXCT
U 0103, 3377,4751,1203,4374,4367,0700,0000,0000,0421 ; 2246 J/TRAP ;GO TRAP
U 0105, 0104,4751,1217,4374,4007,0700,0000,0000,0002 ; 2247 =0101 HALT [CSL] ;"HA" COMMAND TO 8080
; 2248 =0111
; 2249 START: VMA_[PC], ;LOAD VMA
; 2250 FETCH, ;INDICATE INSTRUCTION FETCH
U 0107, 0117,3443,0100,4174,4007,0700,0200,0014,0012 ; 2251 J/XCTGO ;GO GET INSTRUCTION
; 2252 ;THE NEXT SET OF CASES ARE USED WHEN THERE IS A FETCH
; 2253 ; IN PROGRESS
; 2254 =1000
; 2255 NICOND-FETCH:
; 2256 =1001 [AR]_O XWD [423], ;TRAP TYPE 3
; 2257 TURN OFF PXCT,
; 2258 J/TRAP
U 0111, 3377,4751,1203,4374,4367,0700,0000,0000,0423 ; 2259 =1010 [AR]_O XWD [422], ;TRAP TYPE 2
; 2260 TURN OFF PXCT,
; 2261 J/TRAP
U 0112, 3377,4751,1203,4374,4367,0700,0000,0000,0422 ; 2262 =1011 [AR]_O XWD [421], ;TRAP TYPE 1
; 2263 TURN OFF PXCT,
; 2264 J/TRAP
U 0113, 3377,4751,1203,4374,4367,0700,0000,0000,0421 ; 2265 =1101 HALT [CSL] ;"HA" COMMAND TO 8080
U 0115, 0104,4751,1217,4374,4007,0700,0000,0000,0002 ; 2266 =1111
; 2267 XCTGO: MEM READ, ;WAIT FOR MEMORY
; 2268 [HR]_MEM, ;PUT DATA IN HR
; 2269 LOAD INST, ;LOAD IR & AC #
U 0117, 0334,3771,0002,4365,5617,0700,0200,0000,0002 ; 2270 J/INCP C ;GO BUMP PC
; 2271 =
; 2272
```


; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 64
THE INSTRUCTION LOOP -- START NEXT INSTRUCTION

```
U 0334, 0201,0111,0701,2170,4366,6700,0200,0010,0010
; 2273 ;HERE WE POINT PC TO NEXT INSTRUCTION WHILE WE WAIT FOR
; 2274 ; EFFECTIVE ADDRESS LOGIC TO SETTLE
; 2275 INPC: VMA_[PC]+1, ;ADDRESS OF NEXT INSTRUCTION
; 2276 FETCH/1, ;INSTRUCTION FETCH
; 2277 TURN OFF PXCT, ;CLEAR EFFECT OF PXCT
; 2278 EA MODE DISP ;DISPACTH OF INDEXING AND @
; 2279
; 2280 ;MAIN EFFECTIVE ADDRESS CALCULATION
; 2281 =0001
; 2282 EACALC:
; 2283 ;
; 2284 ; THE FIRST 4 CASES ARE USED ONLY FOR JRST
; 2285 ;
; 2286
; 2287 ;CASE 0 -- JRST O,FOO(XR)
; 2288 [PC]_[HR]+XR, ;UPDATE PC
; 2289 HOLD LEFT, ;ONLY RH
; 2290 LOAD VMA, FETCH, ;START GETTING IT
; 2291 NEXT INST FETCH ;START NEXT INST
; 2292
; 2293 ;CASE 2 -- JRST O,FOO
; 2294 [PC]_[HR], ;NEW PC
; 2295 HOLD LEFT, ;ONLY RH
; 2296 LOAD VMA, FETCH, ;START GETTING IT
; 2297 NEXT INST FETCH ;START NEXT INST
; 2298
; 2299 ;CASE 4 -- JRST O,@FOO(XR)
; 2300 [HR]_[HR]+XR, ;ADD IN INDEX
; 2301 START READ, ;START TO FETCH @ WORD
; 2302 LOAD VMA, ;PUT ADDRESS IN VMA
; 2303 J/FETIND ;GO DO MEM WAIT (FORGET ABOUT JRST)
; 2304
; 2305 ;CASE 6 -- JRST O,@FOO
; 2306 VMA_[HR], ;LOAD UP ADDRESS
; 2307 START READ, ;START TO FETCH @ WORD
; 2308 J/FETIND ;GO DO MEM WAIT (FORGET ABOUT JRST)
; 2309

U 0201, 0110,0551,0201,2270,4156,4700,0200,0014,0012

U 0203, 0110,3441,0201,4170,4156,4700,0200,0014,0012

U 0205, 0343,0551,0202,2270,4007,0700,0200,0004,0012

U 0207, 0343,3443,0200,4174,4007,0700,0200,0004,0012
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 65
THE INSTRUCTION LOOP -- START NEXT INSTRUCTION

```
U 0211, 0213,0551,0202,2270,4007,0700,0000,0000,0000
; 2310 ;
; 2311 ;THESE 4 ARE FOR THE NON-JRST CASE
; 2312 ;
; 2313 ;
; 2314 ;CASE 10 -- JUST INDEXING
; 2315 INDEX: [HR]_[HR]+XR, ;ADD IN INDEX REGISTER
; 2316 HOLD LEFT ;JUST DO RIGHT HALF
; 2317 ;
; 2318 ;CASE 12 -- NO INDEXING OR INDIRECT
; 2319 NOMOD: [AR]_EA, ;PUT O..E IN AR
; 2320 PXCT DATA, AREAD ;DO ONE OR MORE OF THE FOLLWING
; 2321 ; ACCORDING TO THE DROM:
; 2322 ;1. LOAD VMA
; 2323 ;2. START READ OR WRITE
; 2324 ;3. DISPATCH TO 40-57
; 2325 ; OR DIRECTLY TO EXECUTE CODE
; 2326 ;
; 2327 ;CASE 14 -- BOTH INDEXING AND INDIRECT
; 2328 BOTH: [HR]_[HR]+XR, ;ADD IN INDEX REGISTER
; 2329 LOAD VMA, PXCT EA, ;PUT ADDRESS IN VMA
; 2330 START READ, J/FETIND ;START CYCLE AND GO WAIT FOR DATA
; 2331 ;
; 2332 ;CASE 16 -- JUST INDIRECT
; 2333 INDRCT: VMA_[HR], ;LOAD ADDRESS OF @ WORD
; 2334 START READ, PXCT EA ;START CYCLE
; 2335 ;
; 2336 ;
; 2337 ;HERE TO FETCH INDIRECT WORD
; 2338 FETIND: MEM READ, [HR]_MEM, ;GET DATA WORD
; 2339 HOLD LEFT, ;JUST RIGHT HALF
; 2340 LOAD IND EA ;RELOAD @ AND INDEX FLOPS
; 2341 ;
; 2342 XCT2: VMA_[PC], ;PUT PC BACK IN VMA
; 2343 FETCH/1, ;TURN ON FETCH FLAG
; 2344 EA MODE DISP, ;REDO CALCULATION FOR
; 2345 J/EACALC ; NEW WORD
; 2346 ;
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 66
THE INSTRUCTION LOOP -- FETCH ARGUMENTS

```
; 2347 .TOC "THE INSTRUCTION LOOP -- FETCH ARGUMENTS"
; 2348 ;HERE ON AREAD DISP TO HANDLE VARIOUS CASES OF ARGUMENT FETCH
; 2349
; 2350 ;CASE 0 -- READ (E)
; 2351 40: MEM READ, ;WAIT FOR DATA
; 2352 [AR]_MEM, ;PUT WORD IN AR
U 0040, 0000,3771,0003,4365,5001,2700,0200,0000,0002 ; 2353 INST DISP ;GO TO EXECUTE CODE
; 2354
; 2355 ;CASE 1 -- WRITE (E)
; 2356 41: [AR]_AC, ;PUT AC IN AR
U 0041, 0000,3771,0003,0276,6001,2700,0000,0000,0000 ; 2357 INST DISP ;GO TO EXECUTE CODE
; 2358
; 2359 ;CASE 2 -- DOUBLE READ
; 2360 42: MEM READ, ;WAIT FOR DATA
U 0042, 0363,3771,0003,4365,5007,0700,0200,0000,0002 ; 2361 [AR]_MEM ;PUT HI WORD IN AR
; 2362 VMA [HR]+1, PXCT DATA, ;POINT TO E+1
U 0363, 0366,0111,0702,4170,4007,0700,0200,0004,0312 ; 2363 START READ ;START MEMORY CYCLE
; 2364 MEM READ, ;WAIT FOR DATA
; 2365 [ARX]_MEM, ;LOW WORD IN ARX
U 0366, 0000,3771,0004,4365,5001,2700,0200,0000,0002 ; 2366 INST DISP ;GO TO EXECUTE CODE
; 2367
; 2368 ;CASE 3 -- DOUBLE AC
; 2369 43: [AR]_AC ;GET HIGH AC
U 0043, 0371,3771,0003,0276,6007,0700,0000,0000,0000 ; 2370 [ARX]_AC[1], ;PUT C(AC+1) IN ARX
U 0371, 0000,3771,0004,1276,6001,2701,0000,0000,1441 ; 2371 INST DISP ;GO TO EXECUTE CODE
; 2372
; 2373 ;CASE 4 -- SHIFT
; 2374 44:
; 2375 SHIFT: READ [AR], ;LOOK AT EFFECTIVE ADDRESS
; 2376 SKIP DP18, ;SEE IF LEFT OR RIGHT
; 2377 SC_SHIFT-1, ;PUT NUMBER OF PLACES TO SHIFT IN
; 2378 LOAD FE, ; SC AND FE
U 0044, 0000,3333,0003,4174,4001,2530,3000,0041,5777 ; 2379 INST DISP ;GO DO THE SHIFT
; 2380
; 2381 ;CASE 5 -- SHIFT COMBINED
; 2382 45: Q_AC[1] ;PUT LOW WORD IN Q
U 0045, 0401,3772,0000,1275,5007,0701,0000,0000,1441 ; 2383 [BR]_AC*.5 LONG ;PUT AC IN BR & SHIFT BR!Q RIGHT
U 0401, 0406,3776,0005,0274,4007,0701,0000,0000,0000 ; 2384 [BR]_[BR]*.5 LONG, ;SHIFT BR!Q 1 MORE PLACE RIGHT
; 2385 J/SHIFT ;GO DO SHIFT SETUP
; 2386
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 67
THE INSTRUCTION LOOP -- FETCH ARGUMENTS

```
U 0046, 0402,3770,0303,4344,4007,0700,0000,0000,0000
; 2387 ;CASE 6 -- FLOATING POINT IMMEDIATE
; 2388 46: [AR]_[AR] SWAP, ;FLIP BITS TO LEFT HALF
; 2389 J/FPRO ;JOIN COMMON F.P. CODE
; 2390
; 2391 ;CASE 7 -- FLOATING POINT
; 2392 47: MEM READ, ;WAIT FOR MEMORY (SPEC/MEM WAIT)
; 2393 [AR]_MEM ;DATA INTO AR
; 2394 =0
; 2395 FPRO: READ [AR], ;LOOK AT NUMBER
; 2396 SC_EXP, FE_EXP, ;PUT EXPONENT IN SC & FE
; 2397 SKIP DPO, ;SEE IF NEGATIVE
; 2398 CALL [ARSIGN] ;EXTEND AR SIGN
; 2399 FPR1: [ARX]_O, ;ZERO ARX
; 2400 INST_DISP ;GO TO EXECUTE CODE
; 2401
; 2402 ;CASE 10 -- READ THEN PREFETCH
; 2403 50: MEM READ, ;WAIT FOR DATA
; 2404 [AR]_MEM THEN FETCH, ;PUT DATA IN AR AND START A READ
; 2405 ; VMA HAS PC+1.
; 2406 INST_DISP ;GO DO IT
; 2407
; 2408 ;CASE 11 -- DOUBLE FLOATING READ
; 2409 51: SPEC MEM READ, ;WAIT FOR DATA
; 2410 [BR]_MEM, ;HOLD IN BR
; 2411 SC_EXP, FE_EXP, ;SAVE EXPONENT
; 2412 SKIP DPO, 3T ;SEE IF MINUS
; 2413 =0 [AR]_[AR]+1, ;POINT TO E+1
; 2414 LOAD VMA, PXCT DATA, ;PUT IN VMA
; 2415 START READ, J/DFPR1 ;GO GET POSITIVE DATA
; 2416 [AR]_[AR]+1, ;POINT TO E+1
; 2417 LOAD VMA, PXCT DATA, ;PUT IN VMA
; 2418 START READ ;GO GET NEGATIVE DATA
; 2419 [BR]_-SIGN, ;SMEAR MINUS SIGN
; 2420 J/DFPR2 ;CONTINUE BELOW
; 2421 DFPR1: [BR]_+SIGN ;SMEAR PLUS SIGN
; 2422 DFPR2: MEM READ, 3T, ;WAIT FOR MEMORY
; 2423 [ARX]_(MEM.AND.[MAG])* .5,
; 2424 ASH ;SET SHIFT PATHS
; 2425 [AR]_[BR]*.5 ;SHIFT AR
; 2426 [AR]_[AR]*.5, ;COMPLETE SHIFTING
; 2427 SC_FE ;PAGE FAIL MAY HAVE ZAPPED
; 2428 ; THE SC.
; 2429 VMA [PC], FETCH, ;GET NEXT INST
; 2430 INST_DISP ;DO THIS ONE
; 2431
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 68
THE INSTRUCTION LOOP -- FETCH ARGUMENTS

```
U 0052, 0414,4443,0000,4174,4007,0040,0000,0000,0000 ; 2432 ;CASE 12 -- TEST FOR IO LEGAL
U 0414, 2621,4551,0202,4374,0007,0700,0000,0077,7740 ; 2433 52: SKIP IO LEGAL ;IS IO LEGAL?
U 0415, 0000,4443,0000,4174,4001,2700,0000,0000,0000 ; 2434 =0 U0 ;NO
; 2435 INST DISP ;YES--DO THE INSTRUCTION
; 2436
; 2437
; 2438 ;CASE 13 -- RESERVED
; 2439 ;53:
; 2440
; 2441 ;CASE 14 -- RESERVED
; 2442 ;54:
; 2443
; 2444 ;CASE 15 -- RESERVED
; 2445 ;55:
; 2446
; 2447 ;CASE 16 -- RESERVED
; 2448 ;56:
; 2449
; 2450 ;CASE 17 -- RESERVED
; 2451 ;57:
; 2452
; 2453 ;EXTEND AR SIGN.
; 2454 ;CALL WITH SKIP ON ARO, RETURNS 1 ALWAYS
; 2455 =0
; 2456 ARSIGN: [AR]_+SIGN, RETURN [1] ;EXTEND + SIGN
U 0434, 0001,4551,0303,4374,0004,1700,0000,0000,0777 ; 2457 [AR]_-SIGN, RETURN [1] ;EXTEND - SIGN
U 0435, 0001,3551,0303,4374,0004,1700,0000,0077,7000 ; 2458
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 69
THE INSTRUCTION LOOP -- STORE ANSWERS

```

; 2459 .TDC "THE INSTRUCTION LOOP -- STORE ANSWERS"
; 2460
; 2461 ;NOTE: INSTRUCTIONS WHICH STORE IN BOTH AC AND MEMORY
; 2462 ; (E.G. ADDB, AOS) MUST STORE IN MEMORY FIRST
; 2463 ; SO THAT IF A PAGE FAIL HAPPENS THE AC IS
; 2464 ; STILL INTACT.
; 2465
; 2466 1500:
; 2467 BWRITE: ;BASE ADDRESS OF BWRITE DISPATCH
; 2468
; 2469 ;CASE 0 -- RESERVED
; 2470 ;1500:
; 2471
; 2472 ;CASE 1 -- RESERVED
; 2473 ;1501:
; 2474
; 2475 ;CASE 2 -- RESERVED
; 2476 ;1502:
; 2477
; 2478 ;CASE 3 -- RESERVED
; 2479 ;1503:
; 2480
; 2481 ;CASE 4 -- STORE SELF
; 2482 1504:
; 2483 STSELF: SKIP IF ACO, ;IS AC # ZERO?
; 2484 J/STBTH1 ;GO TO STORE BOTH CASE
; 2485
; 2486 ;CASE 5 -- STORE DOUBLE AC
; 2487 1505:
; 2488 DAC: AC[1][ARX], ;STORE AC 1
; 2489 J/STAC ;GO STORE AC
; 2490
; 2491 ;CASE 6 -- STORE DOUBLE BOTH (KA10 STYLE MEM_AR ONLY)
; 2492 1506:
; 2493 STDBTH: MEM WRITE, ;WAIT FOR MEMORY
; 2494 MEM[AR], ;STORE AR
; 2495 J/DAC ;NOW STORE AC & AC+1
; 2496
; 2497 ;CASE 7 -- RESERVED
; 2498 ;1507:
; 2499
```

U 1504, 0454,4443,0000,4174,4007,0360,0000,0000,0000

U 1505, 1515,3440,0404,1174,4007,0700,0400,0000,1441

U 1506, 1505,3333,0003,4175,5007,0701,0200,0000,0002

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 70
THE INSTRUCTION LOOP -- STORE ANSWERS

```

; 2500 ;CASE 10 -- RESERVED
; 2501 ;1510:
; 2502
; 2503 ;CASE 11 -- RESERVED
; 2504 ;1511:
; 2505
; 2506 ;CASE 12 -- RESERVED
; 2507 ;1512:
; 2508
; 2509 ;CASE 13 -- RESERVED
; 2510 ;1513:
; 2511
; 2512 ;CASE 14 -- RESERVED
; 2513 1514:
; 2514 FL-BWRITE: ;THE NEXT 4 CASES ARE ALSO
; 2515 ;USED IN FLOATING POINT
; 2516 HALT [BW14]
; 2517
; 2518 ;CASE 15 -- STORE AC
; 2519 1515:
; 2520 STAC: AC [AR], ;STORE AC
; 2521 NEXT INST ;DO NEXT INSTRUCTION
; 2522
; 2523 ;CASE 16 -- STORE IN MEMORY
; 2524 1516:
; 2525 STMEM: MEM WRITE, ;WAIT FOR MEMORY
; 2526 MEM [AR], ;STORE AR
; 2527 J/DONE ;START FETCH OF NEXT
; 2528
; 2529 ;CASE 17 -- STORE BOTH
; 2530 1517:
; 2531 STBOTH: MEM WRITE, ;WAIT FOR MEMORY
; 2532 MEM [AR], ;STORE AR
; 2533 J/STAC ;NOW STORE AC
; 2534
; 2535 =0
; 2536 STBTH1: MEM WRITE, ;WAIT FOR MEMORY
; 2537 MEM [AR], ;STORE AR
; 2538 J/STAC ;NOW STORE AC
; 2539 STORE: MEM WRITE, ;WAIT FOR MEMORY
; 2540 MEM [AR], ;STORE AC
; 2541 J/DONE ;START NEXT INST
; 2542

U 1514, 0104,4751,1217,4374,4007,0700,0000,0000,1000
U 1515, 0100,3440,0303,0174,4156,4700,0400,0000,0000
U 1516, 1400,3333,0003,4175,5007,0701,0200,0000,0002
U 1517, 1515,3333,0003,4175,5007,0701,0200,0000,0002
U 0454, 1515,3333,0003,4175,5007,0701,0200,0000,0002
U 0455, 1400,3333,0003,4175,5007,0701,0200,0000,0002
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 71
MOVE GROUP

```

; 2543 .TOC "MOVE GROUP"
; 2544
; 2545 .DCODE
; 2546 200: R-PF, AC, J/STAC ;MOVE
; 2547 I-PF, AC, J/STAC ;MOVEI
; 2548 W, M, J/MOVE ;MOVEM
; 2549 RW, S, J/STSELF ;MOVES
; 2550
; 2551 204: R-PF, AC, J/MOVS ;MOVS
; 2552 I-PF, AC, J/MOVS ;MOVSI
; 2553 W, M, J/MOVS ;MOVSM
; 2554 RW, S, J/MOVS ;MOVSS
; 2555
; 2556 210: R-PF, AC, J/MOVN ;MOVN
; 2557 I-PF, AC, J/MOVN ;MOVNI
; 2558 W, M, J/MOVN ;MOVNM
; 2559 RW, S, J/MOVN ;MOVNS
; 2560
; 2561 214: R-PF, AC, J/MOVM ;MOVM
; 2562 I-PF, AC, J/STAC ;MOVMI
; 2563 W, M, J/MOVM ;MOVMM
; 2564 RW, S, J/MOVM ;MOVNS
; 2565 .UCODE
; 2566
; 2567 1402:
; 2568 MOVS: [AR]_[AR] SWAP,EXIT
; 2569
; 2570 1403:
; 2571 MOVM: READ [AR], SKIP DPO, J/MOVE
; 2572
; 2573 1404:
; 2574 MOVE: EXIT
; 2575 1405:
; 2576 MOVN: [AR]_[AR], ;NEGATE NUMBER
; 2577 AD FLAGS, 3T, ;UPDATE FLAGS
; 2578 J/MOVE ;STORE ANSWER
; 2579
```

D 0200, 1015, 1515, 1100
D 0201, 0015, 1515, 3000
D 0202, 0116, 1404, 0700
D 0203, 0004, 1504, 1700

D 0204, 1015, 1402, 1100
D 0205, 0015, 1402, 3000
D 0206, 0116, 1402, 0700
D 0207, 0004, 1402, 1700

D 0210, 1015, 1405, 1100
D 0211, 0015, 1405, 3000
D 0212, 0116, 1405, 0700
D 0213, 0004, 1405, 1700

D 0214, 1015, 1403, 1100
D 0215, 0015, 1515, 3000
D 0216, 0116, 1403, 0700
D 0217, 0004, 1403, 1700

U 1402, 1500, 3770, 0303, 4344, 4003, 7700, 0200, 0003, 0001

U 1403, 1404, 3333, 0003, 4174, 4007, 0520, 0000, 0000, 0000

U 1404, 1500, 4443, 0000, 4174, 4003, 7700, 0200, 0003, 0001

U 1405, 1404, 2441, 0303, 4174, 4467, 0701, 4000, 0001, 0001

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 72
EXCH

D 0250, 0015, 1406, 1500

U 1406, 0467, 3771, 0005, 0276, 6007, 0700, 0200, 0003, 0002

U 0467, 1515, 3333, 0005, 4175, 5007, 0701, 0200, 0000, 0002

; 2580 .TOC "EXCH"

; 2581

; 2582

; 2583

; 2584

; 2585

; 2586

; 2587

; 2588

; 2589

; 2590

; 2591

; 2592

; 2593

; 2594

250: R,W TEST, AC, J/EXCH
.UCODE

1406:

EXCH:

[BR]_AC,
START WRITE
MEM WRITE,
MEM_[BR],
J/STAC

; COPY AC TO THE BR
; START A WRITE CYCLE
; COMPLETE WRITE CYCLE
; STORE BR (AC) IN MEMORY
; STORE THE AR IN AC. NOTE: AR
; WAS LOADED WITH MEMORY OPERAND
; AS PART OF INSTRUCTION DISPATCH

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 73
HALFWORD GROUP

```

; 2595 .TOC "HALFWORD GROUP"
; 2596 ; DESTINATION LEFT HALF
; 2597
; 2598 .DCODE
D 0500, 1015, 1410, 1100 ; 2599 500: R-PF, AC, J/HLL
D 0501, 0015, 1410, 3000 ; 2600 I-PF, AC, J/HLL
D 0502, 0016, 1407, 1700 ; 2601 RW, M, J/HRR ;HLLM = HRR EXCEPT FOR STORE
D 0503, 0004, 1404, 1700 ; 2602 RW, S, J/MOVE ;HLLS = MOVES
; 2603
D 0504, 1015, 1411, 1100 ; 2604 R-PF, AC, J/HRL
D 0505, 0015, 1411, 3000 ; 2605 I-PF, AC, J/HRL
D 0506, 0016, 1413, 1700 ; 2606 RW, M, J/HRLM
D 0507, 0004, 1414, 1700 ; 2607 RW, S, J/HRLS
; 2608
D 0510, 1015, 1432, 1100 ; 2609 510: R-PF, AC, J/HLLZ
D 0511, 0015, 1432, 3000 ; 2610 I-PF, AC, J/HLLZ
D 0512, 0116, 1432, 0700 ; 2611 W, M, J/HLLZ
D 0513, 0004, 1432, 1700 ; 2612 RW, S, J/HLLZ
; 2613
D 0514, 1015, 1424, 1100 ; 2614 R-PF, AC, J/HRLZ
D 0515, 0015, 1424, 3000 ; 2615 I-PF, AC, J/HRLZ
D 0516, 0116, 1424, 0700 ; 2616 W, M, J/HRLZ
D 0517, 0004, 1424, 1700 ; 2617 RW, S, J/HRLZ
; 2618
D 0520, 1015, 1433, 1100 ; 2619 520: R-PF, AC, J/HLLO
D 0521, 0015, 1433, 3000 ; 2620 I-PF, AC, J/HLLO
D 0522, 0116, 1433, 0700 ; 2621 W, M, J/HLLO
D 0523, 0004, 1433, 1700 ; 2622 RW, S, J/HLLO
; 2623
D 0524, 1015, 1425, 1100 ; 2624 R-PF, AC, J/HRLO
D 0525, 0015, 1425, 3000 ; 2625 I-PF, AC, J/HRLO
D 0526, 0116, 1425, 0700 ; 2626 W, M, J/HRLO
D 0527, 0004, 1425, 1700 ; 2627 RW, S, J/HRLO
; 2628
D 0530, 1015, 1430, 1100 ; 2629 530: R-PF, AC, J/HLLE
D 0531, 0015, 1430, 3000 ; 2630 I-PF, AC, J/HLLE
D 0532, 0116, 1430, 0700 ; 2631 W, M, J/HLLE
D 0533, 0004, 1430, 1700 ; 2632 RW, S, J/HLLE
; 2633
D 0534, 1015, 1422, 1100 ; 2634 R-PF, AC, J/HRLE
D 0535, 0015, 1422, 3000 ; 2635 I-PF, AC, J/HRLE
D 0536, 0116, 1422, 0700 ; 2636 W, M, J/HRLE
D 0537, 0004, 1422, 1700 ; 2637 RW, S, J/HRLE
; 2638
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 74
HALFWORD GROUP

D 0540, 1015,1407,1100
D 0541, 0015,1407,3000
D 0542, 0016,1410,1700
D 0543, 0004,1404,1700

D 0544, 1015,1412,1100
D 0545, 0015,1412,3000
D 0546, 0016,1415,1700
D 0547, 0004,1416,1700

D 0550, 1015,1420,1100
D 0551, 0015,1420,3000
D 0552, 0116,1420,0700
D 0553, 0004,1420,1700

D 0554, 1015,1426,1100
D 0555, 0015,1426,3000
D 0556, 0116,1426,0700
D 0557, 0004,1426,1700

D 0560, 1015,1421,1100
D 0561, 0015,1421,3000
D 0562, 0116,1421,0700
D 0563, 0004,1421,1700

D 0564, 1015,1427,1100
D 0565, 0015,1427,3000
D 0566, 0116,1427,0700
D 0567, 0004,1427,1700

D 0570, 1015,1417,1100
D 0571, 0015,1417,3000
D 0572, 0116,1417,0700
D 0573, 0004,1417,1700

D 0574, 1015,1423,1100
D 0575, 0015,1423,3000
D 0576, 0116,1423,0700
D 0577, 0004,1423,1700

; 2639 ; DESTINATION RIGHT HALF
; 2640
; 2641 540: R-PF, AC, J/HRR
; 2642 I-PF, AC, J/HRR
; 2643 RW, M, J/HLL
; 2644 RW, S, J/MOVE
; 2645
; 2646 R-PF, AC, J/HLR
; 2647 I-PF, AC, J/HLR
; 2648 RW, M, J/HLRM
; 2649 RW, S, J/HLRS
; 2650
; 2651 550: R-PF, AC, J/HRRZ
; 2652 I-PF, AC, J/HRRZ
; 2653 W, M, J/HRRZ
; 2654 RW, S, J/HRRZ
; 2655
; 2656 R-PF, AC, J/HLRZ
; 2657 I-PF, AC, J/HLRZ
; 2658 W, M, J/HLRZ
; 2659 RW, S, J/HLRZ
; 2660
; 2661 560: R-PF, AC, J/HRRO
; 2662 I-PF, AC, J/HRRO
; 2663 W, M, J/HRRO
; 2664 RW, S, J/HRRO
; 2665
; 2666 R-PF, AC, J/HLRO
; 2667 I-PF, AC, J/HLRO
; 2668 W, M, J/HLRO
; 2669 RW, S, J/HLRO
; 2670
; 2671 570: R-PF, AC, J/HRRE
; 2672 I-PF, AC, J/HRRE
; 2673 W, M, J/HRRE
; 2674 RW, S, J/HRRE
; 2675
; 2676 R-PF, AC, J/HLRE
; 2677 I-PF, AC, J/HLRE
; 2678 W, M, J/HLRE
; 2679 RW, S, J/HLRE
; 2680
; 2681 .UCODE
; 2682

;HRRM = HLL EXCEPT FOR STORE
;HRRS = MOVES

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 75
HALFWORD GROUP

```
U 1407, 1500,3771,0003,0276,0003,7700,0200,0003,0001
U 1410, 1500,3771,0003,0270,6003,7700,0200,0003,0001

; 2683 ;FIRST THE GUYS THAT LEAVE THE OTHER HALF ALONE
; 2684
; 2685 ;THE AR CONTAINS THE MEMORY OPERAND. SO WE WANT TO PUT THE LH OF
; 2686 ; AC INTO AR TO DO A HRR. OBVIOUS THING FOR HLL.
; 2687 1407:
; 2688 HRR: [AR]_AC,HOLD RIGHT,EXIT
; 2689 1410:
; 2690 HLL: [AR]_AC,HOLD LEFT,EXIT
; 2691
; 2692 ;HRL FLOW:
; 2693 ;AT HRL AR CONTAINS:
; 2694 ;
; 2695 ; !-----!-----!
; 2696 ; ! LH OF (E) ! RH OF (E) !
; 2697 ; !-----!-----!
; 2698 ;
; 2699 ;AR_AR SWAP GIVES:
; 2700 ;
; 2701 ; !-----!-----!
; 2702 ; ! RH OF (E) ! LH OF (E) !
; 2703 ; !-----!-----!
; 2704 ;
; 2705 ;AT HLL, AR_AC,HOLD LEFT GIVES:
; 2706 ;
; 2707 ; !-----!-----!
; 2708 ; ! RH OF (E) ! RH OF AC !
; 2709 ; !-----!-----!
; 2710 ;
; 2711 ;THE EXIT MACRO CAUSES THE AR TO BE STORED IN AC (AT STAC).
; 2712 ; THE REST OF THE HALF WORD IN THIS GROUP ARE VERY SIMILAR.
; 2713
; 2714 1411:
; 2715 HRL: [AR]_[AR] SWAP,J/HLL
; 2716 1412:
; 2717 HLR: [AR]_[AR] SWAP,J/HRR
; 2718
; 2719 1413:
; 2720 HRLM: [AR]_[AR] SWAP
; 2721 [AR]_AC,HOLD LEFT,J/MOVS
; 2722 1414:
; 2723 HRLS: [AR]_[AR] SWAP,HOLD RIGHT,EXIT
; 2724
; 2725 1415:
; 2726 HLRM: [AR]_[AR] SWAP
; 2727 [AR]_AC,HOLD RIGHT,J/MOVS
; 2728 1416:
; 2729 HRLS: [AR]_[AR] SWAP,HOLD LEFT,EXIT
; 2730
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 76
HALFWORD GROUP

```
U 1417, 1420,3333,0003,4174,4007,0530,0000,0000,0000
U 1420, 1500,5731,0003,4174,4003,7700,0200,0003,0001
U 1421, 1500,5431,1203,4174,4003,7700,0200,0003,0001
U 1422, 1424,3333,0003,4174,4007,0530,0000,0000,0000
U 1424, 1402,3771,0003,4374,0007,0700,0000,0000,0000
U 1425, 1402,3771,0003,4374,0007,0700,0000,0077,7777
U 1423, 1426,3333,0003,4174,4007,0520,0000,0000,0000
U 1426, 1402,3771,0003,4370,4007,0700,0000,0000,0000
U 1427, 1402,3771,0003,4370,4007,0700,0000,0077,7777
U 1430, 1432,3333,0003,4174,4007,0520,0000,0000,0000
U 1432, 1500,5371,0003,4174,4003,7700,0200,0003,0001
U 1433, 1500,5341,1203,4174,4003,7700,0200,0003,0001

; 2731 ;NOW THE HALFWORD OPS WHICH CONTROL THE "OTHER" HALF.
; 2732 ; ENTER WITH O,,E (E) OR (AC) IN AR
; 2733
; 2734 1417:
; 2735 HRRE: READ [AR],SKIP DP18
; 2736 1420:
; 2737 HRRZ: [AR] LEFT_O, EXIT
; 2738 1421:
; 2739 HRRO: [AR] LEFT_-1, EXIT
; 2740
; 2741 1422:
; 2742 HRLE: READ [AR],SKIP DP18
; 2743 1424:
; 2744 HRLZ: [AR]_#,#/O,HOLD RIGHT,J/MOVS
; 2745 1425:
; 2746 HRLO: [AR]_#,#/777777,HOLD RIGHT,J/MOVS
; 2747
; 2748 1423:
; 2749 HLRE: READ [AR],SKIP DPO
; 2750 1426:
; 2751 HLRZ: [AR]_#,#/O,HOLD LEFT,J/MOVS
; 2752 1427:
; 2753 HLRO: [AR]_#,#/777777,HOLD LEFT,J/MOVS
; 2754
; 2755 1430:
; 2756 HLLE: READ [AR],SKIP DPO
; 2757 1432:
; 2758 HLLZ: [AR] RIGHT_O, EXIT
; 2759 1433:
; 2760 HLLD: [AR] RIGHT_-1, EXIT
; 2761
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 77
DMOVE, DMOVN, DMOVEM, DMOVNM

```

; 2762 .TOC "DMOVE, DMOVN, DMOVEM, DMOVNM"
; 2763
; 2764 .DCODE
D 0120, 0205,1505,1100 ; 2765 120: DBL R, DAC, J/DAC
D 0121, 0215,1434,1100 ; 2766 DBL R, AC, J/DMOVN
; 2767 .UCODE
; 2768
; 2769 1434:
U 1434, 2752,4551,0404,4374,0007,0700,0010,0037,7777 ; 2770 DMOVN: CLEAR ARXO, CALL [DBLNGA]
U 1436, 1515,3440,0404,1174,4007,0700,0400,0000,1441 ; 2771 1436: AC[1]_[ARX], J/STAC
; 2772
; 2773 .DCODE
D 0124, 0300,1567,0100 ; 2774 124: DBL AC, J/DMOVN1
D 0125, 0100,1565,0500 ; 2775 W, J/DMOVNM
; 2776 .UCODE
; 2777
; 2778
; 2779 1565:
U 1565, 2751,3771,0004,1276,6007,0701,0010,0000,1441 ; 2780 DMOVNM: [ARX]_AC[1],CALL [DBLNEG]
; 2781 1567:
; 2782 DMOVN1: [HR]+[ONE], ;GET E+1
; 2783 LOAD VMA, ;PUT THAT IN VMA
; 2784 START WRITE, ;STORE IN E+1
; 2785 PXCT DATA ;DATA CYCLE
; 2786 MEM WRITE, MEM_[ARX] ;STORE LOW WORD
; 2787 VMA_[HR], ;GET E
; 2788 LOAD VMA, ;SAVE IN VMA
; 2789 PXCT DATA, ;OPERAND STORE
; 2790 START WRITE, ;START MEM CYCLE
U 0512, 0455,3443,0200,4174,4007,0700,0200,0003,0312 ; 2791 J/STORE ;GO STORE AR
; 2792
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 78
BOOLEAN GROUP

```
; 2793 .TOC "BOOLEAN GROUP"
; 2794
; 2795 .DCODE
D 0400, 0015,1441,3000 ; 2796 400: I-PF, AC, J/SETZ
D 0401, 0015,1441,3000 ; 2797 I-PF, AC, J/SETZ
D 0402, 0016,1441,2700 ; 2798 IW, M, J/SETZ
D 0403, 0017,1441,2700 ; 2799 IW, B, J/SETZ
; 2800 .UCODE
; 2801
; 2802 1441:
U 1441, 1500,4221,0003,4174,4003,7700,0200,0003,0001 ; 2803 SETZ: [AR]_O, EXIT
; 2804
; 2805 .DCODE
D 0404, 1015,1442,1100 ; 2806 404: R-PF, AC, J/AND
D 0405, 0015,1442,3000 ; 2807 I-PF, AC, J/AND
D 0406, 0016,1442,1700 ; 2808 RW, M, J/AND
D 0407, 0017,1442,1700 ; 2809 RW, B, J/AND
; 2810 .UCODE
; 2811
; 2812 1442:
U 1442, 1500,4551,0303,0274,4003,7700,0200,0003,0001 ; 2813 AND: [AR]_[AR].AND.AC,EXIT
; 2814
; 2815 .DCODE
D 0410, 1015,1443,1100 ; 2816 410: R-PF, AC, J/ANDCA
D 0411, 0015,1443,3000 ; 2817 I-PF, AC, J/ANDCA
D 0412, 0016,1443,1700 ; 2818 RW, M, J/ANDCA
D 0413, 0017,1443,1700 ; 2819 RW, B, J/ANDCA
; 2820 .UCODE
; 2821
; 2822 1443:
U 1443, 1500,5551,0303,0274,4003,7700,0200,0003,0001 ; 2823 ANDCA: [AR]_[AR].AND.NOT.AC,EXIT
; 2824
; 2825 .DCODE
D 0414, 1015,1404,1100 ; 2826 414: R-PF, AC, J/MOVE ;SETM = MOVE
D 0415, 0015,1404,3000 ; 2827 I-PF, AC, J/MOVE
D 0416, 0016,1404,1700 ; 2828 RW, M, J/MOVE ;SETMM = NOP THAT WRITES MEMORY
D 0417, 0017,1404,1700 ; 2829 RW, B, J/MOVE ;SETMB = MOVE THAT WRITES MEMORY
; 2830
; 2831 420: R-PF, AC, J/ANDCM
D 0420, 1015,1444,1100 ; 2832 I-PF, AC, J/ANDCM
D 0421, 0015,1444,3000 ; 2833 RW, M, J/ANDCM
D 0422, 0016,1444,1700 ; 2834 RW, B, J/ANDCM
D 0423, 0017,1444,1700 ; 2835 .UCODE
; 2836
; 2837 1444:
U 1444, 1442,7441,0303,4174,4007,0700,0000,0000,0000 ; 2838 ANDCM: [AR]_.NOT.[AR],J/AND
; 2839
; 2840 .DCODE
D 0424, 0000,1400,1100 ; 2841 424: R, J/DONE
D 0425, 0000,1400,2100 ; 2842 I, J/DONE
D 0426, 0116,1404,0700 ; 2843 W, M, J/MOVE ;SETAM = MOVEM
D 0427, 0116,1404,0700 ; 2844 W, M, J/MOVE ;SETAB, TOO
; 2845 .UCODE
; 2846
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 79
BOOLEAN GROUP

```

; 2847          .DCODE
D 0430, 1015,1445,1100 ; 2848 430:      R-PF,  AC,    J/XOR
D 0431, 0015,1445,3000 ; 2849          I-PF,  AC,    J/XOR
D 0432, 0016,1445,1700 ; 2850          RW,    M,    J/XOR
D 0433, 0017,1445,1700 ; 2851          RW,    B,    J/XOR
; 2852          .UCODE
; 2853
; 2854 1445:
U 1445, 1500,6551,0303,0274,4003,7700,0200,0003,0001 ; 2855 XOR:      [AR]_[AR].XOR.AC,EXIT
; 2856
; 2857          .DCODE
D 0434, 1015,1446,1100 ; 2858 434:      R-PF,  AC,    J/IOR
D 0435, 0015,1446,3000 ; 2859          I-PF,  AC,    J/IOR
D 0436, 0016,1446,1700 ; 2860          RW,    M,    J/IOR
D 0437, 0017,1446,1700 ; 2861          RW,    B,    J/IOR
; 2862          .UCODE
; 2863
; 2864 1446:
U 1446, 1500,3551,0303,0274,4003,7700,0200,0003,0001 ; 2865 IOR:      [AR]_[AR].OR.AC,EXIT
; 2866
; 2867          .DCODE
D 0440, 1015,1447,1100 ; 2868 440:      R-PF,  AC,    J/ANDCB
D 0441, 0015,1447,3000 ; 2869          I-PF,  AC,    J/ANDCB
D 0442, 0016,1447,1700 ; 2870          RW,    M,    J/ANDCB
D 0443, 0017,1447,1700 ; 2871          RW,    B,    J/ANDCB
; 2872          .UCODE
; 2873
; 2874 1447:
U 1447, 1443,7441,0303,4174,4007,0700,0000,0000,0000 ; 2875 ANDCB:    [AR]_.NOT.[AR],J/ANDCA
; 2876
; 2877          .DCODE
D 0444, 1015,1450,1100 ; 2878 444:      R-PF,  AC,    J/EQV
D 0445, 0015,1450,3000 ; 2879          I-PF,  AC,    J/EQV
D 0446, 0016,1450,1700 ; 2880          RW,    M,    J/EQV
D 0447, 0017,1450,1700 ; 2881          RW,    B,    J/EQV
; 2882          .UCODE
; 2883
; 2884 1450:
U 1450, 1500,7551,0303,0274,4003,7700,0200,0003,0001 ; 2885 EQV:      [AR]_[AR].EQV.AC,EXIT
; 2886
; 2887          .DCODE
D 0450, 0015,1451,3000 ; 2888 450:      I-PF,  AC,    J/SETCA
D 0451, 0015,1451,3000 ; 2889          I-PF,  AC,    J/SETCA
D 0452, 0016,1451,2700 ; 2890          IW,    M,    J/SETCA
D 0453, 0017,1451,2700 ; 2891          IW,    B,    J/SETCA
; 2892          .UCODE
; 2893
; 2894 1451:
U 1451, 1500,7771,0003,0274,4003,7700,0200,0003,0001 ; 2895 SETCA:    [AR]_.NOT.AC,EXIT
; 2896
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 80
BOOLEAN GROUP

```

D 0454, 1015,1452,1100
D 0455, 0015,1452,3000
D 0456, 0016,1452,1700
D 0457, 0017,1452,1700

U 1452, 0531,7771,0005,0274,4007,0700,0000,0000,0000
U 0531, 1500,3111,0503,4174,4003,7700,0200,0003,0001

D 0460, 1015,1453,1100
D 0461, 0015,1453,3000
D 0462, 0016,1453,1700
D 0463, 0017,1453,1700

U 1453, 1500,7441,0303,4174,4003,7700,0200,0003,0001

D 0464, 1015,1454,1100
D 0465, 0015,1454,3000
D 0466, 0016,1454,1700
D 0467, 0017,1454,1700

U 1454, 1446,7441,0303,4174,4007,0700,0000,0000,0000

D 0470, 1015,1455,1100
D 0471, 0015,1455,3000
D 0472, 0016,1455,1700
D 0473, 0017,1455,1700

U 1455, 1453,4551,0303,0274,4007,0700,0000,0000,0000

D 0474, 0015,1456,3000
D 0475, 0015,1456,3000
D 0476, 0016,1456,2700
D 0477, 0017,1456,2700

U 1456, 1500,2441,0703,4174,4003,7700,4200,0003,0001

; 2897 .DCODE
; 2898 454: R-PF, AC, J/ORCA
; 2899 I-PF, AC, J/ORCA
; 2900 RW, M, J/ORCA
; 2901 RW, B, J/ORCA
; 2902 .UCODE
; 2903
; 2904 1452:
; 2905 ORCA: [BR]_.NOT.AC
; 2906 [AR]_[AR].OR.[BR],EXIT
; 2907
; 2908 .DCODE
; 2909 460: R-PF, AC, J/SETCM
; 2910 I-PF, AC, J/SETCM
; 2911 RW, M, J/SETCM
; 2912 RW, B, J/SETCM
; 2913 .UCODE
; 2914
; 2915 1453:
; 2916 SETCM: [AR]_.NOT.[AR],EXIT
; 2917
; 2918 .DCODE
; 2919 464: R-PF, AC, J/ORCM
; 2920 I-PF, AC, J/ORCM
; 2921 RW, M, J/ORCM
; 2922 RW, B, J/ORCM
; 2923 .UCODE
; 2924
; 2925 1454:
; 2926 ORCM: [AR]_.NOT.[AR],J/IOR
; 2927
; 2928 .DCODE
; 2929 470: R-PF, AC, J/ORCB
; 2930 I-PF, AC, J/ORCB
; 2931 RW, M, J/ORCB
; 2932 RW, B, J/ORCB
; 2933 .UCODE
; 2934
; 2935 1455:
; 2936 ORCB: [AR]_[AR].AND.AC,J/SETCM
; 2937
; 2938 .DCODE
; 2939 474: I-PF, AC, J/SETO
; 2940 I-PF, AC, J/SETO
; 2941 IW, M, J/SETO
; 2942 IW, B, J/SETO
; 2943 .UCODE
; 2944
; 2945 1456:
; 2946 SETO: [AR]_-[ONE],EXIT
; 2947
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 81
ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO

D 0240, 0400, 1622, 1000
D 0241, 0400, 1632, 1000
D 0242, 0400, 1612, 1000
D 0243, 0000, 1462, 2100
D 0244, 0000, 1466, 3000
D 0245, 0500, 1470, 1000
D 0246, 0500, 1464, 1000

```
; 2948 .TOC "ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO"  
; 2949  
; 2950 .DCODE  
; 2951 240: SH, J/ASH  
; 2952 SH, J/ROT  
; 2953 SH, J/LSH  
; 2954 I, J/JFFO  
; 2955 I-PF, J/ASHC  
; 2956 245: SHC, J/ROTC  
; 2957 SHC, J/LSHC  
; 2958 .UCODE  
; 2959  
; 2960
```

```
; 2961 ;HERE IS THE CODE FOR LOGICAL SHIFT. THE EFFECTIVE ADDRESS IS  
; 2962 ; IN AR.  
; 2963 1612:
```

U 1612, 0532, 3771, 0003, 0276, 6007, 0700, 1000, 0031, 1777

```
; 2964 LSH: [AR]_AC, ;PICK UP AC  
; 2965 FE_-FE-1, ;NEGATIVE SHIFT  
; 2966 J/[LSHL] ;SHIFT LEFT  
; 2967 1613: [AR]_AC.AND.MASK, ;MAKE IT LOOK POSITIVE  
; 2968 FE_FE+1, ;UNDO -1 AT SHIFT  
; 2969 J/ASHR ;GO SHIFT RIGHT  
; 2970
```

U 1613, 0552, 4551, 1203, 0276, 6007, 0700, 1000, 0041, 0001

```
; 2971 LSHL: [AR]_[AR]*2, ;SHIFT LEFT  
; 2972 SHIFT, J/STAC ;FAST SHIFT & GO STORE AC  
; 2973
```

U 0532, 1515, 3445, 0303, 4174, 4007, 0700, 1020, 0041, 0001

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 82
ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO

```
; 2974 ;HERE IS THE CODE FOR ARITHMETIC SHIFT. THE EFFECTIVE ADDRESS IS
; 2975 ; IN AR.
; 2976
; 2977 ASH36 LEFT      "[AR]_[AR]*2 LONG, ASHC, STEP SC, ASH AROV"
; 2978
; 2979 1622:
; 2980 ASH:      Q_O, J/ASHLO      ;HARDWARE ONLY DOES ASHC
; 2981 1623:    [AR]_AC,        ;GET THE ARGUMENT
; 2982          FE_FE+1        ;FE HAS NEGATIVE SHIFT COUNT
; 2983 ASHR:    [AR]_[AR]*.5,   ;SHIFT RIGHT
; 2984          ASH, SHIFT,     ;FAST SHIFT
; 2985          J/STAC         ;STORE AC WHEN DONE
; 2986
; 2987 ASHLO:   [AR]_AC*.5,     ;GET INTO 9 CHIPS
; 2988          STEP SC        ;SEE IF NULL SHIFT
; 2989 =0
; 2990 ASHL:    ASH36 LEFT, J/ASHL ;SHIFT LEFT
; 2991          ;SLOW BECAUSE WE HAVE TO
; 2992          ; TEST FOR OVERFLOW
; 2993
; 2994 ASHX:    [AR]_[AR]*2, J/STAC ;SHIFT BACK INTO 10 CHIPS
; 2995
```

U 1622, 0564,4222,0000,4174,4007,0700,0000,0000,0000
U 1623, 0552,3771,0003,0276,6007,0700,1000,0041,0001
U 0552, 1515,3447,0303,4174,4007,0700,1020,0041,0001
U 0564, 0502,3777,0003,0274,4007,0631,2000,0060,0000
U 0502, 0502,3444,0303,4174,4447,0630,2000,0060,0000
U 0503, 1515,3445,0303,4174,4007,0700,0000,0000,0000

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 83
ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO

```

; 2996 ;HERE IS THE CODE FOR ROTATE. THE EFFECTIVE ADDRESS IS
; 2997 ; IN AR.
; 2998 1632:
; 2999 ROT: [AR]_AC*.5, ;PICK UP THE AC (& SHIFT)
; 3000 FE_-FE-1, ;NEGATIVE SHIFT COUNT
; 3001 J/ROTL ;ROTATE LEFT
; 3002 1633: [AR]_AC*.5, ;PICK UP THE AC (& SHIFT)
; 3003 FE_FE+1 ;NEGATIVE SHIFT COUNT
; 3004 [AR]_[AR]*.5 ;PUT IN 9 DIPS
; 3005 [AR]_[AR]*.5, ;SHIFT RIGHT
; 3006 ROT, SHIFT ;FAST SHIFT
; 3007 ASHXX: [AR]_[AR]*2,J/ASHX ;SHIFT TO STD PLACE
; 3008
; 3009 ROTL: [AR]_[AR]*.5 ;PUT IN RIGHT 36-BITS
; 3010 [AR]_[AR]*2, ;ROTATE LEFT
; 3011 ROT, SHIFT, ;FAST SHIFT
; 3012 J/ASHXX ;ALL DONE--SHIFT BACK
; 3013

U 1632, 0631,3777,0003,0274,4007,0701,1000,0031,1777
U 1633, 0572,3777,0003,0274,4007,0701,1000,0041,0001
U 0572, 0604,3447,0303,4174,4007,0700,0000,0000,0000
U 0604, 0612,3447,0303,4174,4037,0700,1020,0041,0001
U 0612, 0503,3445,0303,4174,4007,0700,0000,0000,0000
U 0631, 0646,3447,0303,4174,4007,0700,0000,0000,0000
U 0646, 0612,3445,0303,4174,4037,0700,1020,0041,0001
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 84
ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO

```
U 1462, 0514,4551,1205,0276,6007,0622,0000,0000,0000
; 3014 1462:
; 3015 JFFO: [BR]_AC.AND.MASK, 4T, ;GET AC WITH NO SIGN
; 3016 SKIP AD.EQ.0 ; EXTENSION. SKIP IF
; 3017 ; ZERO.
; 3018 =0 [PC]_[AR], ;NOT ZERO--JUMP
; 3019 LOAD VMA, FETCH, ;GET NEXT INST
; 3020 J/JFFO1 ;ENTER LOOP
; 3021 AC[1]_O, J/DONE ;ZERO--DONE
; 3022
; 3023 JFFO1: FE_-12. ;WHY -12.? WELL THE
; 3024 ; HARDWARE LOOKS AT
; 3025 ; BIT -2 SO THE FIRST
; 3026 ; 2 STEPS MOVE THE BR
; 3027 ; OVER. WE ALSO LOOK AT
; 3028 ; THE DATA BEFORE THE SHIFT
; 3029 ; SO WE END UP GOING 1 PLACE
; 3030 ; TOO MANY. THAT MEANS THE
; 3031 ; FE SHOULD START AT -3.
; 3032 ; HOWEVER, WE COUNT THE FE BY
; 3033 ; 4 (BECAUSE THE 2 LOW ORDER
; 3034 ; BITS DO NOT COME BACK) SO
; 3035 ; FE_-12.
; 3036 =0
; 3037 JFFOL: [BR]_[BR]*2, ;SHIFT LEFT
; 3038 FE_FE+4, ;COUNT UP BIT NUMBER
; 3039 SKIP DPO, J/JFFOL ;LOOP TILL WE FIND THE BIT
; 3040 [AR]_FE ;GET ANSWER BACK
; 3041 [AR]_[AR].AND.# CLR LH,#/77 ;MASK TO 1 COPY
; 3042 AC[1]_[AR], NEXT INST ;STORE AND EXIT
; 3043

U 0514, 0652,3441,0301,4174,4007,0700,0200,0014,0012
U 0515, 1400,4223,0000,1174,4007,0700,0400,0000,1441

U 0652, 0534,4443,0000,4174,4007,0700,1000,0071,1764

U 0534, 0534,3445,0505,4174,4007,0520,1000,0041,0004
U 0535, 0701,3777,0003,4334,4057,0700,0000,0041,0000
U 0701, 0706,4251,0303,4374,4007,0700,0000,0000,0077
U 0706, 0100,3440,0303,1174,4156,4700,0400,0000,1441
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
 ; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 85
 ROTATES AND LOGICAL SHIFTS -- LSHC

```

; 3044 .TOC "ROTATES AND LOGICAL SHIFTS -- LSHC"
; 3045
; 3046 ;SHIFT CONNECTIONS WHEN THE SPECIAL FUNCTION "LSHC" IS DONE:
; 3047 ;
; 3048 ; !-! |-----|-----|-----|-----|-----|-----|
; 3049 ; !O!-->!0000! HIGH ORDER 36 BITS ! RAM FILE
; 3050 ; !-! |-----|-----|-----|-----|-----|-----|
; 3051 ; ; ; ; ; ; ; ^
; 3052 ; ; ; ; ; ; ; |
; 3053 ; ; ; ; ; ; ; |
; 3054 ; ; ; ; ; ; ; |
; 3055 ; !-----|-----|-----|-----|-----|-----|
; 3056 ; !0000! LOW ORDER 36 BITS ! Q-REGISTER
; 3057 ; !-----|-----|-----|-----|-----|-----|
; 3058 ; ; ; ; ; ; ; ^
; 3059 ; ; ; ; ; ; ; |
; 3060 ; ; ; ; ; ; ; |
; 3061 ; ; ; ; ; ; ; |
; 3062 ; ; ; ; ; ; ; |
; 3063 ; ; ; ; ; ; ; |
; 3064 ; ; ; ; ; ; ; |
; 3065 1464:
; 3066 LSHC: STEP SC, J/LSHCL
; 3067 1465: READ [AR], SC_-SHIFT-1
; 3068 STEP SC
; 3069 =0
; 3070 LSHCR: [BR]_[BR]*.5 LONG,STEP SC,LSHC,J/LSHCR
; 3071 [BR]_[BR]*2 LONG,J/LSHCX
; 3072
; 3073 =0
; 3074 LSHCL: [BR]_[BR]*2 LONG,LSHC,STEP SC,J/LSHCL
; 3075 [BR]_[BR]*2 LONG
; 3076 LSHCX: [BR]_[BR]*2 LONG
; 3077 AC_[BR], J/ASHCQ1
; 3078

```

```

U 1464, 0554,4443,0000,4174,4007,0630,2000,0060,0000
U 1465, 0747,3333,0003,4174,4007,0700,2000,0031,5777
U 0747, 0544,4443,0000,4174,4007,0630,2000,0060,0000

U 0544, 0544,3446,0505,4174,4057,0630,2000,0060,0000
U 0545, 0767,3444,0505,4174,4007,0700,0000,0000,0000

U 0554, 0554,3444,0505,4174,4057,0630,2000,0060,0000
U 0555, 0767,3444,0505,4174,4007,0700,0000,0000,0000
U 0767, 1002,3444,0505,4174,4007,0700,0000,0000,0000
U 1002, 1026,3440,0505,0174,4007,0700,0400,0000,0000

```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 86
ROTATES AND LOGICAL SHIFTS -- ASHC

```
; 3079 .TOC "ROTATES AND LOGICAL SHIFTS -- ASHC"
; 3080
; 3081
; 3082 1466:
; 3083 ASHC: READ [AR], ;PUT AR ON DP
; 3084 SC SHIFT, LOAD FE, ;PUT SHIFT IN BOTH SC AND FE
; 3085 SKIP ADR.EQ.0 ;SEE IF NULL SHIFT
; 3086 =0 Q_AC[1], ;NOT NULL--GET LOW WORD
; 3087 J/ASHC1 ;CONTINUE BELOW
; 3088 NIDISP: NEXT INST ;NULL--ALL DONE
; 3089 ASHC1: [BR]_AC*.5 LONG, ;GET HIGH WORD
; 3090 ;AND SHIFT Q
; 3091 SKIP/SC ;SEE WHICH DIRECTION
; 3092 =0 [BR]_[BR]*.5, ;ADJUST POSITION
; 3093 SC_FE+S#, S#/1776, ;SUBTRACT 2 FROM FE
; 3094 J/ASHCL ;GO LEFT
; 3095 [BR]_[BR]*.5, ;ADJUST POSITION
; 3096 SC_S#-FE, S#/1776 ;SC -2-FE, SC +# OF STEPS
; 3097 =0 ;HERE TO GO RIGHT
; 3098 ASHCR: [BR]_[BR]*.5 LONG, ;GO RIGHT
; 3099 ASHC, ;SET DATA PATHS FOR ASHC (SEE DPE1)
; 3100 STEP SC, J/ASHCR ;COUNT THE STEP AND KEEP LOOPING
; 3101 [BR]_[BR]*2 LONG, ;PUT BACK WHERE IT GOES
; 3102 ASHC, J/ASHCX ;COMPLETE INSTRUCTION
; 3103
; 3104 =0
; 3105 ASHCL: [BR]_[BR]*2 LONG, ;GO LEFT
; 3106 ASHC, ASH AROV, ;SEE IF OVERFLOW
; 3107 STEP SC, J/ASHCL ;LOOP OVER ALL PLACES
; 3108 [BR]_[BR]*2 LONG, ;SHIFT BACK WHERE IT GOES
; 3109 ASHC, ASH AROV ;CAN STILL OVERFLOW
; 3110 ASHCX: AC_[BR]+[BR], 3T, ;PUT BACK HIGH WORD
; 3111 SKIP DPO ;SEE HOW TO FIX LOW SIGN
; 3112 =0 Q_Q.AND.#, #/377777, ;POSITIVE, CLEAR LOW SIGN
; 3113 HOLD RIGHT, J/ASHCQ1 ;GO STORE ANSWER
; 3114 Q_Q.OR.#, #/400000, ;NEGATIVE, SET LOW SIGN
; 3115 HOLD RIGHT ;IN LEFT HALF
; 3116 ASHCQ1: AC[1]_Q, NEXT INST ;PUT BACK Q AND EXIT
; 3117
```

```
; T1QKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984
```

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 87
ROTATES AND LOGICAL SHIFTS -- ROTC

```
U 1470, 0742,4443,0000,4174,4007,0630,2000,0060,0000
U 1471, 1033,3333,0003,4174,4007,0700,2000,0031,5777
U 1033, 0644,4443,0000,4174,4007,0630,2000,0060,0000

U 0644, 0644,3446,0505,4174,4077,0630,2000,0060,0000
U 0645, 0767,3444,0505,4174,4007,0700,0000,0000,0000

U 0742, 0742,3444,0505,4174,4077,0630,2000,0060,0000
U 0743, 0767,3444,0505,4174,4007,0700,0000,0000,0000
```

```
; 3118 .TOC "ROTATES AND LOGICAL SHIFTS -- ROTC"
; 3119
; 3120 ;SHIFT CONNECTIONS WHEN THE SPECIAL FUNCTION "ROTC" IS DONE:
; 3121 ;
; 3122 ;          !-----!-----!
; 3123 ;          >!0000!           HIGH ORDER 36 BITS          ! RAM FILE
; 3124 ;          !-----!-----!
; 3125 ;          :
; 3126 ;          :
; 3127 ;          :
; 3128 ;          :
; 3129 ;          !-----!-----!
; 3130 ;          ..!0000!           LOW ORDER 36 BITS          ! Q-REGISTER
; 3131 ;          !-----!-----!
; 3132 ;          :
; 3133 ;          :
; 3134 ;          :
; 3135 ;          :
; 3136 ;
; 3137 1470:
; 3138 ROTC: STEP SC, J/ROTCL
; 3139 1471: READ [AR], SC_-SHIFT-1
; 3140 STEP SC
; 3141 =0
; 3142 ROTCR: [BR]_[BR]*.5 LONG,STEP SC,ROTC,J/ROTCR
; 3143 [BR]_[BR]*2 LONG,J/LSHCX
; 3144
; 3145 =0
; 3146 ROTCL: [BR]_[BR]*2 LONG,ROTC,STEP SC,J/ROTCL
; 3147 [BR]_[BR]*2 LONG,
; 3148 J/LSHCX
; 3149
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 88
TEST GROUP

```
; 3150 .TOC "TEST GROUP"
; 3151
; 3152 .DCODE
; 3153
; 3154 ;SPECIAL MACROS USED ONLY IN B-FIELD OF TEST INSTRUCTIONS
; 3155 TN- "B/4"
; 3156 TNE "B/0"
; 3157 WORD-TNE "B/10" ;USED IN TIOE
; 3158 TNA "B/0"
; 3159 TNN "B/4"
; 3160 WORD-TNN "B/14" ;USED IN TION
; 3161 TZ- "B/5"
; 3162 TZE "B/1"
; 3163 TZA "B/1"
; 3164 TZN "B/5"
; 3165 TC- "B/6"
; 3166 TCE "B/2"
; 3167 TCA "B/2"
; 3168 TCN "B/6"
; 3169 TO- "B/7"
; 3170 TOE "B/3"
; 3171 TOA "B/3"
; 3172 TON "B/7"
; 3173
; 3174 600: I, J/DONE ;TRN- IS NOP
; 3175 I, J/DONE ;SO IS TLN-
; 3176 I, TNE, J/TDXX
; 3177 I, TNE, J/TSXX
; 3178 I, TNA, J/TDX
; 3179 I, TNA, J/TSX
; 3180 I, TNN, J/TDXX
; 3181 I, TNN, J/TSXX
; 3182
; 3183 610: I, J/DONE ;TDN- IS A NOP
; 3184 I, J/DONE ;TSN- ALSO
; 3185 R, TNE, J/TDXX
; 3186 R, TNE, J/TSXX
; 3187 R, TNA, J/TDX
; 3188 R, TNA, J/TSX
; 3189 R, TNN, J/TDXX
; 3190 R, TNN, J/TSXX
; 3191
; 3192 620: I, TZ-, J/TDX
; 3193 I, TZ-, J/TSX
; 3194 I, TZE, J/TDXX
; 3195 I, TZE, J/TSXX
; 3196 I, TZA, J/TDX
; 3197 I, TZA, J/TSX
; 3198 I, TZN, J/TDXX
; 3199 I, TZN, J/TSXX
; 3200
```

D 0600, 0000, 1400, 2100
D 0601, 0000, 1400, 2100
D 0602, 0000, 1475, 2100
D 0603, 0000, 1474, 2100
D 0604, 0000, 1473, 2100
D 0605, 0000, 1472, 2100
D 0606, 0004, 1475, 2100
D 0607, 0004, 1474, 2100

D 0610, 0000, 1400, 2100
D 0611, 0000, 1400, 2100
D 0612, 0000, 1475, 1100
D 0613, 0000, 1474, 1100
D 0614, 0000, 1473, 1100
D 0615, 0000, 1472, 1100
D 0616, 0004, 1475, 1100
D 0617, 0004, 1474, 1100

D 0620, 0005, 1473, 2100
D 0621, 0005, 1472, 2100
D 0622, 0001, 1475, 2100
D 0623, 0001, 1474, 2100
D 0624, 0001, 1473, 2100
D 0625, 0001, 1472, 2100
D 0626, 0005, 1475, 2100
D 0627, 0005, 1474, 2100

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 89
TEST GROUP

D 0630, 0005,1473,1100
D 0631, 0005,1472,1100
D 0632, 0001,1475,1100
D 0633, 0001,1474,1100
D 0634, 0001,1473,1100
D 0635, 0001,1472,1100
D 0636, 0005,1475,1100
D 0637, 0005,1474,1100

D 0640, 0006,1473,2100
D 0641, 0006,1472,2100
D 0642, 0002,1475,2100
D 0643, 0002,1474,2100
D 0644, 0002,1473,2100
D 0645, 0002,1472,2100
D 0646, 0006,1475,2100
D 0647, 0006,1474,2100

D 0650, 0006,1473,1100
D 0651, 0006,1472,1100
D 0652, 0002,1475,1100
D 0653, 0002,1474,1100
D 0654, 0002,1473,1100
D 0655, 0002,1472,1100
D 0656, 0006,1475,1100
D 0657, 0006,1474,1100
D 0660, 0007,1473,2100
D 0661, 0007,1472,2100
D 0662, 0003,1475,2100
D 0663, 0003,1474,2100
D 0664, 0003,1473,2100
D 0665, 0003,1472,2100
D 0666, 0007,1475,2100
D 0667, 0007,1474,2100

D 0670, 0007,1473,1100
D 0671, 0007,1472,1100
D 0672, 0003,1475,1100
D 0673, 0003,1474,1100
D 0674, 0003,1473,1100
D 0675, 0003,1472,1100
D 0676, 0007,1475,1100
D 0677, 0007,1474,1100

| | | | | |
|--------|------|----|------|--------|
| ; 3201 | 630: | R, | TZ- | J/TDX |
| ; 3202 | | R, | TZ- | J/TSX |
| ; 3203 | | R, | TZE, | J/TDXX |
| ; 3204 | | R, | TZE, | J/TSXX |
| ; 3205 | | R, | TZA, | J/TDX |
| ; 3206 | | R, | TZA, | J/TSX |
| ; 3207 | | R, | TZN, | J/TDXX |
| ; 3208 | | R, | TZN, | J/TSXX |
| ; 3209 | | | | |
| ; 3210 | 640: | I, | TC- | J/TDX |
| ; 3211 | | I, | TC- | J/TSX |
| ; 3212 | | I, | TCE, | J/TDXX |
| ; 3213 | | I, | TCE, | J/TSXX |
| ; 3214 | | I, | TCA, | J/TDX |
| ; 3215 | | I, | TCA, | J/TSX |
| ; 3216 | | I, | TCN, | J/TDXX |
| ; 3217 | | I, | TCN, | J/TSXX |
| ; 3218 | | | | |
| ; 3219 | 650: | R, | TC- | J/TDX |
| ; 3220 | | R, | TC- | J/TSX |
| ; 3221 | | R, | TCE, | J/TDXX |
| ; 3222 | | R, | TCE, | J/TSXX |
| ; 3223 | | R, | TCA, | J/TDX |
| ; 3224 | | R, | TCA, | J/TSX |
| ; 3225 | | R, | TCN, | J/TDXX |
| ; 3226 | | R, | TCN, | J/TSXX |
| ; 3227 | 660: | I, | TO- | J/TDX |
| ; 3228 | | I, | TO- | J/TSX |
| ; 3229 | | I, | TOE, | J/TDXX |
| ; 3230 | | I, | TOE, | J/TSXX |
| ; 3231 | | I, | TOA, | J/TDX |
| ; 3232 | | I, | TOA, | J/TSX |
| ; 3233 | | I, | TON, | J/TDXX |
| ; 3234 | | I, | TON, | J/TSXX |
| ; 3235 | | | | |
| ; 3236 | 670: | R, | TO- | J/TDX |
| ; 3237 | | R, | TO- | J/TSX |
| ; 3238 | | R, | TOE, | J/TDXX |
| ; 3239 | | R, | TOE, | J/TSXX |
| ; 3240 | | R, | TOA, | J/TDX |
| ; 3241 | | R, | TOA, | J/TSX |
| ; 3242 | | R, | TON, | J/TDXX |
| ; 3243 | | R, | TON, | J/TSXX |
| ; 3244 | | | | |

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 90
TEST GROUP

```
; 3245          .UCODE
; 3246
; 3247 ;THESE 64 INSTRUCTIONS ARE DECODED BY MASK MODE(IMMEDIATE OR MEMORY)
; 3248 ; IN THE A FIELD, DISPATCH TO HERE ON THE J FIELD, AND RE-DISPATCH
; 3249 ; FOR THE MODIFICATION ON THE B FIELD.
; 3250
; 3251 ; ENTER WITH O,E OR (E) IN AR, B FIELD BITS 2 AND 3 AS FOLLOWS:
; 3252 ; 0 0  NO MODIFICATION
; 3253 ; 0 1  OS
; 3254 ; 1 0  COMPLEMENT
; 3255 ; 1 1  ONES
; 3256 ; THIS ORDER HAS NO SIGNIFICANCE EXCEPT THAT IT CORRESPONDS TO THE
; 3257 ; ORDER OF INSTRUCTIONS AT TGROUP.
; 3258
; 3259 ;THE BIT 1 OF THE B FIELD IS USED TO DETERMINE THE SENSE
; 3260 ; OF THE SKIP
; 3261 ; 1  SKIP IF AC.AND.MASK .NE. 0 (TXX- AND TXXN)
; 3262 ; 0  SKIP IF AC.AND.MASK .EQ. 0 (TXXA AND TXXE)
; 3263
; 3264 ;BIT 0 IS UNUSED AND MUST BE ZERO
; 3265
; 3266
; 3267 1472:
; 3268 TSX:  [AR]_[AR] SWAP          ;TSXX AND TLXX
; 3269 1473:
; 3270 TDX:  [BR]_0,TEST DISP      ; ALWAYS AND NEVER SKIP CASES
; 3271
; 3272 1474:
; 3273 TSXX: [AR]_[AR] SWAP          ;TSXE, TSXN, TLXE, TLXN
; 3274 1475:
; 3275 TDXX: [BR]_[AR].AND.AC,      ;TDXE, TDXN, TRXE, TRXN
; 3276 TEST DISP
; 3277
```

U 1472, 1473,3770,0303,4344,4007,0700,0000,0000,0000

U 1473, 0014,4221,0005,4174,4003,7700,0000,0000,0000

U 1474, 1475,3770,0303,4344,4007,0700,0000,0000,0000

U 1475, 0014,4551,0305,0274,4003,7700,0000,0000,0000

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 91
TEST GROUP

```
U 0014, 1400,3333,0005,4174,4007,0571,0000,0000,0000
U 0015, 1036,7441,0303,4174,4007,0700,0000,0000,0000
U 0016, 1043,6551,0303,0274,4007,0700,0000,0000,0000
U 0017, 1043,3551,0303,0274,4007,0700,0000,0000,0000
U 1036, 1043,4551,0303,0274,4007,0700,0000,0000,0000
U 1043, 0014,3440,0303,0174,4007,0700,0400,0000,0000

; 3278 ;TEST DISP DOES AN 8 WAY BRANCH BASED ON THE B-FIELD OF DROM
; 3279
; 3280 =1100
; 3281 TEST-TABLE:
; 3282
; 3283 ;CASE 0 & 4 -- TXNX
; 3284 TXXX: READ [BR], TXXX TEST, 3T, J/DONE
; 3285
; 3286 ;CASE 1 & 5 -- TXZ AND TXZX
; 3287 [AR]_.NOT.[AR],J/TXZX
; 3288
; 3289 ;CASE 2 & 6 -- TXC AND TXCX
; 3290 [AR]_[AR].XOR.AC,J/TDONE
; 3291
; 3292 ;CASE 3 & 7 -- TXO AND TXOX
; 3293 [AR]_[AR].OR.AC,J/TDONE
; 3294
; 3295 ;THE SPECIAL FUNCTION TXXX TEST CAUSES A MICROCODE SKIP IF
; 3296 ; AD.EQ.O AND DROM B IS 0-3 OR AD.NE.O AND DROM B IS 4-7.
; 3297
; 3298 TXZX: [AR]_[AR].AND.AC
; 3299 TDONE: AC_[AR],J/TXXX
; 3300 ; READ BR,TXXX TEST,J/DONE
; 3301
```

; T1OKI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 92
COMPARE -- CAI, CAM

```
; 3302 .TOC "COMPARE -- CAI, CAM"
; 3303
; 3304 .DCODE
; 3305
; 3306 ;SPECIAL B-FIELD ENCODING USED BY SKIP-JUMP-COMPARE CLASS
; 3307 ; INSTRUCTIONS:
; 3308
; 3309 SJC- "B/0" ;NEVER
; 3310 SJCL "B/1" ;LESS
; 3311 SJCE "B/2" ;EQUAL
; 3312 SJCLE "B/3" ;LESS EQUAL
; 3313 SJCA "B/4" ;ALWAYS
; 3314 SJCGE "B/5" ;GREATER THAN OR EQUAL
; 3315 SJCN "B/6" ;NOT EQUAL
; 3316 SJCG "B/7" ;GREATER
; 3317
; 3318 .UCODE
; 3319
; 3320 ;COMPARE TABLE
; 3321 =1000
; 3322 SKIP-COMP-TABLE:
; 3323
; 3324 ;CASE 0 -- NEVER
; 3325 DONE
; 3326
; 3327 ;CASE 1 -- LESS
; 3328 READ [AR], SKIP DPO,J/DONE
; 3329
; 3330 ;CASE 2 -- EQUAL
; 3331 SKIPE: READ [AR], SKIP AD.EQ.O,J/DONE
; 3332
; 3333 ;CASE 3 -- LESS OR EQUAL
; 3334 READ [AR], SKIP AD.LE.O,J/DONE
; 3335
; 3336 ;CASE 4 -- ALWAYS
; 3337 VMA_[PC]+1, NEXT INST FETCH, FETCH
; 3338
; 3339 ;CASE 5 -- GREATER THAN OR EQUAL
; 3340 READ [AR], SKIP DPO,J/SKIP
; 3341
; 3342 ;CASE 6 -- NOT EQUAL
; 3343 READ [AR], SKIP AD.EQ.O,J/SKIP
; 3344
; 3345 ;CASE 7 -- GREATER
; 3346 READ [AR], SKIP AD.LE.O,J/SKIP
; 3347

U 0250, 0110,3443,0100,4174,4156,4700,0200,0014,0012
U 0251, 1400,3333,0003,4174,4007,0520,0000,0000,0000
U 0252, 1400,3333,0003,4174,4007,0621,0000,0000,0000
U 0253, 1400,3333,0003,4174,4007,0421,0000,0000,0000
U 0254, 0110,0111,0701,4170,4156,4700,0200,0014,0012
U 0255, 0372,3333,0003,4174,4007,0520,0000,0000,0000
U 0256, 0372,3333,0003,4174,4007,0621,0000,0000,0000
U 0257, 0372,3333,0003,4174,4007,0421,0000,0000,0000
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 93
COMPARE -- CAI, CAM

```

; 3348 .DCODE
D 0300, 0000, 1400, 2100 ; 3349 300: I, SJC-, J/DONE ;CAI
D 0301, 0001, 1476, 2100 ; 3350 I, SJCL, J/CAIM
D 0302, 0002, 1476, 2100 ; 3351 I, SJCE, J/CAIM
D 0303, 0003, 1476, 2100 ; 3352 I, SJCLE, J/CAIM
D 0304, 0004, 1476, 2100 ; 3353 I, SJCA, J/CAIM
D 0305, 0005, 1476, 2100 ; 3354 I, SJCGE, J/CAIM
D 0306, 0006, 1476, 2100 ; 3355 I, SJCN, J/CAIM
D 0307, 0007, 1476, 2100 ; 3356 I, SJCG, J/CAIM
; 3357
D 0310, 0000, 1476, 1100 ; 3358 310: R, SJC-, J/CAIM ;CAM
D 0311, 0001, 1476, 1100 ; 3359 R, SJCL, J/CAIM
D 0312, 0002, 1476, 1100 ; 3360 R, SJCE, J/CAIM
D 0313, 0003, 1476, 1100 ; 3361 R, SJCLE, J/CAIM
D 0314, 0004, 1476, 1100 ; 3362 R, SJCA, J/CAIM
D 0315, 0005, 1476, 1100 ; 3363 R, SJCGE, J/CAIM
D 0316, 0006, 1476, 1100 ; 3364 R, SJCN, J/CAIM
D 0317, 0007, 1476, 1100 ; 3365 R, SJCG, J/CAIM
; 3366 .UCODE
; 3367
; 3368 1476:
U 1476, 0250, 2551, 0303, 0274, 4003, 7701, 4000, 0000, 0000 ; 3369 CAIM: [AR]_AC-[AR], 3T, SKIP-COMP DISP
; 3370
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 94
ARITHMETIC SKIPS -- AOS, SOS, SKIP

```
; 3371 .TOC "ARITHMETIC SKIPS -- AOS, SOS, SKIP"
; 3372 ;ENTER WITH (E) IN AR
; 3373
; 3374 .DCODE
; 3375 330: R, SJC-, J/SKIPS ;NOT A NOP IF AC .NE. O
; 3376 R, SJCL, J/SKIPS
; 3377 R, SJCE, J/SKIPS
; 3378 R, SJCLE, J/SKIPS
; 3379 R, SJCA, J/SKIPS
; 3380 R, SJCGE, J/SKIPS
; 3381 R, SJCN, J/SKIPS
; 3382 R, SJCG, J/SKIPS
; 3383 .UCODE
; 3384
; 3385 1477:
; 3386 SKIPS: FIX [AR] SIGN,
; 3387 SKIP IF ACO
; 3388 =0 AC [AR].SKIP-COMP DISP
; 3389 SKIP-COMP DISP
; 3390
; 3391 .DCODE
; 3392 350: RW, SJC-, J/AOS
; 3393 RW, SJCL, J/AOS
; 3394 RW, SJCE, J/AOS
; 3395 RW, SJCLE, J/AOS
; 3396 RW, SJCA, J/AOS
; 3397 RW, SJCGE, J/AOS
; 3398 RW, SJCN, J/AOS
; 3399 RW, SJCG, J/AOS
; 3400 .UCODE
; 3401
; 3402 1431:
; 3403 AOS: [AR][AR]+1, 3T, AD FLAGS
; 3404 XOS: START WRITE
; 3405 MEM WRITE.MEM_[AR].J/SKIPS
; 3406
; 3407 .DCODE
; 3408 370: RW, SJC-, J/SOS
; 3409 RW, SJCL, J/SOS
; 3410 RW, SJCE, J/SOS
; 3411 RW, SJCLE, J/SOS
; 3412 RW, SJCA, J/SOS
; 3413 RW, SJCGE, J/SOS
; 3414 RW, SJCN, J/SOS
; 3415 RW, SJCG, J/SOS
; 3416 .UCODE
; 3417
; 3418 1437:
; 3419 SOS: [AR][AR]-1, 3T, AD FLAGS, J/XOS
; 3420
```

D 0330, 0000,1477,1100
D 0331, 0001,1477,1100
D 0332, 0002,1477,1100
D 0333, 0003,1477,1100
D 0334, 0004,1477,1100
D 0335, 0005,1477,1100
D 0336, 0006,1477,1100
D 0337, 0007,1477,1100

U 1477, 0744,3770,0303,4174,0007,0360,0000,0000,0000
U 0744, 0250,3440,0303,0174,4003,7700,0400,0000,0000
U 0745, 0250,4443,0000,4174,4003,7700,0000,0000,0000

D 0350, 0000,1431,1500
D 0351, 0001,1431,1500
D 0352, 0002,1431,1500
D 0353, 0003,1431,1500
D 0354, 0004,1431,1500
D 0355, 0005,1431,1500
D 0356, 0006,1431,1500
D 0357, 0007,1431,1500

U 1431, 1046,0111,0703,4174,4467,0701,0000,0001,0001
U 1046, 1053,4443,0000,4174,4007,0700,0200,0003,0002
U 1053, 1477,3333,0003,4175,5007,0701,0200,0000,0002

D 0370, 0000,1437,1500
D 0371, 0001,1437,1500
D 0372, 0002,1437,1500
D 0373, 0003,1437,1500
D 0374, 0004,1437,1500
D 0375, 0005,1437,1500
D 0376, 0006,1437,1500
D 0377, 0007,1437,1500

U 1437, 1046,1111,0703,4174,4467,0701,4000,0001,0001

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 95
CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ

U 0270, 0100,3440,0505,0174,4156,4700,0400,0000,0000

U 0271, 0762,3770,0505,0174,4007,0520,0400,0000,0000

U 0272, 0762,3770,0505,0174,4007,0621,0400,0000,0000

U 0273, 0762,3770,0505,0174,4007,0421,0400,0000,0000

U 0274, 0764,3440,0505,0174,4007,0700,0400,0000,0000

U 0275, 0764,3770,0505,0174,4007,0520,0400,0000,0000

U 0276, 0764,3770,0505,0174,4007,0621,0400,0000,0000

U 0277, 0764,3770,0505,0174,4007,0421,0400,0000,0000

U 0762, 0110,3443,0100,4174,4156,4700,0200,0014,0012

U 0763, 0110,3441,0301,4170,4156,4700,0200,0014,0012

U 0764, 0110,3441,0301,4170,4156,4700,0200,0014,0012

U 0765, 0110,3443,0100,4174,4156,4700,0200,0014,0012

```
; 3421 .TOC "CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ"  
; 3422 ; ENTER WITH E IN AR  
; 3423  
; 3424 =1000  
; 3425 JUMP-TABLE:  
; 3426  
; 3427 ;CASE 0 -- NEVER  
; 3428 AC_[BR], NEXT INST  
; 3429  
; 3430 ;CASE 1 -- LESS  
; 3431 AC_[BR] TEST, SKIP DPO, J/JUMP-  
; 3432  
; 3433 ;CASE 2 -- EQUAL  
; 3434 AC_[BR] TEST, SKIP AD.EQ.O, J/JUMP-  
; 3435  
; 3436 ;CASE 3 -- LESS THAN OR EQUAL  
; 3437 AC_[BR] TEST, SKIP AD.LE.O, J/JUMP-  
; 3438  
; 3439 ;CASE 4 -- ALWAYS  
; 3440 JUMPA: AC_[BR], J/JUMPA  
; 3441  
; 3442 ;CASE 5 -- GREATER THAN OR EQUAL TO  
; 3443 AC_[BR] TEST, SKIP DPO, J/JUMPA  
; 3444  
; 3445 ;CASE 6 -- NOT EQUAL  
; 3446 AC_[BR] TEST, SKIP AD.EQ.O, J/JUMPA  
; 3447  
; 3448 ;CASE 7 -- GREATER  
; 3449 AC_[BR] TEST, SKIP AD.LE.O, J/JUMPA  
; 3450  
; 3451 =0  
; 3452 JUMP-: DONE  
; 3453 JUMPA  
; 3454  
; 3455 =0  
; 3456 JUMPA: JUMPA  
; 3457 DONE  
; 3458  
; 3459
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 96
CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ

D 0320, 0000,1400,2100 ; 3460
D 0321, 0001,1440,2100 ; 3461 320: .DCODE
D 0322, 0002,1440,2100 ; 3462 I, SJC-, J/DONE
D 0323, 0003,1440,2100 ; 3463 I, SJCL, J/JUMP
D 0324, 0004,1520,2100 ; 3464 I, SJCE, J/JUMP
D 0325, 0005,1440,2100 ; 3465 I, SJCLE, J/JUMP
D 0326, 0006,1440,2100 ; 3466 I, SJCA, J/JRST
D 0327, 0007,1440,2100 ; 3467 I, SJCGE, J/JUMP
; 3468 I, SJCN, J/JUMP
; 3469 I, SJCG, J/JUMP
; 3470 .UCODE

U 1440, 0270,3771,0005,0276,6003,7700,0000,0000 ; 3470
; 3471 1440:
; 3472 JUMP: [BR]_AC,JUMP DISP
; 3473
; 3474 .DCODE

D 0340, 0000,1611,3000 ; 3475 340: I-PF, SJC-, J/AOJ
D 0341, 0001,1611,2100 ; 3476 I, SJCL, J/AOJ
D 0342, 0002,1611,2100 ; 3477 I, SJCE, J/AOJ
D 0343, 0003,1611,2100 ; 3478 I, SJCLE, J/AOJ
D 0344, 0004,1611,2100 ; 3479 I, SJCA, J/AOJ
D 0345, 0005,1611,2100 ; 3480 I, SJCGE, J/AOJ
D 0346, 0006,1611,2100 ; 3481 I, SJCN, J/AOJ
D 0347, 0007,1611,2100 ; 3482 I, SJCG, J/AOJ
; 3483 .UCODE
; 3484
; 3485 1611:
; 3486 AOJ: [BR]_AC+1, AD FLAGS, 4T, JUMP DISP
; 3487
; 3488 .DCODE

U 1611, 0270,0551,0705,0274,4463,7702,0000,0001,0001 ; 3489 360: I-PF, SJC-, J/SOJ
; 3490 I, SJCL, J/SOJ
; 3491 I, SJCE, J/SOJ
; 3492 I, SJCLE, J/SOJ
; 3493 I, SJCA, J/SOJ
; 3494 I, SJCGE, J/SOJ
; 3495 I, SJCN, J/SOJ
; 3496 I, SJCG, J/SOJ
; 3497 .UCODE
; 3498
; 3499 1542:
; 3500 SOJ: [BR]_AC-1, AD FLAGS, 4T, JUMP DISP
; 3501
; 3502 .DCODE

D 0360, 0000,1542,3000 ; 3503 252: I, SJCGE, J/AOBJ
D 0361, 0001,1542,2100 ; 3504 I, SJCL, J/AOBJ
D 0362, 0002,1542,2100 ; 3505 .UCODE
D 0363, 0003,1542,2100 ; 3506
D 0364, 0004,1542,2100 ; 3507
D 0365, 0005,1542,2100 ; 3508 1547:
D 0366, 0006,1542,2100 ; 3509 AOBJ: [BR]_AC+1000001, ;ADD 1 TO BOTH HALF WORDS
D 0367, 0007,1542,2100 ; 3510 INH CRY18, 3T, ;NO CARRY INTO LEFT HALF
; 3511 JUMP DISP ;HANDLE EITHER AOBJP OR AOBJN

U 1542, 0270,2551,0705,0274,4463,7702,4000,0001,0001 ; 3511

D 0252, 0005,1547,2100 ; 3511

U 1547, 0270,0551,1505,0274,4403,7701,0000,0000,0000 ; 3511

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 97
AC DECODE JUMPS -- JRST, JFCL

```

; 3512 .TOC "AC DECODE JUMPS -- JRST, JFCL"
; 3513
; 3514 .DCODE
D 0254, 0000, 1520, 6000 ; 3515 254: I,VMA/O, AC DISP, J/JRST ;DISPATCHES TO 1 OF 16
; 3516 ; PLACES ON AC BITS
D 0255, 0000, 1540, 2100 ; 3517 I, J/JFCL
; 3518 .UCODE
; 3519
; 3520 ;JRST DISPATCHES TO ONE OF 16 LOC'NS ON AC BITS
; 3521
; 3522 =0000
; 3523 1520:
U 1520, 0110, 3441, 0301, 4170, 4156, 4700, 0200, 0014, 0012 ; 3524 JRST: JUMPA ;(0) JRST O.
U 1521, 0110, 3441, 0301, 4170, 4156, 4700, 0200, 0014, 0012 ; 3525 1521: JUMPA ;(1) PORTAL IS SAME AS JRST
; 3526 1522: VMA [PC]-1, START READ, ;(2) JRSTF
U 1522, 0150, 1113, 0701, 4170, 4007, 0700, 4200, 0004, 0012 ; 3527 J/JRSTF
U 1523, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3528 1523: UUO ;(3)
U 1524, 1010, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000 ; 3529 1524: SKIP KERNEL, J/HALT ;(4) HALT
; 3530 1525:
; 3531 XJRSTFO: VMA [AR], START READ, ;(5) XJRSTF
U 1525, 2215, 3443, 0300, 4174, 4007, 0700, 0200, 0004, 0012 ; 3532 J/XJRSTF
U 1526, 0320, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000 ; 3533 1526: SKIP KERNEL, J/XJEN ;(6) XJEN
U 1527, 1024, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000 ; 3534 1527: SKIP KERNEL, J/XPCW ;(7) XPCW
; 3535 1530: VMA [PC]-1, START READ, ;(10)
U 1530, 1014, 1113, 0701, 4170, 4007, 0040, 4200, 0004, 0012 ; 3536 SKIP IO LEGAL, J/JRST10
U 1531, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3537 1531: UUO ;(11)
; 3538 1532: VMA [PC]-1, START READ, ;(12) JEN
U 1532, 0300, 1113, 0701, 4170, 4007, 0040, 4200, 0004, 0012 ; 3539 SKIP IO LEGAL, J/JEN
U 1533, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3540 1533: UUO ;(13)
U 1534, 1034, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000 ; 3541 1534: SKIP KERNEL, J/SFM ;(14) SFM
U 1535, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3542 1535: UUO ;(15)
U 1536, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3543 1536: UUO ;(16)
U 1537, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740 ; 3544 1537: UUO ;(17)
; 3545
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 98
AC DECODE JUMPS -- JRST, JFCL

; 3546
; 3547
; 3548
; 3549
U 0150, 1064,3771,0002,4365,5217,0700,0210,0000,0002
U 0152, 0110,3441,0301,4170,4156,4700,0200,0014,0012
; 3550
; 3551
; 3552
U 1064, 0030,4443,0000,2174,4006,6700,0000,0000,0000
; 3553
; 3554
; 3555
; 3556
; 3557
U 0030, 0002,3773,0000,2274,4464,1700,0000,0001,0004
; 3558
; 3559
; 3560
U 0032, 0002,3333,0002,4174,4464,1700,0000,0001,0004
; 3561
; 3562
; 3563
; 3564
; 3565
U 0034, 1117,0551,0202,2270,4007,0700,0200,0004,0012
; 3566
; 3567
; 3568
; 3569
; 3570
U 0036, 1117,3443,0200,4174,4007,0700,0200,0004,0112
; 3571
; 3572
; 3573
U 1117, 1064,3771,0002,4365,5217,0700,0200,0000,0002
; 3574
; 3575

=0*
JRSTF: MEM READ, ;WAIT FOR DATA
[HR]_MEM, ;STICK IN HR
LOAD INST EA, ;LOAD @ AND XR
CALL [JRSTO] ;COMPUTE EA AGAIN
JUMPA ;JUMP
JRSTO: EA MODE DISP ;WHAT TYPE OF EA?
=100*
READ XR, ;INDEXED
LOAD FLAGS, ;GET FLAGS FROM XR
UPDATE USER, ;ALLOW USER TO SET
RETURN [2] ;ALL DONE
READ [HR], ;PLAIN
LOAD FLAGS, ;LOAD FLAGS FROM INST
UPDATE USER, ;ALLOW USER TO SET
RETURN [2] ;RETURN
[HR]_[HR]+XR, ;BOTH
LOAD VMA, ;FETCH IND WORD
START READ, ;START MEM CYCLE
J/JRST1 ;CONTINUE BELOW
VMA_[HR], ;INDIRECT
START READ, ;FETCH IND WORD
PXCT EA, ;SETUP PXCT STUFF
J/JRST1 ;CONTINUE BELOW
JRST1: MEM READ, ;WAIT FOR DATA
[HR]_MEM, ;LOAD THE HR
LOAD INST EA, ;LOAD @ AND XR
J/JRSTO ;LOOP BACK

```

; 3576 =0
U 1010, 2621,4551,0202,4374,0007,0700,0000,0077,7740 ; 3577 HALT: UUO ;USER MODE
U 1011, 2211,3441,0301,4174,4007,0700,0000,0000,0000 ; 3578 [PC]_[AR] ;EXEC MODE--CHANGE PC
U 2211, 0104,4751,1217,4374,4007,0700,0000,0000,0001 ; 3579 HALT [HALT] ;HALT INSTRUCTION
; 3580
; 3581 =0
U 1014, 2621,4551,0202,4374,0007,0700,0000,0077,7740 ; 3582 JRST10: UUO
U 1015, 0303,4443,0000,4174,4007,0700,0000,0000,0000 ; 3583 J/JEN2 ;DISMISS INTERRUPT
; 3584 =0000
U 0300, 2621,4551,0202,4374,0007,0700,0000,0077,7740 ; 3585 JEN: UUO ; FLAGS
; 3586 MEM READ,
; 3587 [HR]_MEM, ;GET INST
; 3588 LOAD INST EA, ;LOAD XR & @
; 3589 CALL [JRSTO] ;COMPUTE FLAGS
; 3590 =0011
U 0303, 2416,4553,1400,4374,4007,0331,0010,0007,7400 ; 3591 JEN2: DISMISS ;DISMISS INTERRUPT
U 0307, 3510,3770,1416,4344,4007,0700,0010,0000,0000 ; 3592 =0111 CALL LOAD PI ;RELOAD PI HARDWARE
U 0317, 0110,3441,0301,4170,4156,4700,0200,0014,0012 ; 3593 =1111 JUMPA ;GO JUMP
; 3594 =
; 3595
; 3596 1540:
; 3597 JFCL: JFCL FLAGS, ;ALL DONE IN HARDWARE
; 3598 SKIP JFCL, ;SEE IF SKIPS
; 3599 3T, ;ALLOW TIME
U 1540, 0762,4443,0000,4174,4467,0551,0000,0001,0010 ; 3600 J/JUMP- ;JUMP IF WE SHOULD
; 3601

```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 100
EXTENDED ADDRESSING INSTRUCTIONS

```
; 3602 .TOC "EXTENDED ADDRESSING INSTRUCTIONS"
; 3603
; 3604 =0000
; 3605 XUJEN: UJ0 ;HERE IF USER MODE
; 3606 DISMISS ;CLEAR HIGHEST INTERRUPT
; 3607 =0101 READ [MASK], LOAD PI ;NO MORE INTERRUPTS
; 3608 =1101 ABORT MEM CYCLE, ;AVOID INTERRUPT PAGE FAIL
; 3609 J/XJRSTFO ;START READING FLAG WORD
; 3610 =
; 3611
; 3612 XJRSTF: MEM READ, [BR]_MEM ;PUT FLAGS IN BR
; 3613 [AR]_[AR]+1, ;INCREMENT ADDRESS
; 3614 LOAD VMA, ;PUT RESULT IN VMA
; 3615 START READ ;START MEMORY
; 3616 MEM READ, [PC]_MEM, ;PUT DATA IN PC
; 3617 HOLD LEFT ;IGNORE SECTION NUMBER
; 3618 READ [BR], LOAD FLAGS, ;LOAD NEW FLAGS
; 3619 UPDATE USER ;BUT HOLD USER FLAG
; 3620 PISET: [FLG]_[FLG].AND.NOT.#, ;CLEAR PI CYCLE
; 3621 FLG.PI/1, J/PIEXIT ;RELOAD PI HARDWARE
; 3622 ; INCASE THIS IS AN
; 3623 ; INTERRUPT INSTRUCTION
; 3624
; 3625 =0
; 3626 XPCW: UJ0 ;USER MODE
; 3627 [BR]_FLAGS ;PUT FLAGS IN BR
; 3628 =0*0
; 3629 PIXPCW: VMA [AR], START WRITE, ;STORE FLAGS
; 3630 CALL [STOBR] ;PUT BR IN MEMORY
; 3631 =1*0 VMA [AR]+1, LOAD VMA,
; 3632 START WRITE, ;PREPEARE TO STORE PC
; 3633 CALL [STOPC] ;PUT PC IN MEMORY
; 3634 =1*1 [AR]_[AR]+1, ;DO NEW PC PART
; 3635 START READ, J/XJRSTF
; 3636 =
; 3637
; 3638 =0
; 3639 SFM: UJ0
; 3640 VMA [AR], START WRITE ;STORE FLAGS
; 3641 [AR]_FLAGS, J/STORE ;STORE AND EXIT
; 3642
```

U 0320, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 0321, 2416,4553,1400,4374,4007,0331,0010,0007,7400
U 0325, 0335,3333,0012,4174,4437,0700,0000,0000,0000

U 0335, 1525,4223,0000,4364,4277,0700,0200,0000,0010

U 2215, 2511,3771,0005,4365,5007,0700,0200,0000,0002

U 2511, 2513,0111,0703,4174,4007,0700,0200,0004,0012

U 2513, 2567,3771,0001,4361,5007,0700,0200,0000,0002

U 2567, 2576,3333,0005,4174,4467,0700,0000,0001,0004

U 2576, 0305,5551,1313,4374,4007,0700,0000,0001,0000

U 1024, 2621,4551,0202,4374,0007,0700,0000,0077,7740

U 1025, 0020,4521,1205,4074,4007,0700,0000,0000,0000

U 0020, 3571,3443,0300,4174,4007,0700,0210,0003,0012

U 0024, 3572,0111,0703,4170,4007,0700,0210,0003,0012

U 0025, 2215,0111,0703,4174,4007,0700,0200,0004,0002

U 1034, 2621,4551,0202,4374,0007,0700,0000,0077,7740

U 1035, 2577,3443,0300,4174,4007,0700,0200,0003,0012

U 2577, 0455,4521,1203,4074,4007,0700,0000,0000,0000

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 101
XCT

```

; 3643 .TOC "XCT"
; 3644
; 3645 .DCODE
D 0256, 0000,1541,1100 ; 3646 256: R, J/XCT ;OPERAND FETCHED AS DATA
; 3647 .UCODE
; 3648
; 3649 1541:
U 1541, 1120,4443,0000,4174,4007,0340,0000,0000,0000 ; 3650 XCT: SKIP KERNEL ;SEE IF MAY BE PXCT
; 3651 =0
; 3652 XCT1A: [HR] [AR], ;STUFF INTO HR
; 3653 DBUS/DP, ;PLACE ON DBUS FOR IR
; 3654 LOAD INST, ;LOAD IR, AC, XR, ETC.
; 3655 PXCT/E1, ;ALLOW XR TO BE PREVIOUS
U 1120, 2600,3441,0302,4174,4617,0700,0000,0000,0100 ; 3656 J/XCT1 ;CONTINUE BELOW
; 3657
; 3658 READ [HR], ;LOAD PXCT FLAGS
; 3659 LOAD PXCT, ;
; 3660 J/XCT1A ;CONTINUE WITH NORMAL FLOW
U 1121, 1120,3333,0002,4174,4167,0700,0000,0000,0000 ; 3661
; 3662 XCT1: WORK[YSAVE]_[HR] CLR LH, ;SAVE FOR IO INSTRUCTIONS
; 3663 J/XCT2 ;GO EXECUTE IT
; 3664
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 102
STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ

D 0260, 0000,1544,2100
D 0261, 0002,1543,3100
D 0262, 0002,1545,2100
D 0263, 0000,1546,2100

U 1543, 2601,3771,0005,4365,5007,0700,0200,0000,0002

U 2601, 1122,0551,1504,0274,4407,0311,0200,0003,0712

U 1544, 2601,3741,0105,4074,4467,0700,0000,0005,0000

U 1122, 0220,3333,0005,4175,5003,7701,0200,0000,0002

U 1123, 2602,3333,0005,4175,5007,0701,0200,0000,0002

U 2602, 0220,4443,0000,4174,4463,7700,0000,0001,2000

U 0220, 0221,3441,0301,4174,4007,0700,0200,0014,0012

U 0221, 0100,3440,0404,0174,4156,4700,0400,0000,0000

U 0222, 1400,3440,0404,0174,4007,0700,0400,0000,0000

```
; 3665 .TOC "STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ"
; 3666
; 3667 .DCODE
; 3668 260: I, B/O, J/PUSHJ
; 3669 IR, B/2, J/PUSH
; 3670 I, B/2, J/POP
; 3671 I, J/POPJ
; 3672 .UCODE
; 3673
; 3674 ;ALL START WITH E IN AR
; 3675 1543:
; 3676 PUSH: MEM READ, ;PUT MEMOP IN BR
; 3677 [BR] MEM ;
; 3678 PUSH1: [ARX]_AC+1000001, ;BUMP BOTH HALVES OF AC
; 3679 INH CRY18, ;NO CARRY
; 3680 LOAD VMA, ;START TO STORE ITEM
; 3681 START WRITE, ;START MEM CYCLE
; 3682 PXCT STACK WORD, ;THIS IS THE STACK DATA WORD
; 3683 3T, ;ALLOW TIME
; 3684 SKIP CRYO, ;GO TO STMAC, SKIP IF PDL OV
; 3685 J/STMAC ;
; 3686
; 3687 1544:
; 3688 PUSHJ: [BR]_PC WITH FLAGS, ;COMPUTE UPDATED FLAGS
; 3689 CLR FPD, ;CLEAR FIRST-PART-DONE
; 3690 J/PUSH1 ; AND JOIN PUSH CODE
; 3691
; 3692 =0
; 3693 STMAC: MEM WRITE, ;WAIT FOR MEMORY
; 3694 MEM [BR], ;STORE BR ON STACK
; 3695 B DISP, ;SEE IF PUSH OR PUSHJ
; 3696 J/JSTAC ;BELOW
; 3697 ;WE MUST STORE THE STACK WORD PRIOR TO SETTING PDL OV IN CASE OF
; 3698 ; PAGE FAIL.
; 3699 MEM WRITE, ;WAIT FOR MEMORY
; 3700 MEM [BR] ;STORE BR
; 3701 SETPDL: SET PDL OV, ;OVERFLOW
; 3702 B DISP, ;SEE IF PUSH OR PUSHJ
; 3703 J/JSTAC ;BELOW
; 3704
; 3705 =00
; 3706 JSTAC: [PC]_[AR], ;PUSHJ--LOAD PC
; 3707 LOAD VMA, ;LOAD ADDRESS
; 3708 FETCH ;GET NEXT INST
; 3709 JSTAC1: AC [ARX], ;STORE BACK STACK PTR
; 3710 NEXT INST ;DO NEXT INST
; 3711 AC [ARX], ;UPDATE STACK POINTER
; 3712 J/DONE ;DO NEXT INST
; 3713 =
; 3714
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 103
STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ

```
U 1545, 2603,3771,0004,0276,6007,0701,0200,0004,0712
; 3715 1545:
; 3716 POP: [ARX]_AC, ;GET POINTER
; 3717 LOAD VMA, ;ADDRESS OF STACK WORD
; 3718 START READ, 3T, ;START CYCLE
; 3719 PXCT STACK WORD ;FOR PXCT
; 3720
; 3721 MEM READ, ;LOAD BR (QUIT IF PAGE FAIL)
; 3722 [BR]_MEM ;STACK WORD TO BR
; 3723
; 3724 [ARX]_[ARX]+#, ;UPDATE POINTER
; 3725 #/777777, ;-1 IN EACH HALF
; 3726 INH CRY18, 3T, ;BUT NO CARRY
; 3727 SKIP CRYO ;SEE IF OVERFLOW
; 3728
; 3729 =0 VMA [AR], ;EFFECTIVE ADDRESS
; 3730 PXCT DATA, ;FOR PXCT
; 3731 START WRITE, ;WHERE TO STORE RESULT
; 3732 J/POPX1 ;OVERFLOW
; 3733
; 3734 VMA [AR], ;EFFECTIVE ADDRESS
; 3735 PXCT DATA, ;FOR PXCT
; 3736 START WRITE ;WHERE TO STORE RESULT
; 3737
; 3738 MEM WRITE, ;WAIT FOR MEM
; 3739 MEM [BR], ;STORE BR
; 3740 B DISP, ;POP OR POPJ?
; 3741 J/JSTAC ;STORE POINTER
; 3742
; 3743
; 3744 POPX1: MEM WRITE, ;WAIT FOR MEMORY
; 3745 MEM [BR], ;STORE BR
; 3746 J/SETPDL ;GO SET PDL OV
; 3747
; 3748 1546:
; 3749 POPJ: [ARX]_AC, ;GET POINTER
; 3750 LOAD VMA, ;POINT TO STACK WORD
; 3751 PXCT STACK WORD, 3T, ;FOR PXCT
; 3752 START READ ;START READ
; 3753 [ARX]_[ARX]+#, ;UPDATE POINTER
; 3754 #/777777, ;-1 IN BOTH HALFS
; 3755 INH CRY18, 3T, ;INHIBIT CARRY 18
; 3756 SKIP CRYO ;SEE IF OVERFLOW
; 3757 =0 SET PDL OV ;SET OVERFLOW
; 3758 MEM READ, [PC]_MEM, ;STICK DATA IN PC
; 3759 HOLD LEFT, ;NO FLAGS
; 3760 J/JSTAC1 ;STORE POINTER
; 3761
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 104
STACK INSTRUCTIONS -- ADJSP

```
D 0105, 0000,1551,3000
U 1551, 2610,3770,0303,4344,0007,0700,0000,0000,0000
U 2610, 1130,3771,0005,0276,6007,0521,0000,0000,0000
U 1130, 1132,0113,0503,0174,4407,0521,0400,0000,0000
U 1131, 1134,0113,0503,0174,4407,0521,0400,0000,0000
U 1132, 0100,4443,0000,4174,4156,4700,0000,0000,0000
U 1133, 0603,4443,0000,4174,4467,0700,0000,0001,2000
U 1134, 0603,4443,0000,4174,4467,0700,0000,0001,2000
U 1135, 0100,4443,0000,4174,4156,4700,0000,0000,0000

; 3762 .TOC "STACK INSTRUCTIONS -- ADJSP"
; 3763
; 3764 .DCODE
; 3765 105: I-PF, B/O, J/ADJSP
; 3766 .UCODE
; 3767
; 3768
; 3769 1551:
ADJSP: [AR]_[AR] SWAP, ;MAKE 2 COPIES OF RH
; 3770 HOLD RIGHT
; 3771 [BR]_AC, ;READ AC, SEE IF MINUS
; 3772 3T,
; 3773 SKIP DPO
; 3774 =0 AC [BR]+[AR], ;UPDATE AC
; 3775 INH CRY18, ;NO CARRY
; 3776 SKIP DPO, ;SEE IF STILL OK
; 3777 3T, ;ALLOW TIME
; 3778 J/ADJSP1 ;TEST FOR OFLO
; 3779 AC [BR]+[AR], ;UPDATE AC
; 3780 INH CRY18, ;NO CARRY
; 3781 SKIP DPO, ;SEE IF STILL MINUS
; 3782 3T, ;ALLOW TIME FOR SKIP
; 3783 J/ADJSP2 ;CONTINUE BELOW
; 3784
; 3785 =0
; 3786 ADJSP1: NEXT INST ;NO OVERFLOW
; 3787 SET PDL OV, ;SET PDL OV
; 3788 J/NIDISP ;GO DO NICOND DISP
; 3789
; 3790 =0
; 3791 ADJSP2: SET PDL OV, ;SET PDL OV
; 3792 J/NIDISP ;GO DO NICOND DISP
; 3793 NEXT INST ;NO OVERFLOW
; 3794
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
 ; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 105
 SUBROUTINE CALL/RETURN -- JSR, JSP, JSA, JRA

| | | | |
|--|--------|--------|--|
| | ; 3795 | .TOC | "SUBROUTINE CALL/RETURN -- JSR, JSP, JSA, JRA" |
| | ; 3796 | | |
| | ; 3797 | .DCODE | |
| D 0264, 0000,1552,2100 | ; 3798 | 264: | I, J/JSR |
| D 0265, 0000,1550,2100 | ; 3799 | | I, J/JSP |
| D 0266, 0000,1554,2100 | ; 3800 | | I, J/JSA |
| D 0267, 0000,1555,2100 | ; 3801 | | I, J/JRA |
| | ; 3802 | .UCODE | |
| | ; 3803 | | |
| | ; 3804 | 1550: | |
| U 1550, 2611,3741,0105,4074,4007,0700,0000,0000,0000 | ; 3805 | JSR: | [BR]_PC WITH FLAGS ;GET PC WITH FLAGS |
| | ; 3806 | | CLR FPD, ;CLEAR FIRST-PART-DONE |
| | ; 3807 | | AC [BR], ;STORE FLAGS |
| U 2611, 0764,3440,0505,0174,4467,0700,0400,0005,0000 | ; 3808 | | J/JUMPA ;GO JUMP |
| | ; 3809 | | |
| | ; 3810 | 1552: | |
| U 1552, 2612,3741,0105,4074,4467,0700,0000,0005,0000 | ; 3811 | JSR: | [BR]_PC WITH FLAGS, ;GET PC WITH FLAGS |
| | ; 3812 | | CLR FPD ;CLEAR FIRST-PART-DONE |
| U 2612, 2613,3443,0300,4174,4007,0700,0200,0003,0012 | ; 3813 | | VMA [AR], ;EFFECTIVE ADDRESS |
| | ; 3814 | | START WRITE ;STORE OLD PC WORD |
| U 2613, 2614,3333,0005,4175,5007,0701,0200,0000,0002 | ; 3815 | | MEM WRITE, ;WAIT FOR MEMORY |
| | ; 3816 | | MEM [BR], ;STORE |
| | ; 3817 | | [PC]_[AR]+1000001, ;PC E+1 |
| | ; 3818 | | HOLD LEFT, ;NO JUNK IN LEFT |
| | ; 3819 | | 3T, ;ALLOW TIME FOR DBM |
| U 2614, 0107,0551,0301,4370,4007,0701,0000,0000,0001 | ; 3820 | | J/START ;START AT E+1 |
| | ; 3821 | | |
| | ; 3822 | | |
| | ; 3823 | 1554: | |
| U 1554, 2615,3441,0305,4174,4007,0700,0200,0003,0002 | ; 3824 | JSR: | [BR]_[AR], ;SAVE E |
| U 2615, 0060,3770,0304,4344,4007,0700,0000,0000,0000 | ; 3825 | | START WRITE ;START TO STORE |
| | ; 3826 | | [ARX]_[AR] SWAP ;ARX LEFT E |
| | ; 3827 | =0*0 | [AR]_AC, ;GET OLD AC |
| U 0060, 2757,3771,0003,0276,6007,0700,0010,0000,0000 | ; 3828 | | CALL [IBPX] ;SAVE AR IN MEMORY |
| | ; 3829 | =1*0 | [ARX]_[PC], ;ARX NOW HAS E,,PC |
| | ; 3830 | | HOLD LEFT, ; |
| U 0064, 3573,3441,0104,4170,4007,0700,0010,0000,0000 | ; 3831 | | CALL [AC ARX] ;GO PUT ARX IN AC |
| | ; 3832 | =1*1 | [PC]_[BR]+1000001, ;NEW PC |
| | ; 3833 | | 3T, ;ALLOW TIME |
| | ; 3834 | | HOLD LEFT, ;NO JUNK IN PC LEFT |
| U 0065, 0107,0551,0501,4370,4007,0701,0000,0000,0001 | ; 3835 | | J/START ;START AT E+1 |
| | ; 3836 | | |
| | ; 3837 | | |
| | ; 3838 | 1555: | |
| U 1555, 2616,3771,0005,0276,6007,0700,0000,0000,0000 | ; 3839 | JRA: | [BR]_AC ;GET AC |
| U 2616, 2617,3770,0505,4344,4007,0700,0000,0000,0000 | ; 3840 | | [BR]_[BR] SWAP ;OLD E IN BR RIGHT |
| | ; 3841 | | VMA [BR], ;LOAD VMA |
| U 2617, 2620,3443,0500,4174,4007,0700,0200,0004,0012 | ; 3842 | | START READ ;FETCH SAVED AC |
| | ; 3843 | | MEM READ, ;WAIT FOR MEMORY |
| | ; 3844 | | [BR]_MEM, ;LOAD BR WITH SAVE AC |
| U 2620, 0274,3771,0005,4365,5007,0700,0200,0000,0002 | ; 3845 | | J/JUMPA ;GO JUMP |
| | ; 3846 | | |

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 106
ILLEGAL INSTRUCTIONS AND UOO'S

```
; 3847 .TOC "ILLEGAL INSTRUCTIONS AND UOO'S"  
; 3848 ;LUOO'S TRAP TO CURRENT CONTEXT  
; 3849  
; 3850 .DCODE  
; 3851 O30: I, B/0, J/LUOO  
; 3852 I, B/1, J/LUOO  
; 3853 I, B/2, J/LUOO  
; 3854 I, B/3, J/LUOO  
; 3855 I, B/4, J/LUOO  
; 3856 I, B/5, J/LUOO  
; 3857 I, B/6, J/LUOO  
; 3858 I, B/7, J/LUOO  
; 3859  
; 3860 ;MONITOR UOO'S -- TRAP TO EXEC  
; 3861  
; 3862 O40: I, J/MUOO ;CALL  
; 3863 I, J/MUOO ;INIT  
; 3864 I, J/MUOO  
; 3865 I, J/MUOO  
; 3866 I, J/MUOO  
; 3867 I, J/MUOO  
; 3868 I, J/MUOO  
; 3869 I, J/MUOO ;CALLI  
; 3870 I, J/MUOO ;OPEN  
; 3871 I, J/MUOO ;TTCALL  
; 3872 I, J/MUOO  
; 3873 I, J/MUOO  
; 3874 I, J/MUOO  
; 3875 I, J/MUOO ;RENAME  
; 3876 I, J/MUOO ;IN  
; 3877 I, J/MUOO ;OUT  
; 3878 I, J/MUOO ;SETSTS  
; 3879 I, J/MUOO ;STATO  
; 3880 I, J/MUOO ;GETSTS  
; 3881 I, J/MUOO ;STATZ  
; 3882 I, J/MUOO ;INBUF  
; 3883 I, J/MUOO ;OUTBUF  
; 3884 I, J/MUOO ;INPUT  
; 3885 I, J/MUOO ;OUTPUT  
; 3886 I, J/MUOO ;CLOSE  
; 3887 I, J/MUOO ;RELEAS  
; 3888 I, J/MUOO ;MTAPE  
; 3889 I, J/MUOO ;UGETF  
; 3890 I, J/MUOO ;USETI  
; 3891 I, J/MUOO ;USETO  
; 3892 I, J/MUOO ;LOOKUP  
; 3893 I, J/MUOO ;ENTER  
; 3894  
; 3895 ;EXPANSION OPCODES  
; 3896  
; 3897 100: I, J/UOO ;UJEN  
; 3898 I, J/UOO101  
; 3899 I, J/UOO102 ;GFAD  
; 3900 I, J/UOO103 ;GFSB  
; 3901
```

D 0030, 0000,1557,2100
D 0031, 0001,1557,2100
D 0032, 0002,1557,2100
D 0033, 0003,1557,2100
D 0034, 0004,1557,2100
D 0035, 0005,1557,2100
D 0036, 0006,1557,2100
D 0037, 0007,1557,2100

D 0040, 0000,1556,2100
D 0041, 0000,1556,2100
D 0042, 0000,1556,2100
D 0043, 0000,1556,2100
D 0044, 0000,1556,2100
D 0045, 0000,1556,2100
D 0046, 0000,1556,2100
D 0047, 0000,1556,2100
D 0050, 0000,1556,2100
D 0051, 0000,1556,2100
D 0052, 0000,1556,2100
D 0053, 0000,1556,2100
D 0054, 0000,1556,2100
D 0055, 0000,1556,2100
D 0056, 0000,1556,2100
D 0057, 0000,1556,2100
D 0060, 0000,1556,2100
D 0061, 0000,1556,2100
D 0062, 0000,1556,2100
D 0063, 0000,1556,2100
D 0064, 0000,1556,2100
D 0065, 0000,1556,2100
D 0066, 0000,1556,2100
D 0067, 0000,1556,2100
D 0070, 0000,1556,2100
D 0071, 0000,1556,2100
D 0072, 0000,1556,2100
D 0073, 0000,1556,2100
D 0074, 0000,1556,2100
D 0075, 0000,1556,2100
D 0076, 0000,1556,2100
D 0077, 0000,1556,2100

D 0100, 0000,1556,2100
D 0101, 0000,1661,2100
D 0102, 0000,1662,2100
D 0103, 0000,1663,2100

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 107
ILLEGAL INSTRUCTIONS AND UUU'S

```

; 3902 ;RESERVED OPCODES
; 3903
; 3904 000: I, J/UUU
; 3905 104: I, J/JSYS ;JSYS
; 3906 106: I, J/UUU106 ;GFMP
; 3907 I, J/UUU107 ;GFDV
; 3908 130: I, B/0, J/FP-LONG ;UFA
; 3909 I, B/1, J/FP-LONG ;DFN
; 3910 141: I, B/2, J/FP-LONG ;FADL
; 3911 151: I, B/3, J/FP-LONG ;FSBL
; 3912 161: I, B/4, J/FP-LONG ;FMPL
; 3913 171: I, B/5, J/FP-LONG ;FDVL
; 3914 247: I, J/UUU247 ;RESERVED
; 3915 .UCODE
; 3916
; 3917 1661:
; 3918 UUU101: UUU
; 3919 1662:
; 3920 UUU102: UUU
; 3921 1663:
; 3922 UUU103: UUU
; 3923 1664:
; 3924 JSYS: UUU
; 3925 1666:
; 3926 UUU106: UUU
; 3927 1667:
; 3928 UUU107: UUU
; 3929 1660:
; 3930 FP-LONG:UUU
; 3931 1665:
; 3932 UUU247: UUU
; 3933

D 0000, 0000,1556,2100
D 0104, 0000,1664,2100
D 0106, 0000,1666,2100
D 0107, 0000,1667,2100
D 0130, 0000,1660,2100
D 0131, 0001,1660,2100
D 0141, 0002,1660,2100
D 0151, 0003,1660,2100
D 0161, 0004,1660,2100
D 0171, 0005,1660,2100
D 0247, 0000,1665,2100

U 1661, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1662, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1663, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1664, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1666, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1667, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1660, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 1665, 2621,4551,0202,4374,0007,0700,0000,0077,7740
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 108
ILLEGAL INSTRUCTIONS AND UUU'S

```
U 1556, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 2621, 1136,4751,1204,4374,4007,0700,0000,0000,0424
U 1136, 3670,0111,1104,4174,4007,0700,0010,0000,0000

ED

U 1137, 0310,3333,0004,4174,4007,0700,0200,0021,1016

U 0022, 3443,3333,0002,4175,5007,0701,0210,0000,0002
U 0023, 3571,0111,0704,4170,4007,0700,0210,0023,1016

; 3934 ;HERE FOR UUU'S WHICH TRAP TO EXEC
; 3935 1556:
; 3936 UUU: ;THIS TAG IS USED FOR ILLEGAL THINGS WHICH DO UUU TRAPS
; 3937 MUUU: ;THIS TAG IS USED FOR MONITOR CALL INSTRUCTIONS
; 3938 [HR]_[HR].AND.#, ;MASK OUT @ AND XR
; 3939 #/777740, ;MASK
; 3940 HOLD RIGHT ;KEEP RIGHT
; 3941 ;THE UUU MACRO DOES THE ABOVE INSTRUCTION AND GOES TO UUOGO
; 3942 UUOGO: [ARX]_O XWD [424] ;HERE FROM UUU MACRO
; 3943 ;GET OFFSET TO UPT
; 3944 =O [ARX]_[ARX]+[UBR], ;ADDRESS OF MUUU WORD
; 3945 CALL [ABORT] ;STOP MEMORY
; 3946 .IF/KIPAGE
; 3947 .IF/KLPAGE
; 3948 READ [EBR], ;IF BOTH POSSIBLE, SEE WHICH IS ENAB

; 3949 SKIP DPO ;KL PAGING ??
; 3950 =O
; 3951 .ENDIF/KLPAGE
; 3952 READ [ARX], ;GET THE ADDRESS
; 3953 LOAD VMA, ;START WRITE
; 3954 VMA PHYSICAL WRITE, ;ABSOLUTE ADDRESS
; 3955 J/KIMUUO ;GO STORE KI STYLE
; 3956 .ENDIF/KIPAGE
; 3957 .IF/KLPAGE
; 3958 [AR]_[HR] SWAP ;PUT IN RIGHT HALF
; 3959 =O [AR]_FLAGS, ;FLAGS IN LEFT HALF
; 3960 HOLD RIGHT, ;JUST WANT FLAGS
; 3961 CALL [UUOFLG] ;CLEAR TRAP FLAGS
; 3962 READ [ARX], ;LOOK AT ADDRESS
; 3963 LOAD VMA, ;LOAD THE VMA
; 3964 VMA PHYSICAL WRITE ;STORE FLAG WORD
; 3965 =O* MEM WRITE, ;WAIT FOR MEMORY
; 3966 MEM_[AR], CALL [NEXT] ;STORE
; 3967 MEM WRITE, ;WAIT FOR MEMORY
; 3968 MEM [PC] ;STORE FULL WORD PC
; 3969 =OOO [HR]_O, ;SAVE E
; 3970 HOLD RIGHT, CALL [NEXT] ;BUT CLEAR OPCODE
; 3971 .ENDIF/KLPAGE
; 3972 =O10
; 3973 UUOPCW: MEM WRITE, ;WAIT FOR MEMORY
; 3974 MEM_[HR], ;STORE INSTRUCTION IN KI
; 3975 ; OR FULL WORD E IN KL
; 3976 CALL [GETPCW] ;GET PROCESS-CONTEXT-WORD
; 3977
; 3978 =O11 NEXT [ARX] PHYSICAL WRITE, ;POINT TO NEXT WORD
; 3979 CALL [STOBR] ;STORE PROCESS CONTEXT WORD
; 3980
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 109
ILLEGAL INSTRUCTIONS AND UUD'S

```
U 0027, 2622,4751,1205,4374,4007,0700,0000,0000,0430 ; 3981 ;NOW WE MUST PICK ONE OF 8 NEW PC WORDS BASED ON PC FLAGS
; 3982 =111 [BR]_O XWD [430] ;OFFSET INTO UPT
; 3983 =
U 2622, 2623,0111,1105,4174,4007,0700,0000,0000,0000 ; 3984 [BR]_[BR]+[UBR] ;ADDRESS OF WORD
U 2623, 2624,4521,1203,4074,4007,0700,0000,0000,0000 ; 3985 [AR]_FLAGS ;GET FLAGS
; 3986 TL [AR], ;LOOK AT FLAGS
; 3987 #/600 ;TRAP SET?
U 2624, 1140,4553,0300,4374,4007,0321,0000,0000,0600 ; 3988 =0 [BR]_[BR].OR.#, ;YES--POINT TO TRAP CASE
; 3989 #/1, ;...
U 1140, 1141,3551,0505,4370,4007,0700,0000,0000,0001 ; 3990 HOLD LEFT ;LEAVE LEFT ALONE
; 3991 TL [AR], ;USER OR EXEC
U 1141, 1142,4553,0300,4374,4007,0321,0000,0001,0000 ; 3992 #/10000 ;...
; 3993 =0 [BR]_[BR].OR.#, ;USER
; 3994 #/4, ;POINT TO USER WORDS
U 1142, 1143,3551,0505,4370,4007,0700,0000,0000,0004 ; 3995 HOLD LEFT
; 3996 READ [BR], ;LOOK AT ADDRESS
; 3997 LOAD VMA, ;PLACE IN VMA
; 3998 VMA PHYSICAL, ;PHYSICAL ADDRESS
U 1143, 2625,3333,0005,4174,4007,0700,0200,0024,1016 ; 3999 START READ ;GET NEW PC WORD
U 2625, 2626,3771,0003,4365,5007,0700,0200,0000,0002 ; 4000 GOEXEC: MEM READ, ;WAIT FOR DATA
; 4001 [AR] MEM ;STICK IN AR
; 4002 READ [AR], ;LOOK AT DATA
; 4003 LOAD FLAGS, ;LOAD NEW FLAGS
; 4004 LEAVE USER, ;ALLOW USER TO LOAD
; 4005 LOAD PCU, ;SET PCU FROM USER
U 2626, 0764,3333,0003,4174,4467,0700,0000,0000,0404 ; 4006 J/JUMPA ;JUMP
; 4007
; 4008 .IF/KIPAGE
; 4009 ;HERE FOR TOPS-10 STYLE PAGING
; 4010
; 4011 =00
; 4012 KIMUUD: MEM WRITE, ;STORE INSTRUCTION
; 4013 MEM [HR], CALL [NEXT] ;IN MEMORY
U 0310, 2630,3333,0002,4175,5007,0701,0210,0000,0002 ; 4014 =10 [AR]_PC WITH FLAGS, ;GET PC WORD
U 0312, 2627,3741,0103,4074,4007,0700,0010,0000,0000 ; 4015 CALL [UUOFLG] ;CLEAR TRAP FLAGS
; 4016 =11 MEM WRITE, ;STORE PC WORD
; 4017 MEM [AR], ;...
U 0313, 0022,3333,0003,4175,5007,0701,0200,0000,0002 ; 4018 J/UUOPCW ;GO STORE PROCESS CONTEXT
; 4019 .ENDIF/KIPAGE
; 4020
; 4021 UUOFLG: [AR]_[AR].AND.NOT.#, ;CLEAR TRAP FLAGS
; 4022 #/600, HOLD RIGHT, ; IN WORD TO SAVE
U 2627, 0001,5551,0303,4374,0004,1700,0000,0000,0600 ; 4023 RETURN [1] ; BACK TO CALLER
; 4024
; 4025 NEXT: NEXT [ARX] PHYSICAL WRITE, ;POINT TO NEXT WORD
U 2630, 0002,0111,0704,4170,4004,1700,0200,0023,1016 ; 4026 RETURN [2]
; 4027
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 110
ILLEGAL INSTRUCTIONS AND UUO'S

```
U 1557, 0400,4751,1203,4374,4007,0700,0000,0000,0040
; 4028 ;HERE FOR LUUO'S
; 4029 1557:
; 4030 LUUO: [AR]_O XWD [40] ;AR GET CONSTANT 40
; 4031 ;THE LUUO MACRO DOES THE ABOVE INSTRUCTION AND GOES TO LUUO1
; 4032 400: ;FOR SIMULATOR
; 4033 LUUO1: READ [AR], ;LOAD 40 INTO
; 4034 LOAD VMA, ; THE VMA AND
; 4035 START WRITE ; PREPARE TO STORE
; 4036 [HR]_[HR].AND.#, ;CLEAR OUT INDEX AND @
; 4037 #/777740, ; ..
; 4038 HOLD RIGHT
; 4039 MEM WRITE, ;STORE LUUO IN 40
; 4040 MEM [HR]
; 4041 VMA_[AR]+1, ;POINT TO 41
; 4042 LOAD VMA, ;PUT 41 IN VMA
; 4043 START READ, ;START FETCH
; 4044 J/CONT1 ;GO EXECUTE THE INSTRUCTION
; 4045
```

; TIOKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 111
ARITHMETIC -- ADD, SUB

```

; 4046 .TOC "ARITHMETIC -- ADD, SUB"
; 4047
; 4048 .DCODE
D 0270, 1015,1560,1100 ; 4049 270: R-PF, AC, J/ADD
D 0271, 0015,1560,3000 ; 4050 I-PF, AC, J/ADD
D 0272, 0016,1560,1700 ; 4051 RW, M, J/ADD
D 0273, 0017,1560,1700 ; 4052 RW, B, J/ADD
; 4053 .UCODE
; 4054
; 4055 1560:
U 1560, 1500,0551,0303,0274,4463,7701,0200,0001,0001 ; 4056 ADD: [AR]_[AR]+AC, ;DO THE ADD
; 4057 AD FLAGS EXIT, 3T ;UPDATE CARRY FLAGS
; 4058 ;STORE ANSWER
; 4059 ;MISSES 3-TICKS BY 3 NS.
; 4060
; 4061
; 4062 .DCODE
D 0274, 1015,1561,1100 ; 4063 274: R-PF, AC, J/SUB
D 0275, 0015,1561,3000 ; 4064 I-PF, AC, J/SUB
D 0276, 0016,1561,1700 ; 4065 RW, M, J/SUB
D 0277, 0017,1561,1700 ; 4066 RW, B, J/SUB
; 4067 .UCODE
; 4068
; 4069 1561:
U 1561, 1500,2551,0303,0274,4463,7701,4200,0001,0001 ; 4070 SUB: [AR]_AC-[AR], ;DO THE SUBTRACT
; 4071 AD FLAGS EXIT, 3T ;UPDATE PC CARRY FLAGS
; 4072 ;ALL DONE
; 4073 ;MISSES 3-TICKS BY 3 NS.
; 4074
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 112
ARITHMETIC -- DADD, DSUB

D 0114, 0205,1457,1100
D 0115, 0205,1615,1100

U 1457, 1144,0551,0404,1274,4007,0562,0000,0000,1441

U 1144, 2635,0551,0303,0274,4467,0702,4000,0001,0001
U 1145, 2634,7441,1205,4174,4007,0700,0000,0000,0000

U 2634, 1144,3111,0503,4170,4007,0700,0000,0000,0000

U 1615, 1146,2551,0404,1274,4007,0562,4000,0000,1441

U 1146, 2635,2551,0303,0274,4467,0702,0000,0001,0001

U 1147, 2635,2551,0303,0274,4467,0702,4000,0001,0001

U 2635, 1150,3770,0303,4174,0007,0520,0000,0000,0000
U 1150, 1404,4551,0404,4374,0007,0700,0000,0037,7777
U 1151, 1404,3551,0404,4374,0007,0700,0000,0040,0000

```
; 4075 .TOC "ARITHMETIC -- DADD, DSUB"
; 4076
; 4077 .DCODE
; 4078 114: DBL R, DAC, J/DADD
; 4079 DBL R, DAC, J/DSUB
; 4080 .UCODE
; 4081
; 4082 1457:
; 4083 DADD: [ARX]_[ARX]+AC[1], 4T, ;ADD LOW WORDS
; 4084 SKIP CRY1 ;SEE IF CARRY TO HIGH WORD
; 4085 =0
; 4086 DADD1: [AR]_[AR]+AC, ;ADD HIGH WORDS
; 4087 ADD .25, ;ADD IN ANY CARRY FROM LOW WORD
; 4088 AD FLAGS, 4T, ;UPDATE PC FLAGS
; 4089 J/CPYSGN ;COPY SIGN TO LOW WORD
; 4090 [BR]_.NOT.[MASK] ;SET BITS 35 AND 36 IN
; 4091 [AR]_[AR].OR.[BR], ; AR SO THAT ADD .25 WILL
; 4092 HOLD LEFT, J/DADD1 ; ADD 1.
; 4093
; 4094 1615:
; 4095 DSUB: [ARX]_AC[1]-[ARX], 4T, ;SUBTRACT LOW WORD
; 4096 SKIP CRY1 ;SEE IF CARRY
; 4097 =0 [AR]_AC-[AR]-.25, ;NO CARRY
; 4098 AD FLAGS, 4T, ;UPDATE PC FLAGS
; 4099 J/CPYSGN ;GO COPY SIGN
; 4100 [AR]_AC-[AR], 4T, ;THERE WAS A CARRY
; 4101 AD FLAGS ;UPDATE CARRY FLAGS
; 4102
; 4103 CPYSGN: FIX [AR] SIGN, SKIP DPO
; 4104 =0 [ARX]_[ARX].AND.#, #/377777, HOLD RIGHT, J/MOVE
; 4105 [ARX]_[ARX].OR.#, #/400000, HOLD RIGHT, J/MOVE
; 4106
; 4107
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 113
ARITHMETIC -- MUL, IMUL

```

; 4108 .TOC "ARITHMETIC -- MUL, IMUL"
; 4109
; 4110 .DCODE
; 4111 220: R-PF, AC, J/IMUL
; 4112 I-PF, AC, J/IMUL
; 4113 RW, M, J/IMUL
; 4114 RW, B, J/IMUL
; 4115 .UCODE
; 4116 1641:
; 4117 IMUL: [BRX]_[AR], AC ;COPY C(E)
; 4118 Q_AC, SC_35. ;GET THE AC
; 4119 =O** [BRX]_[BRX]*.5 LONG, ;SHIFT RIGHT
; 4120 CALL [MULSUB] ;MULTIPLY
; 4121 READ [ARX], SKIP AD.EQ.O ;SEE IF FITS
; 4122 =O [ARX]_[ARX]*2, J/IMUL2 ;NOT ZERO--SHIFT LEFT
; 4123 IMUL1: [AR]_Q, EXIT ;POSITIVE
; 4124
; 4125 IMUL2: [MASK].AND.NOT.[ARX], ;SEE IF ALL SIGN BITS
; 4126 SKIP AD.EQ.O ;
; 4127 =O FIX [ARX] SIGN, ;NOT ALL SIGN BITS
; 4128 SKIP DPO, J/IMUL3 ;GIVE + OR - OVERFLOW
; 4129 [AR]_[MAG].EQV.Q, EXIT ;NEGATIVE
; 4130
; 4131 IMUL3: [AR]_Q, SET AROV, J/MOVE
; 4132 [AR]_[MAG].EQV.Q, SET AROV, J/MOVE
; 4133
; 4134
; 4135 .DCODE
; 4136 224: R-PF, DAC, J/MUL
; 4137 I-PF, DAC, J/MUL
; 4138 RW, M, J/MUL
; 4139 RW, DBL B, J/MUL
; 4140 .UCODE
; 4141
; 4142
; 4143 1571:
; 4144 MUL: Q_[AR], AC ;COPY C(E)
; 4145 [TO]_[AR] ;SAVE FOR OVERFLOW TEST
; 4146 [BRX]_AC, SC_35. ;GET THE AC
; 4147 =O** [BRX]_[BRX]*.5 LONG, ;SHIFT OVER
; 4148 CALL [MULSUB] ;MULTIPLY
; 4149 [AR]_[ARX]*2 ;SHIFT OVER
; 4150 FIX [AR] SIGN, SKIP DPO ;SEE IF NEGATIVE
; 4151 =O [ARX]_[MAG].AND.Q, ;POSITIVE
; 4152 EXIT
; 4153 [TO].AND.[BRX], SKIP DPO ;TRIED TO SQUARE 1BO?
; 4154 =O [ARX]_[MAG].EQV.Q, EXIT ;NO
; 4155 [ARX]_[MAG].EQV.Q, ;YES
; 4156 SET AROV, J/MOVE
; 4157
; 4158

D 0220, 1015,1641,1100
D 0221, 0015,1641,3000
D 0222, 0016,1641,1700
D 0223, 0017,1641,1700

U 1641, 2636,3441,0306,0174,4007,0700,0000,0000,0000
U 2636, 0031,3772,0000,0275,5007,0700,2000,0071,0043

U 0031, 2662,3446,0606,4174,4007,0700,0010,0000,0000
U 0035, 1152,3333,0004,4174,4007,0621,0000,0000,0000
U 1152, 2637,3445,0404,4174,4007,0700,0000,0000,0000
U 1153, 1500,3221,0003,4174,4003,7700,0200,0003,0001

U 2637, 1154,5113,0412,4174,4007,0621,0000,0000,0000

U 1154, 1156,3770,0404,4174,0007,0520,0000,0000,0000
U 1155, 1500,7001,0003,4174,4003,7700,0200,0003,0001

U 1156, 1404,3221,0003,4174,4467,0700,0000,0041,1000
U 1157, 1404,7001,0003,4174,4467,0700,0000,0041,1000

D 0224, 1005,1571,1100
D 0225, 0005,1571,3000
D 0226, 0016,1571,1700
D 0227, 0006,1571,1700

U 1571, 2640,3442,0300,0174,4007,0700,0000,0000,0000
U 2640, 2641,3441,0316,4174,4007,0700,0000,0000,0000
U 2641, 0171,3771,0006,0276,6007,0700,2000,0071,0043

U 0171, 2662,3446,0606,4174,4007,0700,0010,0000,0000
U 0175, 2642,3445,0403,4174,4007,0700,0000,0000,0000
U 2642, 1160,3770,0303,4174,0007,0520,0000,0000,0000

U 1160, 1500,4001,0004,4174,4003,7700,0200,0003,0001
U 1161, 1162,4113,0616,4174,4007,0520,0000,0000,0000
U 1162, 1500,7001,0004,4174,4003,7700,0200,0003,0001

U 1163, 1404,7001,0004,4174,4467,0700,0000,0041,1000
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 114
ARITHMETIC -- DMUL

D 0116, 0205, 1566, 1100

U 1566, 2643, 3447, 0303, 4174, 4007, 0700, 0000, 0000, 0000
U 2643, 2644, 4117, 0004, 4174, 4007, 0700, 0000, 0000, 0000

U 2644, 0120, 3441, 0405, 4174, 4007, 0350, 0000, 0000, 0000

U 0120, 2656, 4557, 0006, 1274, 4007, 0701, 0010, 0000, 1441

U 0121, 2646, 4557, 0004, 1274, 4007, 0701, 0000, 0000, 1442
U 0124, 0543, 3223, 0000, 1174, 4007, 0700, 0400, 0000, 1443

U 0543, 0623, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043
U 0547, 2645, 3441, 0416, 4174, 4007, 0700, 0000, 0000, 0000

U 2645, 0130, 3227, 0004, 1174, 4007, 0700, 0400, 0000, 1442
U 2646, 0130, 3777, 0016, 1276, 6007, 0701, 0000, 0000, 1441

U 0130, 2657, 3777, 0006, 0274, 4007, 0701, 0010, 0000, 0000

U 0131, 2661, 0113, 1616, 1174, 4007, 0701, 0400, 0000, 1441
U 0134, 2647, 3223, 0000, 1174, 4007, 0700, 0400, 0000, 1442

U 2647, 0563, 0111, 1604, 4174, 4007, 0700, 0000, 0000, 0000

U 0563, 0623, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043

U 0567, 2650, 3445, 0404, 4174, 4467, 0700, 0000, 0005, 0000

```
; 4159 .TOC "ARITHMETIC -- DMUL"
; 4160
; 4161 .DCODE
; 4162 116: DBL R, DAC, J/DMUL
; 4163 .UCODE
; 4164
; 4165 .IF/FULL
; 4166 1566:
; 4167 DMUL: [AR]_[AR]*.5 ;SHIFT MEM OPERAND RIGHT
; 4168 [ARX]_([ARX].AND.[MAG])* .5
; 4169 [BR]_[ARX], ;COPY LOW WORD
; 4170 SKIP FPD ;SEE IF FIRST PART DONE
; 4171 ;
; 4172 ; BRX * BR ==> C(E+1) * C(AC+1)
; 4173 ;
; 4174 =000 [BRX]_(AC[1].AND.[MAG])* .5, 3T, ;GET LOW AC
; 4175 CALL [DMULGO] ;START MULTIPLY
; 4176 [ARX]_(AC[2].AND.[MAG])* .5, 3T, ;FIRST PART DONE
; 4177 J/DMUL1 ;GO DO SECOND PART
; 4178 =100 AC[3]_Q ;SALT AWAY 1 WORD OF PRODUCT
; 4179 =
; 4180 ;
; 4181 ; BRX * Q ==> C(E) * C(AC+1)
; 4182 ;
; 4183 =0** Q_[AR], SC 35., ;GO MULT NEXT HUNK
; 4184 CALL [QMUL] ;
; 4185 [TO]_[ARX] ;SAVE PRODUCT
; 4186 AC[2]_Q, [ARX]_Q*.5, ;SAVE PRODUCT
; 4187 J/DMUL2 ;GO DO HIGH HALF
; 4188 DMUL1: [TO]_AC[1]*.5 ;RESTORE TO
; 4189 =0*0
; 4190 ;
; 4191 ; BRX * BR ==> C(AC) * C(E+1)
; 4192 ;
; 4193 DMUL2: [BRX]_AC*.5, ;PREPARE TO DO HIGH HALF
; 4194 CALL [DBLMUL] ;GO DO IT
; 4195 AC[1]_[TO]*2, 3T, ;INTERRUPT, SAVE TO
; 4196 J/DMLINT ;SET FPD AND INTERRUPT
; 4197 AC[2]_Q ;SAVE PRODUCT
; 4198 =
; 4199 [ARX]_[ARX]+[TO] ;PREPARE FOR LAST MUL
; 4200 ;
; 4201 ; BRX * Q ==> C(AC) * C(E)
; 4202 ;
; 4203 =0** Q_[AR], SC 35., ;DO THE LAST MULTIPLY
; 4204 CALL [QMUL] ;GO DO IT
; 4205 [ARX]_[ARX]*2, ;SHIFT BACK
; 4206 CLR FPD ;CLEAR FPD
; 4207
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 115
ARITHMETIC -- DMUL

```
U 2650, 1164,3770,0404,0174,4007,0520,0400,0000,0000 ; 4208
U 1164, 2655,3223,0000,1174,4007,0700,0400,0000,1441 ; 4209
U 1165, 2651,7003,0000,1174,4007,0700,0400,0000,1441 ; 4210
U 2651, 2652,3772,0000,1275,5007,0701,0000,0000,1442 ; 4211
U 2652, 2653,7003,0000,1174,4007,0700,0400,0000,1442 ; 4212
U 2653, 2654,3772,0000,1275,5007,0701,0000,0000,1443 ; 4213
U 2654, 2655,7003,0000,1174,4007,0700,0400,0000,1443 ; 4214
; 4215
U 2655, 1166,3741,0103,4074,4007,0520,0000,0000,0000 ; 4216
U 1166, 0110,3443,0100,4174,4156,4700,0200,0014,0012 ; 4217
U 1167, 1400,4443,0000,4174,4467,0700,0000,0041,1000 ; 4218
; 4219
; 4220
; 4221
U 2656, 2657,4221,0004,4174,4007,0700,0000,0000,0000 ; 4222
U 2657, 2660,3442,0500,4174,4007,0700,2000,0071,0043 ; 4223
U 2660, 0623,3447,0606,4174,4007,0700,0000,0000,0000 ; 4224
; 4225
; 4226
U 0623, 2664,3446,1200,4174,4007,0700,0010,0000,0000 ; 4227
; 4228
U 0627, 0004,0113,0404,4174,4464,1701,0000,0001,0001 ; 4229
; 4230
U 2661, 2565,4443,0000,4174,4467,0700,0000,0003,0000 ; 4231
; 4232
; 4233
; 4234
; 4235
; 4236
; 4237

; 4208 AC [ARX] TEST, SKIP DPO ;PUT BACK INTO AC
; 4209 =0 AC[1]_Q, J/DMTRAP ;POSITIVE
; 4210 AC[1]_[MAG].EQV.Q ;NEGATIVE
; 4211 Q AC[2]
; 4212 AC[2]_[MAG].EQV.Q
; 4213 Q AC[3]
; 4214 AC[3]_[MAG].EQV.Q
; 4215 DMTRAP: [AR]_PC WITH FLAGS, ;LOOK AT FLAGS
; 4216 SKIP DPO ;SEE IF AROV SET?
; 4217 =0 DONE ;NO--ALL DONE
; 4218 SET AROV, J/DONE ;YES--FORCE TRAP 1 ALSO
; 4219
; 4220
; 4221 ;WAYS TO CALL MULTIPLY
; 4222 DMULGO: [ARX]_0 ;CLEAR ARX
; 4223 DBLMUL: Q [BR], SC_35.
; 4224 [BRX]_[BRX]*.5
; 4225 =0**
; 4226 QMULT: Q_Q*.5,
; 4227 CALL [MULTIPLY]
; 4228 [ARX]+[ARX], AD FLAGS, ;TEST FOR OVERFLOW
; 4229 3T, RETURN [4] ;AND RETURN
; 4230
; 4231 DMLINT: SET FPD, J/FIXPC ;SET FPD, BACKUP PC
; 4232 ; INTERRUPT
; 4233 .IFNOT/FULL
; 4234 1566:
; 4235 DMUL: UUD
; 4236 .ENDIF/FULL
; 4237
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 116
ARITHMETIC -- DMUL

U 2662, 2663, 3446, 0606, 4174, 4007, 0700, 0000, 0000, 0000

U 2663, 0122, 4226, 0004, 4174, 4007, 0630, 2000, 0060, 0000

U 2664, 0122, 3446, 1200, 4174, 4007, 0630, 2000, 0060, 0000

```
; 4238 ;MULTIPLY SUBROUTINE
; 4239 ;ENTERED WITH:
; 4240 ;     MULTIPLIER IN Q
; 4241 ;     MULTIPLICAND IN BRX
; 4242 ;RETURNS 4 WITH PRODUCT IN ARX!Q
; 4243
; 4244 MUL STEP      "A/BRX,B/ARX,DEST/Q_Q*.5,ASHC,STEP SC,MUL DISP"
; 4245 MUL FINAL   "A/BRX,B/ARX,DEST/Q_Q*2"
; 4246
; 4247 MULSUB: [BRX]_[BRX]*.5 LONG
; 4248 MULSB1: [ARX]_O*.5 LONG,           ;CLEAR ARX AND SHIFT Q
; 4249         STEP SC,                   ;COUNT FIRST STEP
; 4250         J/MUL+                       ;ENTER LOOP
; 4251
; 4252 ;MULTIPLY SUBROUTINE
; 4253 ;ENTERED WITH:
; 4254 ;     MULTIPLIER IN Q
; 4255 ;     MULTIPLICAND IN BRX
; 4256 ;     PARTIAL PRODUCT IN ARX
; 4257 ;RETURNS 4 WITH Q*BRX+ARX IN ARX!Q
; 4258
; 4259 MULTIPLY:
; 4260         Q_Q*.5,                       ;SHIFT Q
; 4261         STEP SC,                       ;COUNT FIRST STEP
; 4262         J/MUL+                         ;ENTER LOOP
; 4263
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 117
ARITHMETIC -- DMUL

```
U 0122, 0122,3336,0604,4174,4046,2630,2000,0060,0000
; 4264 ;HERE FOR POSITIVE STEPS
; 4265 =010
; 4266 MUL+: AD/B, ;O IN A POSITIVE STEP
; 4267 MUL STEP, ;DON'T ADD
; 4268 J/MUL+ ;SHIFT
; 4269 =011 ;KEEP POSITIVE
; 4270 AD/B, ;DONE
; 4271 MUL FINAL, ;DON'T ADD
; 4272 RETURN [4] ;SHIFT
; 4273 =110 ;SHIFT Q AND RETURN
; 4274 AD/B-A-.25, ADD .25, ;1 IN A POSITIVE STEP
; 4275 MUL STEP, ;SUBTRACT
; 4276 J/MUL- ;SHIFT AND COUNT
; 4277 =111 ;NEGATIVE NOW
; 4278 AD/B-A-.25, ADD .25, ;DONE
; 4279 MUL FINAL, ;SUBTRACT
; 4280 RETURN [4] ;SHIFT
; 4281 ; AND RETURN
; 4282 ;HERE FOR NEGATIVE STEPS
; 4283 =010 ;O IN NEGATIVE STEP
; 4284 MUL-: AD/A+B, ;ADD
; 4285 MUL STEP, ;SHIFT AND COUNT
; 4286 J/MUL+ ;POSITIVE NOW
; 4287 =011 ;DONE
; 4288 AD/A+B, ;ADD
; 4289 MUL FINAL, ;SHIFT
; 4290 RETURN [4] ;FIX Q AND RETURN
; 4291 =110 ;1 IN NEGATIVE STEP
; 4292 AD/B, ;DON'T ADD
; 4293 MUL STEP, ;SHIFT AND COUNT
; 4294 J/MUL- ;STILL NEGATIVE
; 4295 =111 ;DONE
; 4296 AD/B, ;DON'T ADD
; 4297 MUL FINAL, ;SHIFT
; 4298 RETURN [4] ;FIX Q AND RETURN
; 4299
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 118
ARITHMETIC -- DIV, IDIV

```

; 4300 .TOC "ARITHMETIC -- DIV, IDIV"
; 4301
; 4302
; 4303 230: R-PF, DAC, J/IDIV
; 4304 I-PF, DAC, J/IDIV
; 4305 RW, M, J/IDIV
; 4306 RW, DBL B, J/IDIV
; 4307
; 4308 234: R-PF, DAC, J/DIV
; 4309 I-PF, DAC, J/DIV
; 4310 RW, M, J/DIV
; 4311 RW, DBL B, J/DIV
; 4312 .UCODE
; 4313
; 4314 1600:
; 4315 IDIV: [BR]_[AR], AC ;COPY MEMORY OPERAND
; 4316 Q_AC, ;LOAD Q
; 4317 SKIP DPO ;SEE IF MINUS
; 4318 =0 [AR]_O, ;EXTEND + SIGN
; 4319 J/DIV1 ;NOW SAME AS DIV
; 4320 [AR]_-1, ;EXTEND - SIGN
; 4321 J/DIV1 ;SAME AS DIV
; 4322
; 4323 1601:
; 4324 DIV: [BR]_[AR] ;COPY MEM OPERAND
; 4325 [AR]_AC ;GET AC
; 4326 Q_AC[1] ;AND AC+1
; 4327 READ [AR], ;TEST FOR NO DIVIDE
; 4328 SKIP AD.EQ.O
; 4329 =000 .NOT.[AR], ;SEE IF ALL SIGN BITS IN AR
; 4330 SKIP AD.EQ.O, ;
; 4331 J/DIVA ;CONTINUE BELOW
; 4332 =001
; 4333 DIV1: READ [BR], ;SEE IF DIVIDE BY
; 4334 SKIP AD.EQ.O ; ZERO
; 4335 =100
; 4336 DIV2: SC_34, ;NOT ZERO--LOAD STEP COUNT
; 4337 CALL [DIVSUB] ;DIVIDE
; 4338 =101 NO DIVIDE ;DIVIDE BY ZERO
; 4339 =110 [ARX]_[AR], ;COPY REMAINDER
; 4340 J/IMUL1 ;STORE ANSWER
; 4341 =
; 4342
; 4343
; 4344 =0
; 4345 DIVA: [BRX]_[AR], ;HIGH WORD IS NOT SIGNS
; 4346 J/DIVB ;GO TEST FOR NO DIVIDE
; 4347 READ [BR], ;ALL SIGN BITS
; 4348 SKIP AD.EQ.O, ;SEE IF ZERO DIVIDE
; 4349 J/DIV2 ;BACK TO MAIN FLOW
; 4350

D 0230, 1005, 1600, 1100
D 0231, 0005, 1600, 3000
D 0232, 0016, 1600, 1700
D 0233, 0006, 1600, 1700

D 0234, 1005, 1601, 1100
D 0235, 0005, 1601, 3000
D 0236, 0016, 1601, 1700
D 0237, 0006, 1601, 1700

U 1600, 2665, 3441, 0305, 0174, 4007, 0700, 0000, 0000, 0000
U 2665, 1170, 3772, 0000, 0275, 5007, 0520, 0000, 0000, 0000
U 1170, 0161, 4221, 0003, 4174, 4007, 0700, 0000, 0000, 0000
U 1171, 0161, 2441, 0703, 4174, 4007, 0700, 4000, 0000, 0000

U 1601, 2666, 3441, 0305, 4174, 4007, 0700, 0000, 0000, 0000
U 2666, 2667, 3771, 0003, 0276, 6007, 0700, 0000, 0000, 0000
U 2667, 2670, 3772, 0000, 1275, 5007, 0701, 0000, 0000, 1441

U 2670, 0160, 3333, 0003, 4174, 4007, 0621, 0000, 0000, 0000

U 0160, 1172, 7443, 0300, 4174, 4007, 0621, 0000, 0000, 0000

U 0161, 0164, 3333, 0005, 4174, 4007, 0621, 0000, 0000, 0000

U 0164, 0370, 4443, 0000, 4174, 4007, 0700, 2010, 0071, 0042
U 0165, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0051, 1000

U 0166, 1153, 3441, 0304, 4174, 4007, 0700, 0000, 0000, 0000

U 1172, 2671, 3441, 0306, 4174, 4007, 0700, 0000, 0000, 0000

U 1173, 0164, 3333, 0005, 4174, 4007, 0621, 0000, 0000, 0000
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 119
ARITHMETIC -- DIV, IDIV

```
U 2671, 2672,3221,0004,4174,4007,0700,0000,0000,0000 ; 4351 DIVB: [ARX] Q ;MAKE ABS VALUES
; 4352 READ [AR], ;SEE IF +
U 2672, 0330,3333,0003,4174,4007,0520,0000,0000,0000 ; 4353 SKIP DPO
; 4354 =00 READ [BR], ;SEE IF +
; 4355 SKIP DPO,
U 0330, 1174,3333,0005,4174,4007,0520,0000,0000,0000 ; 4356 J/DIVC ;CONTINUE BELOW
; 4357 CLEAR [ARX]0, ;FLUSH DUPLICATE SIGN
U 0331, 2753,4551,0404,4374,0007,0700,0010,0037,7777 ; 4358 CALL [DBLNG1] ;NEGATE AR!ARX
; 4359 =11 READ [BR], ;SEE IF TOO BIG
; 4360 SKIP DPO,
U 0333, 1174,3333,0005,4174,4007,0520,0000,0000,0000 ; 4361 J/DIVC
; 4362 =
; 4363 =0
; 4364 DIVC: [AR]-[BR], ;COMPUTE DIFFERENCE
; 4365 SKIP DPO, ;SEE IF IT GOES
; 4366 3T, ;ALLOW TIME
; 4367 J/NODIV ;TEST
U 1174, 1176,2113,0305,4174,4007,0521,4000,0000,0000 ; 4368 [AR]+[BR],
; 4369 SKIP DPO, ;SAME TEST FOR -VE BR
; 4370 3T,
U 1175, 1176,0113,0305,4174,4007,0521,0000,0000,0000 ; 4371 J/NODIV
; 4372 =0 ;
; 4373 NODIV: NO DIVIDE ;TOO BIG
; 4374 [AR]_[BRX], ;FITS
U 1176, 0603,4443,0000,4174,4467,0700,0000,0051,1000 ; 4375 J/DIV1 ;GO BACK AND DIVIDE
; 4376
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 120
ARITHMETIC -- DDIV

```
D 0117, 0205,1627,1100
U 1627, 2673,4112,0400,4174,4007,0700,0000,0000,0000
U 2673, 1200,3447,0305,4174,4007,0421,0000,0000,0000
U 1200, 1204,3446,0505,4174,4007,0700,0000,0000,0000
U 1201, 1202,3446,0505,4174,4007,0520,0000,0000,0000
U 1202, 1204,4003,0000,4174,4007,0621,0000,0000,0000
U 1203, 2674,4751,1217,4374,4007,0700,0000,0000,0005
U 2674, 2675,3662,0000,4374,0007,0700,0000,0060,0000
U 2675, 2676,2222,0000,4174,4007,0700,4000,0000,0000
U 2676, 2700,2446,0505,4174,4047,0700,0040,0000,0000
U 1204, 2677,3446,0505,4174,4047,0700,0000,0000,0000
U 1205, 0603,4443,0000,4174,4467,0700,0000,0051,1000
U 2677, 2700,4751,1217,4374,4007,0700,0000,0000,0004
U 2700, 2701,3221,0006,0174,4007,0700,0000,0000,0000
U 2701, 0054,3777,0003,0274,4007,0520,0000,0000,0000
U 0054, 1206,4552,0000,1275,5007,0701,0000,0000,1441
U 0055, 2742,6551,1717,4374,4007,0700,0010,0000,0007
U 0075, 0054,3447,0303,4174,4007,0700,0000,0000,0000
U 1206, 2724,3446,0303,4174,4007,0700,0010,0000,0000
U 1207, 1210,2113,0305,4174,4007,0521,4000,0000,0000
U 1210, 1212,2113,0305,4174,4007,0620,4000,0000,0000
U 1211, 2702,3221,0004,4174,4007,0700,0000,0000,0000
U 1212, 0033,3333,0017,4174,4003,5701,0000,0000,0000
U 1213, 1214,1003,0600,4174,4007,0521,4000,0000,0000
U 1214, 0033,3333,0017,4174,4003,5701,0000,0000,0000
U 1215, 2702,3221,0004,4174,4007,0700,0000,0000,0000

; 4377 .TOC "ARITHMETIC -- DDIV"
; 4378
; 4379 .DCODE
; 4380 117: DBL R, DAC, J/DDIV
; 4381 .UCODE
; 4382
; 4383 .IF/FULL
; 4384 1627:
; 4385 DDIV: Q [ARX].AND.[MAG] ;COPY LOW WORD
; 4386 [BR]_[AR]*.5, ;COPY MEMORY OPERAND
; 4387 SKIP AD.LE.0 ;SEE IF POSITIVE
; 4388 =0 [BR]_[BR]*.5 LONG, ;POSITIVE
; 4389 J/DDIV1 ;CONTINUE BELOW
; 4390 [BR]_[BR]*.5 LONG, ;NEGATIVE OR ZERO
; 4391 SKIP DPO ;SEE WHICH?
; 4392 =0 [MAG].AND.Q, ;SEE IF ALL ZERO
; 4393 SKIP AD.EQ.0, J/DDIV1 ;CONTINUE BELOW
; 4394 [T1]_O XWD [5] ;NEGATE MEM OP
; 4395 Q.Q.DR.#, #/600000, ;SIGN EXTEND THE LOW
; 4396 HOLD RIGHT ;WORD
; 4397 Q -Q ;MAKE Q POSITIVE
; 4398 [BR]_(-[BR]-.25)*.5 LONG, ;NEGATE HIGH WORD
; 4399 ASHC, MULTI PREC/1, ;USE CARRY FROM LOW WORD
; 4400 J/DDIV3 ;CONTINUE BELOW
; 4401 =0
; 4402 DDIV1: [BR]_[BR]*.5 LONG, ;SHIFT OVER 1 PLACE
; 4403 ASHC, J/DDIV2 ;CONTINUE BELOW
; 4404 NO DIVIDE ;DIVIDE BY ZERO
; 4405 DDIV2: [T1]_O XWD [4] ;MEM OPERAND IS POSITIVE
; 4406 DDIV3: [BRX]_Q, AC ;COPY Q
; 4407
; 4408 [AR]_AC*.5, 2T, SKIP DPO ;GET AC--SEE IF NEGATIVE
; 4409 =0*1*0
; 4410 DDIV3A: Q AC[1].AND.[MAG], ;POSITIVE (OR ZERO)
; 4411 J/DDIV4 ;CONTINUE BELOW
; 4412 [T1]_[T1].XOR.#, ;NEGATIVE
; 4413 #/7, CALL [QDNEG] ;UPDATE SAVED FLAGS
; 4414 =1*1*1 [AR]_[AR]*.5, ;SHIFT AR OVER
; 4415 J/DDIV3A ;GO BACK AND LOAD Q
; 4416 =
; 4417 =0
; 4418 DDIV4: [AR]_[AR]*.5 LONG, ;SHIFT AR OVER
; 4419 CALL [DDIV5] ;SHIFT 1 MORE PLACE
; 4420 [AR]-[BR], 3T, SKIP DPO ;TEST MAGNITUDE
; 4421 =0 [AR]-[BR], 2T,
; 4422 SKIP AD.EQ.0, J/DDIV5
; 4423 [ARX]_Q, J/DDIV5A ;ANSWER FITS
; 4424
; 4425 =0
; 4426 DDIV5: READ [T1], 3T, DISP/DP, J/NODDIV
; 4427 Q-[BRX], 3T, SKIP DPO
; 4428 =0 READ [T1], 3T, DISP/DP, J/NODDIV
; 4429 [ARX]_Q ;COPY LOW WORD
; 4430
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 121
ARITHMETIC -- DDIV

```
U 2702, 0314,4552,0000,1275,5007,0701,0000,0000,1442 ; 4431 ;HERE WITH EVERYTHING SETUP AND READY TO GO
U 0314, 1240,3446,1200,4174,4007,0700,2010,0071,0042 ; 4432 DDIV5A: Q_AC[2].AND.[MAG]
U 0316, 2703,3224,0016,4174,4007,0700,0000,0000,0000 ; 4433 =0* Q_Q*.5, SC_34., CALL [DBLDIV]
U 2703, 2704,0002,1600,4174,4007,0700,0000,0000,0000 ; 4434 [TO]_Q*2 LONG
U 2704, 1216,4003,0000,1174,4007,0700,0400,0000,1440 ; 4435 Q_Q+[TO]
U 1216, 2724,3442,0400,4174,4007,0700,0010,0000,0000 ; 4436 AC[0]_Q.AND.[MAG] ;STORE ANSWER
U 1217, 2705,3221,0004,4174,4007,0700,0000,0000,0000 ; 4437 =0 Q_[ARX], CALL [DDIVS] ;SHIFT OUT EXTRA ZERO BIT
U 2705, 0354,4552,0000,1275,5007,0701,0000,0000,1443 ; 4438 [ARX]_Q ;
; 4439 Q_AC[3].AND.[MAG]
; 4440 =0* [TO]_[AR]*.5 LONG, ;SHIFT Q, PUT AR ON DP
; 4441 SC_34., ;LOAD SHIFT COUNT
; 4442 SKIP DPO, ;LOOK AT AR SIGN
; 4443 CALL [DBLDIV] ;GO DIVIDE
U 0354, 1240,3446,0316,4174,4007,0520,2010,0071,0042 ; 4444 [TO]_Q*2 LONG
U 0356, 2706,3224,0016,4174,4007,0700,0000,0000,0000 ; 4445 READ [T1], 3T, DISP/DP ;WHAT SIGN IS QUO
U 2706, 0056,3333,0017,4174,4003,5701,0000,0000,0000 ; 4446 =1110 [TO]_[TO]+Q, ;POSITIVE QUO
; 4447 J/DDIV5B ;CONTINUE BELOW
; 4448 [TO]_-Q*2 ;NEGATIVE QUO
; 4449 AD/-D-.25, DBUS/RAM, 3T,
; 4450 RAMADR/AC#, DEST/Q_AD,
; 4451 MULTI PREC/1
U 2707, 2710,1772,0000,0274,4007,0701,0040,0000,0000 ; 4452 AC_Q, SKIP AD.EQ.O
U 2710, 1220,3223,0000,0174,4007,0621,0400,0000,0000 ; 4453 =0 AC[1]_[TO], J/DDIV5C
U 1220, 2712,3440,1616,1174,4007,0700,0400,0000,1441 ; 4454 AC[1]_O, J/DDIV6
U 1221, 2714,4223,0000,1174,4007,0700,0400,0000,1441 ; 4455
; 4456 DDIV5B: AC[1]_[TO].AND.[MAG], J/DDIV6 ;STORE LOW WORD IN + CASE
; 4457
; 4458 DDIV5C: [TO]_[TO].OR.#, #/400000, HOLD RIGHT
U 2711, 2714,4113,1600,1174,4007,0700,0400,0000,1441 ; 4459 AC[1]_[TO]
; 4460
; 4461 DDIV6: READ [AR], SKIP DPO ;LOOK AT AR SIGN
U 2712, 2713,3551,1616,4374,0007,0700,0000,0040,0000 ; 4462 =0
; 4463 DDIV7: Q_[ARX], J/DDIV8
U 2713, 2714,3440,1616,1174,4007,0700,0400,0000,1441 ; 4464 Q_[ARX]+[BRX]
; 4465 [AR]_[AR]+[BR],
; 4466 MULTI PREC/1
U 2714, 1222,3333,0003,4174,4007,0520,0000,0000,0000 ; 4467 Q_Q+[BRX]
; 4468 [AR]_[AR]+[BR],
; 4469 MULTI PREC/1
U 1222, 2720,3442,0400,4174,4007,0700,0000,0000,0000 ; 4470 DDIV8: READ [T1], 3T, DISP/DP
U 1223, 2715,0112,0406,4174,4007,0700,0000,0000,0000 ; 4471 =1101
; 4472 DDIV8A: [AR]_[AR]*2 LONG, ASHC, ;POSITIVE REMAINDER
; 4473 J/DDIV9 ;CONTINUE BELOW
; 4474 Q_-Q ;NEGATE REMAINDER IN ARIQ
U 2715, 2716,0111,0503,4174,4007,0700,0040,0000,0000 ; 4475 [AR]_(-[AR]-.25)*2 LONG,
U 2716, 2717,0002,0600,4174,4007,0700,0000,0000,0000 ; 4476 MULTI PREC/1, ASHC
; 4477
U 2717, 2720,0111,0503,4174,4007,0700,0040,0000,0000
U 2720, 0355,3333,0017,4174,4003,5701,0000,0000,0000
U 0355, 2722,3444,0303,4174,4047,0700,0000,0000,0000
U 0357, 2721,2222,0000,4174,4007,0700,4000,0000,0000
U 2721, 2722,2444,0303,4174,4047,0700,0040,0000,0000
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 . Page 122
ARITHMETIC -- DDIV

```
U 2722, 1224,0113,0303,1174,4007,0521,0400,0000,1442 ; 4478 DDIV9: AC[2]_[AR]+[AR], 3T,  
; 4479 SKIP DPO  
; 4480 =0 AC[3]_Q.AND.[MAG],  
U 1224, 0100,4003,0000,1174,4156,4700,0400,0000,1443 ; 4481 NEXT INST  
U 1225, 2723,4002,0000,1174,4007,0700,0000,0000,1443 ; 4482 Q_Q.AND.[MAG], AC[3]  
; 4483 AC[3]_[MAG].EQV.Q,  
U 2723, 0100,7003,0000,1174,4156,4700,0400,0000,1443 ; 4484 NEXT INST  
; 4485  
; 4486  
; 4487 ;HERE IF WE WANT TO SET NO DIVIDE  
; 4488 =11011  
U 0033, 2742,4443,0000,4174,4007,0700,0010,0000,0000 ; 4489 NODDIV: CALL [QDNEG] ;FIXUP AC TO AC+3  
U 0037, 0603,4443,0000,4174,4467,0700,0000,0051,1000 ; 4490 NO DIVIDE ;ABORT DIVIDE  
; 4491  
U 2724, 0001,3446,0303,4174,4044,1700,0000,0000,0000 ; 4492 DDIVS: [AR]_[AR]*.5 LONG, ASHC, RETURN [1]  
; 4493 .IFNOT/FULL  
; 4494 1627:  
; 4495 DDIV: UUO  
; 4496 .ENDIF/FULL  
; 4497
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 . Page 123
ARITHMETIC -- DIVIDE SUBROUTINE

```
; 4498 .TOC "ARITHMETIC -- DIVIDE SUBROUTINE"  
; 4499  
; 4500 ;HERE IS THE SUBROUTINE TO DO DIVIDE  
; 4501 ;ENTER WITH:  
; 4502 ; AR!Q = D'END  
; 4503 ; BR = D'SOR  
; 4504 ;RETURN 2 WITH:  
; 4505 ; AR = REMAINDER  
; 4506 ; Q = QUOTIENT  
; 4507 ;CALLER MUST CHECK FOR ZERO DIVIDE PRIOR TO CALL  
; 4508 ;  
; 4509 =1000  
; 4510 DIVSUB: Q_Q.AND.#, ;CLEAR SIGN BIT IN  
; 4511 #737777, ;MASK  
; 4512 HOLD RIGHT, ;JUST CLEAR BIT 0  
; 4513 CALL [DIVSGN] ;DO REAL DIVIDE  
; 4514 =1100 RETURN [2] ;ALL POSITIVE  
; 4515 =1101 Q_-Q, RETURN [2] ;-QUO +REM  
; 4516 =1110 Q_-Q ;ALL NEGATIVE  
; 4517 =1111 [AR]_-[AR], RETURN [2] ;NEGATIVE REMAINDER  
; 4518
```

U 0370, 2725,4662,0000,4374,0007,0700,0010,0037,7777
U 0374, 0002,4443,0000,4174,4004,1700,0000,0000,0000
U 0375, 0002,2222,0000,4174,4004,1700,4000,0000,0000
U 0376, 0377,2222,0000,4174,4007,0700,4000,0000,0000
U 0377, 0002,2441,0303,4174,4004,1700,4000,0000,0000

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 124
ARITHMETIC -- DIVIDE SUBROUTINE

U 2725, 1226,3333,0003,4174,4007,0520,0000,0000,0000
U 1226, 2727,4221,0004,4174,4007,0700,0000,0000,0000
U 1227, 1230,2222,0000,4174,4007,0621,4000,0000,0000
U 1230, 2726,7441,0303,4174,4007,0700,0000,0000,0000
U 1231, 2726,2441,0303,4174,4007,0700,4000,0000,0000

U 2726, 2727,3771,0004,4374,4007,0700,0000,0010,0000
U 2727, 1232,3333,0005,4174,4007,0520,0000,0000,0000

U 1232, 2731,3446,0303,4174,4007,0700,0000,0000,0000
U 1233, 2730,2441,0505,4174,4007,0700,4000,0000,0000

U 2730, 1232,3551,0404,4374,4007,0700,0000,0004,0000
U 2731, 2732,3447,0303,4174,4007,0700,0000,0000,0000
U 2732, 2733,3447,0505,4174,4007,0700,0000,0000,0000
U 2733, 2734,3447,0505,4174,4007,0700,0000,0000,0000
U 2734, 1234,1114,0503,4174,4067,0630,6000,0060,0000

U 1234, 1234,0114,0503,4174,4067,0630,2100,0060,0000
U 1235, 2735,3444,1717,4174,4067,0700,0100,0000,0000
U 2735, 1236,3447,0303,4174,4007,0520,0000,0000,0000

U 1236, 2736,3444,0303,4174,4007,0700,0000,0000,0000
U 1237, 1236,0111,0503,4174,4007,0700,0000,0000,0000
U 2736, 2737,3444,0303,4174,4007,0700,0000,0000,0000
U 2737, 2740,4002,1200,4174,4007,0700,0000,0000,0000

U 2740, 0014,3333,0004,4174,4000,1701,0000,0000,0000

```
; 4519 ;HERE IS THE INNER DIVIDE SUBROUTINE
; 4520 ;SAME SETUP AS DIVSUB
; 4521 ;RETURNS WITH AR AND Q POSITIVE AND
; 4522 ;      14 IF ALL POSITIVE
; 4523 ;      15 IF -QUO
; 4524 ;      16 IF ALL NEGATIVE
; 4525 ;      17 IF NEGATIVE REMAINDER
; 4526
; 4527 BASIC DIV STEP "DEST/Q_Q*2, DIV, A/BR, B/AR, STEP SC"
; 4528 DIV STEP "BASIC DIV STEP, AD/A+B, DIVIDE/1"
; 4529 FIRST DIV STEP "BASIC DIV STEP, AD/B-A-.25, ADD .25"
; 4530
; 4531 DIVSGN: READ [AR], SKIP DPO
; 4532 =0 [AR]_O, J/DVSB2 ;REMAINDER IS POSITIVE
; 4533 Q_-Q, SKIP AD.EQ.O ;COMPLEMENT LOW WORD
; 4534 =0 [AR]_.NOT.[AR], J/DVSB1 ;COMPLEMENT HI WORD
; 4535 [AR]_-[AR] ;TWO'S COMPLEMENT HI WORD SINCE
; ; LOW WORD WAS ZERO
; 4536
; 4537 DVSB1: [AR]_#, #/10000 ;REMAINDER IS NEGATIVE
; 4538 DVSB2: READ [BR], SKIP DPO ;IS THE DIVISOR NEGATIVE
; 4539 =0
; 4540 DVSB3: [AR]_[AR]*.5 LONG, ;START TO PUT IN 9-CHIPS
; 4541 J/DIVSET ;JOIN MAIN STREAM
; 4542 [BR]_-[BR] ;COMPLEMENT DIVISOR
; 4543 [AR]_[AR].OR.#, ;ADJUST SIGN OF QUOTIENT
; 4544 #/40000, J/DVSB3 ;USE 9 CHIPS
; 4545 DIVSET: [AR]_[AR]*.5
; 4546 [BR]_[BR]*.5
; 4547 [BR]_[BR]*.5
; 4548 FIRST DIV STEP
; 4549 ;HERE IS THE MAIN DIVIDE LOOP
; 4550 =0
; 4551 DIVIDE: DIV STEP, J/DIVIDE
; 4552 [T1]_[T1]*2 LONG, DIVIDE/1, DIV
; 4553 [AR]_[AR]*.5, SKIP DPO
; 4554 =0
; 4555 FIX++: [AR]_[AR]*2 LONG, J/FIX1++
; 4556 [AR]_[AR]+[BR], J/FIX++
; 4557 FIX1++: [AR]_[AR]*2 LONG
; 4558 Q_[MASK].AND.Q
; 4559 READ [ARX], 3T, ;RETURN TO 1 OF 4 PLACES
; 4560 DISP/1, ;BASED ON SIGN OF RESULT
; 4561 J/14 ;RETURN
; 4562
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 125
ARITHMETIC -- DOUBLE DIVIDE SUBROUTINE

```
; 4563 .TOC "ARITHMETIC -- DOUBLE DIVIDE SUBROUTINE"
; 4564 .IF/FULL
; 4565 ;CALL WITH:
; 4566 ; AR!ARX!Q = 3 WORD DV'END
; 4567 ; BR!BRX = 2 WORD DV'SOR
; 4568 ;RETURN 2 WITH:
; 4569 ; AR!ARX = 2 WORD REMAINDER
; 4570 ; CORRECT IF POSITIVE (Q IS ODD)
; 4571 ; WRONG (BY BR!BRX) IF NEGATIVE (Q IS EVEN)
; 4572 ; Q = 1 WORD QUOTIENT
; 4573 ;CALLER MUST CHECK FOR ZERO DIVIDE PRIOR TO CALL
; 4574 ;
; 4575 ;NOTE: THIS SUBROUTINE ONLY WORKS FOR POSITIVE NUMBERS
; 4576 ;
; 4577 =0
; 4578 ;HERE FOR NORMAL STARTUP
; 4579 DBLDIV: [ARX]_([ARX]-[BRX])*2 LONG, ;SUBTRACT LOW WORD
; 4580 LSHC, J/DIVHI ;GO ENTER LOOP
; 4581 ;SKIP ENTRY POINT IF FINAL STEP IN PREVIOUS ENTRY WAS IN ERROR
; 4582 [ARX]_([ARX]+[BRX])*2 LONG, ;CORRECTION STEP
; 4583 LSHC, J/DIVHI ;GO ENTER LOOP
; 4584 ;
; 4585 ;HERE IS DOUBLE DIVIDE LOOP
; 4586 DIVHI: AD/A+B, ;ADD (HARDWARE MAY OVERRIDE)
; 4587 A/BR, B/AR, ;OPERANDS ARE AR AND BR
; 4588 DEST/AD*2, ;SHIFT LEFT
; 4589 SHSTYLE/NORM, ;SET SHIFT PATHS (SEE DPE1)
; 4590 MULTI PREC/1, ;INJECT SAVED BITS
; 4591 STEP SC ;COUNT DOWN LOOP
; 4592 =0 AD/A+B, ;ADD (HARDWARE MAY OVERRIDE)
; 4593 A/BRX, B/ARX, ;LOW WORDS
; 4594 DEST/Q Q*2, ;SHIFT WHOLE MESS LEFT
; 4595 SHSTYLE/DIV, ;SET SHIFT PATHS (SEE DPE1)
; 4596 DIVIDE/1, ;SAVE BITS
; 4597 J/DIVHI ;KEEP LOOPING
; 4598 ;HERE WHEN ALL DONE
; 4599 DEST/Q Q*2, DIV, ;SHIFT IN LAST Q BIT
; 4600 DIVIDE/1, ;GENERATE BIT
; 4601 B/HR, RETURN [2] ;ZERO HR AND RETURN
; 4602
```

U 1240, 2741, 1114, 0604, 4174, 4057, 0700, 4000, 0000, 0000

U 1241, 2741, 0114, 0604, 4174, 4057, 0700, 0000, 0000, 0000

U 2741, 1242, 0115, 0503, 4174, 4007, 0630, 2040, 0060, 0000

U 1242, 2741, 0114, 0604, 4174, 4067, 0700, 0100, 0000, 0000

U 1243, 0002, 4444, 0002, 4174, 4064, 1700, 0100, 0000, 0000

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 126
ARITHMETIC -- SUBROUTINES FOR ARITHMETIC

```
U 2742, 2743, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1443
U 2743, 1244, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1443
U 1244, 2746, 7772, 0000, 1274, 4007, 0701, 0000, 0000, 1442
U 1245, 2744, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1442
U 2744, 1246, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1442
U 1246, 2747, 7772, 0000, 1274, 4007, 0701, 0000, 0000, 1441
U 1247, 2745, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1441
U 2745, 1250, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1441
U 1250, 2750, 7771, 0003, 0274, 4007, 0700, 0000, 0000, 0000
U 1251, 2750, 1771, 0003, 0274, 4007, 0701, 4000, 0000, 0000
U 2746, 1246, 4003, 0000, 1174, 4007, 0700, 0400, 0000, 1442
U 2747, 1250, 4003, 0000, 1174, 4007, 0700, 0400, 0000, 1441
U 2750, 0024, 3440, 0303, 0174, 4004, 1700, 0400, 0000, 0000
U 2751, 2752, 4551, 0404, 4374, 0007, 0700, 0000, 0037, 7777
U 2752, 1252, 2441, 0404, 4174, 4007, 0621, 4000, 0000, 0000
U 1252, 2202, 7441, 0303, 4174, 4467, 0700, 0000, 0001, 0001
U 1253, 2202, 2441, 0303, 4174, 4467, 0701, 4000, 0001, 0001
U 2753, 1254, 2441, 0404, 4174, 4007, 0621, 4000, 0000, 0000
U 1254, 2202, 7441, 0303, 4174, 4007, 0700, 0000, 0000, 0000
U 1255, 2202, 2441, 0303, 4174, 4007, 0700, 4000, 0000, 0000

; 4603 .TOC "ARITHMETIC -- SUBROUTINES FOR ARITHMETIC"
; 4604
; 4605 ;QUAD WORD NEGATE
; 4606 ;ARGUMENT IN AC!AC1!AC2!AC3
; 4607 ;LEAVES COPY OF AC!AC1 IN AR!Q
; 4608 ;RETURNS TO CALL!24
; 4609 QDNEG: Q_-AC[3]
; 4610 AC[3]_Q.AND.[MAG], ;PUT BACK LOW WORD
; 4611 SKIP AD.EQ.O ;SEE IF ANY CARRY
; 4612 =0
; 4613 COM2A: Q_.NOT.AC[2], J/COM2 ;CARRY--DO 1'S COMPLEMENT
; 4614 Q_-AC[2] ;NEXT WORD
; 4615 AC[2]_Q.AND.[MAG], ;PUT BACK WORD
; 4616 SKIP AD.EQ.O
; 4617 =0
; 4618 COM1A: Q_.NOT.AC[1], J/COM1
; 4619 Q_-AC[1]
; 4620 AC[1]_Q.AND.[MAG],
; 4621 SKIP AD.EQ.O
; 4622 =0
; 4623 COMOA: [AR]_.NOT.AC, J/COMO
; 4624 [AR]_-AC, 3T, J/COMO
; 4625
; 4626 COM2: AC[2]_Q.AND.[MAG], J/COM1A
; 4627 COM1: AC[1]_Q.AND.[MAG], J/COMOA
; 4628 COMO: AC_[AR], RETURN [24]
; 4629 .ENDIF/FULL
; 4630
; 4631 ;DOUBLE WORD NEGATE
; 4632 ;ARGUMENT IN AR AND ARX
; 4633 ;RETURNS TO CALL!2
; 4634
; 4635 DBLNEG: CLEAR ARXO ;FLUSH DUPLICATE SIGN
; 4636 DBLNGA: [ARX]_-[ARX], ;FLIP LOW WORD
; 4637 SKIP AD.EQ.O ;SEE IF CARRY
; 4638 =0 [AR]_.NOT.[AR], ;NO CARRY-- 1 COMP
; 4639 AD_FLAGS, J/CLARXO ;CLEAR LOW SIGN
; 4640 [AR]_-[AR], ;CARRY
; 4641 AD_FLAGS, 3T, J/CLARXO
; 4642
; 4643 ;SAME THING BUT DOES NOT SET PC FLAGS
; 4644 DBLNG1: [ARX]_-[ARX], SKIP AD.EQ.O
; 4645 =0 [AR]_.NOT.[AR], J/CLARXO
; 4646 [AR]_-[AR], J/CLARXO
; 4647
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 127
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984 ARITHMETIC -- SUBROUTINES FOR ARITHMETIC

```
; 4648 .NOBIN
; 4649 .TOC "BYTE GROUP -- IBP, ILDB, LDB, IDPB, DPB"
; 4650
; 4651
; 4652 ;ALL FIVE INSTRUCTIONS OF THIS GROUP ARE CALLED WITH THE BYTE POINTER
; 4653 ;IN THE AR. ALL INSTRUCTIONS SHARE COMMON SUBROUTINES.
; 4654
; 4655 ;IBP OR ADJBP
; 4656 ;IBP IF AC#0, ADJBP OTHERWISE
; 4657 ; HERE WITH THE BASE POINTER IN AR
; 4658
; 4659 ;HERE IS A MACRO TO DO IBP. WHAT HAPPENS IS:
; 4660 ; THE AR IS PUT ON THE DP.
; 4661 ; THE BR IS LOADED FROM THE DP WITH BITS 0-5 FROM SCAD
; 4662 ; THE SCAD COMPUTES P-S
; 4663 ; IBPS IS CALLED WITH A 4-WAY DISPATCH ON SCADO AND FIRST-PART-DONE
; 4664 ;THE MACRO IS WRITTEN WITH SEVERAL SUB-MACROS BECAUSE OF RESTRICTIONS
; 4665 ; IN THE MICRO ASSEMBLER
; 4666
; 4667 IBP DP "AD/D, DEST/A, A/AR, B/BR, DBUS/DBM, DBM/DP, BYTE/BYTE1"
; 4668 IBP SCAD "SCAD/A-B, SCADA/BYTE1, SCADB/SIZE"
; 4669 IBP SPEC "SCAD DISP, SKIP FPD"
; 4670 CALL IBP "IBP DP, IBP SCAD, IBP SPEC, CALL [IBPS], DT/3T"
; 4671
; 4672 SET P TO 36-S "AD/D,DEST/A,A/BR,B/AR,DBUS/DBM,DBM/DP,SCAD/A-B,SCADB/SIZE,BYTE/BYTE1,SCADA/PTR44"
; 4673
; 4674 ;THE FOLLOWING MACRO IS USED FOR COUNTING SHIFTS IN THE BYTE ROUTINES. IT
; 4675 ; USES THE FE AND COUNTS BY 8. NOTE: BYTE STEP IS A 2S WEIGHT SKIP NOT 1S.
; 4676 BYTE STEP "SCAD/A+B,SCADA/S#,S#/1770,SCADB/FE,LOAD FE, 3T,SCAD DISP"
; 4677
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 128
BYTE GROUP -- IBP, ILDB, LDB, IDPB, DPB

```
; 4678          .BIN
; 4679
; 4680          .DCODE
D 0133, 0015, 1610, 1100      ; 4681 133: R, AC, J/IBP ;OR ADJBP
D 0134, 0000, 1620, 1500      ; 4682 134: R,W TEST, J/ILDB ;CAN'T USE RPW BECAUSE OF FP
;
; 4683          R, J/LDB
; 4684          R,W TEST, J/IDPB
; 4685          R, J/DPB
; 4686          .UCODE
; 4687          1610:
; 4688          IBP: SKIP IF ACO ;SEE IF ADJBP
; 4689          =000 WORK[ADJPTR]_[AR], ;SAVE POINTER
; 4690          J/ADJBP ;GO ADJUST BYTE POINTER
U 0240, 3011, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0223 ; 4691 =001 CALL IBP ;BUMP BYTE POINTER
U 0241, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000 ; 4692 =101 DONE ;POINTER STORED
U 0245, 0110, 3443, 0100, 4174, 4156, 4700, 0200, 0014, 0012 ; 4693 =
; 4694
; 4695          1620:
U 1620, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000 ; 4696 ILDB: CALL IBP ;BUMP BYTE POINTER
; 4697          1624:
; 4698          LDB: READ [AR], ;LOOK AT POINTER
; 4699          LOAD BYTE EA, FE_P, 3T, ;GET STUFF OUT OF POINTER
; 4700          CALL [BYTEA] ;COMPUTE EFFECTIVE ADDRESS
U 1624, 2761, 3333, 0003, 4174, 4217, 0701, 1010, 0073, 0500 ; 4701 1625: VMA[PC], FETCH ;START FETCH OF NEXT INST
U 1625, 0411, 3443, 0100, 4174, 4007, 0700, 0200, 0014, 0012 ; 4702 =0* READ [AR], ;LOOK AT POINTER
; 4703          FE_FE.AND.S#, S#/0770, ;MASK OUT JUNK IN FE
; 4704          BYTE DISP, ;DISPATCH ON BYTE SIZE
; 4705          CALL [LDB1] ;GET BYTE
; 4706          AC_[AR], CLR FPD, ;STORE AC
; 4707          J/NIDISP ;GO DO NEXT INST
; 4708
; 4709          1630:
U 1630, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000 ; 4710 IDPB: CALL IBP ;BUMP BYTE POINTER
; 4711          1634:
; 4712          DPB: [ARX]_AC*2 ;PUT 7 BIT BYTE IN 28-34
; 4713          AD/A, A/ARX, SCAD/A, ;PUT THE BYTE INTO
; 4714          SCADA/BYTE5, 3T, ; INTO THE FE REGISTER
; 4715          LOAD FE ; FE REGISTER
; 4716          [ARX]_AC ;PUT BYTE IN ARX
; 4717          =100 READ [AR], ;LOOK AT BYTE POINTER
; 4718          LOAD BYTE EA, ;LOAD UP EFFECTIVE ADDRESS
; 4719          CALL [BYTEA] ;COMPUTE EFFECTIVE ADDRESS
; 4720          READ [AR], ;LOOK AT POINTER AGAIN
; 4721          BYTE DISP, ;DISPATCH ON SIZE
; 4722          CALL [DPB1] ;GO STORE BYTE
U 0264, 2761, 3333, 0003, 4174, 4217, 0700, 0010, 0000, 0500 ; 4723 =111 CLR FPD, J/DONE ;ALL DONE
U 0265, 0360, 3333, 0003, 4174, 4006, 5701, 0010, 0000, 0000 ; 4724 =
U 0267, 1400, 4443, 0000, 4174, 4467, 0700, 0000, 0005, 0000 ; 4725
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 129
BYTE GROUP -- INCREMENT BYTE POINTER SUBROUTINE

```

; 4726 .TOC "BYTE GROUP -- INCREMENT BYTE POINTER SUBROUTINE"
; 4727
; 4728 =00
U 0350, 2757,3441,0503,4174,4007,0700,0200,0003,0002 ; 4729 IBPS: [AR][BR], START WRITE, J/IBPX ;NO OVERFLOW, BR HAS ANSWER
U 0351, 0004,4443,0000,4174,4004,1700,0000,0000,0000 ; 4730 RETURN [4] ;FIRST PART DONE SET
U 0352, 2756,3770,0503,4334,4017,0700,0000,0032,6000 ; 4731 SET P TO 36-S, J/NXTWRD ;WORD OVERFLOW
U 0353, 0004,4443,0000,4174,4004,1700,0000,0000,0000 ; 4732 RETURN [4] ;FPD WAS SET IGNORE OVERFLOW
; 4733 ; AND RETURN
; 4734
U 2756, 2757,0111,0703,4170,4007,0700,0200,0003,0002 ; 4735 NXTWRD: [AR][AR]+1, HOLD LEFT, START WRITE ;BUMP Y AND RETURN
U 2757, 0004,3333,0003,4175,5004,1701,0200,0000,0002 ; 4736 IBPX: MEM WRITE, MEM_[AR], RETURN [4]
; 4737
; 4738
; 4739 .TOC "BYTE GROUP -- BYTE EFFECTIVE ADDRESS EVALUATOR"
; 4740
; 4741 ;ENTER WITH POINTER IN AR
; 4742 ;RETURN1 WITH (EA) IN VMA AND WORD IN BR
; 4743 BYTEAS: EA MODE DISP, ;HERE TO AVOID FPD
; 4744 J/BYTEAO ;GO COMPUTE EA
; 4745 BYTEA: SET FPD, ;SET FIRST-PART-DONE
; 4746 EA MODE DISP ;DISPATCH
; 4747 =100*
; 4748 BYTEAO: VMA [AR]+XR, ;INDEXING
; 4749 START READ, ;FETCH DATA WORD
; 4750 PXCT BYTE DATA, ;FOR PXCT
; 4751 J/BYTFET ;GO WAIT
; 4752 VMA [AR], ;PLAIN
; 4753 START READ, ;START CYCLE
; 4754 PXCT BYTE DATA, ;FOR PXCT
; 4755 J/BYTFET ;GO WAIT
; 4756 VMA [AR]+XR, ;BOTH
; 4757 START READ, ;START CYCLE
; 4758 PXCT BYTE PTR EA, ;FOR PXCT
; 4759 J/BYTFET ;GO DO INDIRECT
; 4760 VMA [AR], ;JUST @
; 4761 START READ, ;START READ
; 4762 PXCT BYTE PTR EA ;FOR PXCT
; 4763 BYTIND: MEM READ, ;WAIT FOR @ WORD
; 4764 [AR]_MEM, ;PUT IN AR
; 4765 HOLD LEFT, ;JUST IN RH (SAVE P & S)
; 4766 LQAD BYTE EA, ;LOOP BACK
; 4767 J/BYTEAS ; ..
; 4768
; 4769 BYTFET: MEM READ, ;WAIT FOR BYTE DATA
; 4770 [BR]_MEM.AND.MASK, ;WORD UNSIGNED
; 4771 RETURN [1] ;RETURN TO CALLER
; 4772
```

; T1OKI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 130
BYTE GROUP -- LOAD BYTE SUBROUTINE

```
; 4773 .TOC "BYTE GROUP -- LOAD BYTE SUBROUTINE"
; 4774
; 4775 ;CALL WITH:
; 4776 ; WORD IN BR
; 4777 ; POINTER IN AR
; 4778 ; P IN FE
; 4779 ; BYTE DISPATCH
; 4780 ;RETURN2 WITH BYTE IN AR
; 4781 LDB SCAD "SCAD/A,BYTE/BYTE5"
; 4782 7-BIT LDB "AD/D,DBUS/DBM,DBM/DP,DEST/A,A/BR,B/BR, LDB SCAD"
; 4783
; 4784 =000
; 4785 LDB1: GEN 17-FE, 3T, ;GO SEE IF ALL THE BITS
; 4786 SCAD DISP, ; ARE IN THE LEFT HALF
; 4787 J/LDBSWP ;GO TO LDBSWP & SKIP IF LH
; 4788
; 4789 ;HERE ARE THE 7-BIT BYTES
; 4790 =001 7-BIT LDB, SCADA/BYTE1, J/LDB7
; 4791 =010 7-BIT LDB, SCADA/BYTE2, J/LDB7
; 4792 =100 7-BIT LDB, SCADA/BYTE3, J/LDB7
; 4793 =101 7-BIT LDB, SCADA/BYTE4, J/LDB7
; 4794 =111 7-BIT LDB, SCADA/BYTE5, J/LDB7
; 4795 =
; 4796
; 4797 ;FOR 7-BIT BYTES WE HAVE BYTE IN BR 28-35 AND JUNK IN REST OF BR.
; 4798 ; WE JUST MASK THE SELECTED BYTE AND SHIFT ONE PLACE RIGHT.
; 4799 LDB7: AD/ZERO,RSRC/DA, ;LH ZERO, RH D.AND.A
; 4800 DBUS/DBM,DBM/##/376, ;D INPUT IS 376
; 4801 A/BR, ;A IS BR
; 4802 B/AR, ;PUT RESULT IN AR
; 4803 DEST/AD*.5, 3T, ;SHIFT RESULT 1 PLACE
; 4804 RETURN [2] ;RETURN TO CALLER
; 4805

U 0340, 0410,4443,0000,4174,4006,7701,0000,0031,0210

U 0341, 2764,3770,0505,4334,4057,0700,0000,0073,0000
U 0342, 2764,3770,0505,4334,4057,0700,0000,0074,0000
U 0344, 2764,3770,0505,4334,4057,0700,0000,0075,0000
U 0345, 2764,3770,0505,4334,4057,0700,0000,0076,0000
U 0347, 2764,3770,0505,4334,4057,0700,0000,0077,0000

U 2764, 0002,4257,0503,4374,4004,1701,0000,0000,0376
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 131
BYTE GROUP -- LOAD BYTE SUBROUTINE

```
U 0410, 2766,4443,0000,4174,4007,0700,1000,0031,0000
U 0412, 2765,3770,0505,4344,4007,0700,0000,0000,0000

U 2765, 2766,4221,0005,4174,0007,0700,1000,0031,0220

U 2766, 2767,3447,0505,4174,4007,0700,1020,0041,0010
U 2767, 2770,3333,0003,4174,4007,0700,1000,0031,7770
U 2770, 2771,4222,0000,4174,4007,0700,0000,0000,0000

U 2771, 2772,4224,0003,4174,4027,0700,1020,0041,0010
U 2772, 2773,4224,0003,4174,4027,0700,0000,0000,0000
U 2773, 0002,4001,0503,4174,4004,1700,0000,0000,0000

; 4806 ;HERE FOR NORMAL BYTES
; 4807 =00
; 4808 LDBSWP: FE_-FE, ;MAKE P NEGATIVE
; 4809 J/LDBSH ;JOIN MAIN LDB LOOP
; 4810 =10 [BR]_[BR] SWAP ;SHIFT 18 STEPS
; 4811 =
; 4812 [BR]_O, HOLD RIGHT, ;PUT ZERO IN LH
; 4813 FE_-FE+S#, S#/220 ;UPDATE FE
; 4814 LDBSH: [BR]_[BR]*.5, ;SHIFT RIGHT
; 4815 FE_FE+10, ;UPDATE THE FE
; 4816 MULTI SHIFT/1 ;FAST SHIFT
; 4817 READ [AR], FE_-S-10 ;GET SIZE
; 4818 Q_0 ;CLEAR Q
; 4819 GEN MSK [AR], ;PUT MASK IN Q (WIPEOUT AR)
; 4820 FE_FE+10, ;COUNT UP ALL STEPS
; 4821 MULTI SHIFT/1 ;FAST SHIFT
; 4822 GEN MSK [AR] ;ONE MORE BIT
; 4823 [AR]_[BR].AND.Q, RETURN [2]
; 4824
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 132
BYTE GROUP -- LOAD BYTE SUBROUTINE

```
; 4825 .NOBIN
; 4826 .TOC "BYTE GROUP -- DEPOSIT BYTE IN MEMORY"
; 4827
; 4828 ;FLOW FOR DPB (NOT 7-BIT BYTE)
; 4829 ;
; 4830 ;FIRST SET ARX TO -1 AND Q TO ZERO AND ROTATE LEFT
; 4831 ; S PLACES GIVING:
; 4832 ;
; 4833 ; ARX Q
; 4834 ; +-----+-----+
; 4835 ; !11111111111100000!00000000000111111!
; 4836 ; +-----+-----+
; 4837 ; !<--->!
; 4838 ; S BITS
; 4839 ;
; 4840
; 4841 ;NOW THE AC IS LOAD INTO THE ARX AND BOTH THE ARX AND Q
; 4842 ; ARE SHIFTED LEFT P BITS GIVING:
; 4843 ;
; 4844 ; +-----+-----+
; 4845 ; !?????BBBBBB00000!0000011111100000!
; 4846 ; +-----+-----+
; 4847 ; <----><----> <----><---->
; 4848 ; JUNK BYTE MASK P BITS
; 4849 ;
; 4850
; 4851 ;AT THIS POINT WE ARE ALMOST DONE. WE NEED TO AND
; 4852 ; THE BR WITH .NOT. Q TO ZERO THE BITS FOR THE BYTE
; 4853 ; AND AND ARX WITH Q TO MASK OUT THE JUNK THIS GIVES:
; 4854 ;
; 4855 ; ARX
; 4856 ; +-----+
; 4857 ; !00000BBBBBB00000!
; 4858 ; +-----+
; 4859 ;
; 4860 ; AR
; 4861 ; +-----+
; 4862 ; !DDDDDD00000DDDDDD!
; 4863 ; +-----+
; 4864 ;
; 4865 ;WE NOW OR THE AR WITH ARX TO GENERATE THE ANSWER.
; 4866
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 133
BYTE GROUP -- DEPOSIT BYTE IN MEMORY

```
; 4867 .BIN
; 4868
; 4869 ;DEPOSIT BYTE SUBROUTINE
; 4870 ;CALL WITH:
; 4871 ; BYTE POINTER IN AR
; 4872 ; BYTE TO STORE IN ARX
; 4873 ; WORD TO MERGE WITH IN BR
; 4874 ; (E) OF BYTE POINTER IN VMA
; 4875 ; 7-BIT BYTE IN FE
; 4876 ; BYTE DISPATCH
; 4877 ;RETURN2 WITH BYTE IN MEMORY
; 4878 ;
; 4879 DPB SCAD "SCAD/A+B,SCADA/S#,SCADB/FE,S#/O"
; 4880 7-BIT DPB "AD/D,DEST/A,A/BR,DBUS/DBM,DBM/DP,B/AR, DPB SCAD"
; 4881
; 4882 =000
; 4883 DPB1: READ [AR], FE -S-10, J/DPB&LO ;NOT SPECIAL
; 4884 =001 7-BIT DPB, BYTE/BYTE1, J/DPB7
; 4885 =010 7-BIT DPB, BYTE/BYTE2, J/DPB7
; 4886 =100 7-BIT DPB, BYTE/BYTE3, J/DPB7
; 4887 =101 7-BIT DPB, BYTE/BYTE4, J/DPB7
; 4888 =111 7-BIT DPB, BYTE/BYTE5, J/DPB7
; 4889 =
; 4890 DPB7: [MAG]_[MASK]*.5, START WRITE
; 4891 MEM WRITE, MEM_[AR], RETURN [2]
; 4892
; 4893
; 4894 DPBSLO: Q Q ;CLEAR Q
; 4895 GEN MSK [MAG], ;GENERATE MASK IN Q (ZAP MAG)
; 4896 FE FE+10, ;COUNT STEPS
; 4897 MULTI SHIFT/1 ;FAST SHIFT
; 4898 GEN MSK [MAG] ;ONE MORE BITS
; 4899 READ [AR], 3T, FE_P ;AMOUNT TO SHIFT
; 4900 FE FE.AND.S#, S#/O770 ;MASK OUT JUNK
; 4901 Q Q.AND.[MASK], ;CLEAR BITS 36 AND 37
; 4902 FE -FE ;MINUS NUMBER OF STEPS
; 4903 [ARX]_[ARX]*2 LONG, ;SHIFT BYTE AND MASK
; 4904 FE FE+10, ;COUNT OUT STEPS
; 4905 MULTI SHIFT/1 ;FAST SHIFT
; 4906 ;AT THIS POINT WE HAVE DONE ALL THE SHIFTING WE NEED. THE BYTE IS
; 4907 ; IN ARX AND THE MASK IS IN Q.
; 4908 [AR]_.NOT.Q
; 4909 [AR]_[AR].AND.[BR]
; 4910 [ARX]_[ARX].AND.Q
; 4911 [AR]_[AR].OR.[ARX],
; 4912 J/DPB7
; 4913
```

U 0360, 2776, 3333, 0003, 4174, 4007, 0700, 1000, 0031, 7770
U 0361, 2774, 3770, 0503, 4334, 4017, 0700, 0000, 0041, 0000
U 0362, 2774, 3770, 0503, 4334, 4027, 0700, 0000, 0041, 0000
U 0364, 2774, 3770, 0503, 4334, 4037, 0700, 0000, 0041, 0000
U 0365, 2774, 3770, 0503, 4334, 4047, 0700, 0000, 0041, 0000
U 0367, 2774, 3770, 0503, 4334, 4057, 0700, 0000, 0041, 0000

U 2774, 2775, 3447, 1200, 4174, 4007, 0700, 0200, 0003, 0002
U 2775, 0002, 3333, 0003, 4175, 5004, 1701, 0200, 0000, 0002

U 2776, 2777, 4222, 0000, 4174, 4007, 0700, 0000, 0000, 0000

U 2777, 3000, 4224, 0000, 4174, 4027, 0700, 1020, 0041, 0010
U 3000, 3001, 4224, 0000, 4174, 4027, 0700, 0000, 0000, 0000
U 3001, 3002, 3333, 0003, 4174, 4007, 0701, 1000, 0073, 0000
U 3002, 3003, 4443, 0000, 4174, 4007, 0700, 1000, 0051, 0770

U 3003, 3004, 4002, 1200, 4174, 4007, 0700, 1000, 0031, 0000

U 3004, 3005, 3444, 0404, 4174, 4007, 0700, 1020, 0041, 0010

U 3005, 3006, 7221, 0003, 4174, 4007, 0700, 0000, 0000, 0000
U 3006, 3007, 4111, 0503, 4174, 4007, 0700, 0000, 0000, 0000
U 3007, 3010, 4001, 0404, 4174, 4007, 0700, 0000, 0000, 0000

U 3010, 2774, 3111, 0403, 4174, 4007, 0700, 0000, 0000, 0000

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 134
BYTE GROUP -- ADJUST BYTE POINTER

U 3011, 1256,3770,0304,4344,4007,0700,2000,0071,0011
U 1256, 1256,3447,0404,4174,4007,0630,2000,0060,0000
U 1257, 3012,4557,0404,4374,4007,0701,0000,0000,0176
U 3012, 3013,3771,0004,4374,0007,0700,0000,0000,0000
U 3013, 3014,3333,0004,7174,4007,0700,0400,0000,0221
U 3014, 3015,4557,0305,4374,4007,0701,0000,0000,7700
U 3015, 3016,3770,0505,4344,4007,0700,2000,0071,0003
U 3016, 1260,4221,0005,4174,0007,0700,0000,0000,0000

```
; 4914 .TOC "BYTE GROUP -- ADJUST BYTE POINTER"
; 4915 .IF/FULL
; 4916 ;FIRST THE NUMBER OF BYTES PER WORD IS COMPUTED FROM THE
; 4917 ; FOLLOWING FORMULA:
; 4918 ;
; 4919 ; ( P ) ( 36-P )
; 4920 ; BYTES PER WORD = INT( --- ) + INT( ---- )
; 4921 ; ( S ) ( S )
; 4922 ;
; 4923 ;THIS GIVES 2 BYTES PER WORD FOR THE FOLLOWING 12 BIT BYTE:
; 4924 ; !=====!
; 4925 ; ! 6 !//////////! 12 ! 6 !
; 4926 ; !=====!
; 4927 ; P=18 AND S=12
; 4928 ;
; 4929 ;WE GET 3 BYTES/WORD IF THE BYTES FALL IN THE NATURAL PLACE:
; 4930 ; !=====!
; 4931 ; ! 12 !\\\\\\\\\\\\\\\\! 12 !
; 4932 ; !=====!
; 4933 ; P=12 AND S=12
; 4934 ;
; 4935 ;WE COME HERE WITH THE BYTE POINTER IN AR, AND ADJPTR
; 4936 ADJBP: [ARX]_[AR] SWAP, ;MOVE SIZE OVER
; 4937 SC_9. ;READY TO SHIFT
; 4938 =0
; 4939 ADJBPO: [ARX]_[ARX]*.5, ;SHIFT P OVER
; 4940 STEP SC, ;..
; 4941 J/ADJBPO ;..
; 4942 [ARX]_([ARX].AND.#)*.5, ;SHIFT AND MASK
; 4943 3T, ;WAIT
; 4944 #/176 ;6 BIT MASK
; 4945 [ARX]_#, ;CLEAR LH
; 4946 #/0, ;..
; 4947 HOLD RIGHT ;..
; 4948 WORK[ADJP]_[ARX] ;SAVE P
; 4949 [BR]_([AR].AND.#)*.5, ;START ON S
; 4950 3T, ;EXTRACT S
; 4951 #/007700 ;..
; 4952 [BR]_[BR] SWAP, ;SHIFT 18 PLACES
; 4953 SC 3 ;..
; 4954 [BR]_0, ;CLEAR LH
; 4955 HOLD_RIGHT ;..
; 4956
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 135
BYTE GROUP -- ADJUST BYTE POINTER

```
U 1260, 1260,3447,0505,4174,4007,0630,2000,0060,0000
U 1261, 1262,3333,0005,7174,4007,0621,0400,0000,0222
U 1262, 0660,3442,0400,4174,4007,0700,2000,0071,0042
U 1263, 1404,3771,0003,7274,4007,0701,0000,0000,0223
U 0660, 0370,3771,0003,4374,4007,0700,0010,0000,0000
U 0662, 3017,3223,0000,7174,4007,0700,0400,0000,0224
U 3017, 3020,3772,0000,4370,4007,0700,0000,0000,0044
U 3020, 3021,1662,0000,7274,4007,0701,4000,0000,0221
U 3021, 3022,3771,0005,7274,4007,0701,0000,0000,0222
U 3022, 0664,4443,0000,4174,4007,0700,2000,0071,0042
U 0664, 0370,3771,0003,4374,4007,0700,0010,0000,0000
U 0666, 3023,3333,0003,7174,4007,0700,0400,0000,0225
U 3023, 3024,3771,0003,4374,4007,0700,0000,0077,7777
U 3024, 1264,0661,0005,7274,4007,0622,0000,0000,0224
U 1264, 0550,0662,0000,0274,4007,0522,2000,0071,0042
U 1265, 0603,4443,0000,4174,4467,0700,0000,0051,1000

; 4957 =0
; 4958 ADJBP1: [BR]_[BR]*.5, ;SHIFT S OVER
; 4959 STEP SC, ; ..
; 4960 J/ADJBP1 ; ..
; 4961 WORK[ADJS]_[BR], ;SALT S AWAY
; 4962 SKIP AD.EQ.O ;SEE IF ZERO
; 4963 =0 Q_[ARX], ;DIVIDE P BY S
; 4964 SC 34., ;STEP COUNT
; 4965 J/ADJBP2 ;SKIP NEXT WORD
; 4966 [AR]_WORK[ADJPTR], J/MOVE ;S=0 -- SAME AS MOVE
; 4967 =0*
; 4968 ADJBP2: [AR]_#, ;FILL AR WITH SIGN BITS
; 4969 #/O, ;POSITIVE
; 4970 CALL [DIVSUB] ;GO DIVIDE
; 4971 WORK[ADJQ1]_Q ;SAVE QUOTIENT
; 4972 Q #, ;COMPUTE (36-P)/S
; 4973 #/36., ; ..
; 4974 HOLD LEFT ;SMALL ANSWER
; 4975 Q Q-WORK[ADJP] ;SUBTRACT P
; 4976 [BR]_WORK[ADJS] ;DIVIDE BY S
; 4977 SC 34. ;STEP COUNT
; 4978 =0* [AR]_#, ;MORE SIGN BITS
; 4979 #/O, ; ..
; 4980 CALL [DIVSUB] ;GO DIVIDE
; 4981 WORK[ADJR2]_[AR] ;SAVE REMAINDER
; 4982 [AR] #, ;ASSUME NEGATIVE ADJ
; 4983 #/777777 ;EXTEND SIGN
; 4984 AD/D+Q, ;BR_(P/S)+((36-P)/S)
; 4985 DEST/AD, ; ..
; 4986 B/BR, ; ..
; 4987 RAMADR/#, ; ..
; 4988 DBUS/RAM, ; ..
; 4989 WORK/ADJQ1, ; ..
; 4990 4T, ; ..
; 4991 SKIP AD.EQ.O ;SEE IF ZERO
; 4992 =0 Q Q+AC, ;GET ADJUSTMENT
; 4993 SC 34., ;STEP COUNT
; 4994 SKIP DPO, ;GO DO DIVIDE
; 4995 4T, ;WAIT FOR DP
; 4996 J/ADJBP3 ;BELOW
; 4997 NO DIVIDE ;O BYTES/WORD
; 4998
```


; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 136
BYTE GROUP -- ADJUST BYTE POINTER

```
; 4999 ;WE NOW DIVIDE THE ADJUSTMENT BY THE BYTES PER WORD AND FORCE THE
; 5000 ; REMAINDER (R) TO BE A POSITIVE NUMBER (MUST NOT BE ZERO). THE
; 5001 ; QUOTIENT IS ADDED TO THE Y FIELD IN THE BYTE POINTER AND THE NEW
; 5002 ; P FIELD IS COMPUTED BY:
; 5003 ;
; 5004 ; ( ( 36-P ))
; 5005 ; NEW P = 36-((R * S) + RMDR( ---- ))
; 5006 ; ( S ))
; 5007 ;
; 5008 ;WE NOW HAVE BYTES/WORD IN BR AND ADJUSTMENT IN Q. DIVIDE TO GET
; 5009 ; WORDS TO ADJUST BY.
; 5010 =00
; 5011 ADJBP3: [AR]_#, ;POSITIVE ADJUSTMENT
; 5012 #/O.
; 5013 WORK[ADJBPW][BR], ;SAVE BYTES/WORD & COMPUTE
; 5014 CALL [DIVSUB] ; ADJ/(BYTES/WORD)
; 5015 ;WE NOW WANT TO ADJUST THE REMAINDER SO THAT IT IS POSITIVE
; 5016 =11 Q_#, ;ONLY RIGHT HALF
; 5017 #/O, ; ..
; 5018 HOLD RIGHT ; ..
; 5019 =
; 5020 READ [AR], ;ALREADY +
; 5021 SKIP AD.LE.O ; ..
; 5022 =0
; 5023 ADJBP4: AD/D+Q, ;ADD Q TO POINTER AND STORE
; 5024 DEST/AD, ; ..
; 5025 B/BR, ;RESULT TO BR
; 5026 RAMADR/#, ;PTR IS IN RAM
; 5027 DBUS/RAM, ; ..
; 5028 WORK/ADJPTR, ; ..
; 5029 INH CRY18, ;JUST RH
; 5030 3T, ;WAIT FOR RAM
; 5031 J/ADJBP5 ;CONTINUE BELOW
; 5032 Q_Q-1, ;NO--MAKE Q SMALLER
; 5033 HOLD LEFT ; ..
; 5034 [AR]_[AR]+WORK[ADJBPW], ;MAKE REM BIGGER
; 5035 J/ADJBP4 ;NOW HAVE + REMAINDER
; 5036
```

U 0550, 0551,3771,0003,4374,4007,0700,0000,0000,0000

U 0551, 0370,3333,0005,7174,4007,0700,0410,0000,0226

U 0553, 3025,3772,0000,4374,0007,0700,0000,0000,0000

U 3025, 1266,3333,0003,4174,4007,0421,0000,0000,0000

U 1266, 3027,0661,0005,7274,4407,0701,0000,0000,0223

U 1267, 3026,1002,0700,4170,4007,0700,4000,0000,0000

U 3026, 1266,0551,0303,7274,4007,0701,0000,0000,0226

```
; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984
```

```
MICRO 31(254)
```

```
KS10 MICROCODE V124, 27-JUL-84 Page 137
BYTE GROUP -- ADJUST BYTE POINTER
```

```
U 3027, 3030,3441,0306,4174,4007,0700,2000,0071,0043
U 3030, 0062,3772,0000,7274,4007,0701,0000,0000,0222
U 0062, 2662,3446,0606,4174,4007,0700,0010,0000,0000

U 0066, 3031,0661,0003,7274,4007,0701,0000,0000,0225

U 3031, 3032,2555,0303,4374,4007,0701,4000,0000,0044
U 3032, 3033,3770,0303,4344,4007,0700,2000,0071,0011

U 3033, 1270,3771,0003,4370,4007,0700,0000,0000,0000

U 1270, 1270,3445,0303,4174,4007,0630,2000,0060,0000

U 1271, 3034,4551,0505,4374,0007,0700,0000,0000,7777

U 3034, 1400,3113,0305,0174,4007,0700,0400,0000,0000
```

```
; 5037 ADJBP5: [BRX]_[AR], ;COMPUTE R*S
; 5038 SC_35 ;STEP COUNT
; 5039 Q_WORK[ADJS] ;GET S
; 5040 =01* [BRX]_[BRX]*.5 LONG, ;SHIFT OVER
; 5041 CALL [MULSUB] ;..
; 5042 AD/D+Q, ;AR_(R*S)+RMDR(36-P)/S
; 5043 DEST/AD, ;..
; 5044 B/AR, ;..
; 5045 RAMADR/#, ;..
; 5046 3T, ;..
; 5047 DBUS/RAM, ;..
; 5048 WORK/ADJR2 ;..
; 5049 [AR]_(#[AR])*2, ;COMPUTE 36-AR
; 5050 3T, ;AND START LEFT
; 5051 #/36. ;..
; 5052 [AR]_[AR] SWAP, ;PUT THE POSITION BACK
; 5053 SC_9. ;..
; 5054 [AR]_#, ;CLEAR JUNK FROM RH
; 5055 #/0, ;..
; 5056 HOLD LEFT ;..
; 5057 =0
; 5058 ADJBP6: [AR]_[AR]*2, ;LOOP OVER ALL BITS
; 5059 STEP SC, ;..
; 5060 J/ADJBP6 ;..
; 5061 [BR]_[BR].AND.#, ;..
; 5062 #/007777, ;..
; 5063 HOLD RIGHT ;..
; 5064 AC_[AR].OR.[BR], ;ALL DONE
; 5065 J/DONE
;5066 .IFNOT/FULL
;5067
;5068 ADJBP: UUO ;NO ADJBP IN SMALL
;5069 ; MICROCODE
; 5070 .ENDIF/FULL
; 5071
```

Produced on Advanced Information Services Electronic Laser Printer, PNO11E66, DTN: 223-7881

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 138
BYTE GROUP -- ADJUST BYTE POINTER

; 5072 .NOBIN
; 5073 .TOC "BLT"
; 5074
; 5075 ;THIS CODE PROVIDES A GUARANTEED RESULT IN AC ON COMPLETION OF
; 5076 ; THE TRANSFER (EXCEPT IN THE CASE AC IS PART OF BUT NOT THE LAST WORD
; 5077 ; OF THE DESTINATION BLOCK). WHEN AC IS NOT PART OF THE DESTINATION
; 5078 ; BLOCK, IT IS LEFT CONTAINING THE ADDRESSES OF THE FIRST WORD FOLLOWING
; 5079 ; THE SOURCE BLOCK (IN THE LH), AND THE FIRST WORD FOLLOWING THE DEST-
; 5080 ; INATION BLOCK (IN THE RH). IF AC IS THE LAST WORD OF THE DESTINATION
; 5081 ; BLOCK, IT WILL BE A COPY OF THE LAST WORD OF THE SOURCE BLOCK.
; 5082
; 5083 ;IN ADDITION, A SPECIAL-CASE CHECK IS MADE FOR THE CASE IN WHICH EACH
; 5084 ; WORD STORED IS USED AS THE SOURCE OF THE NEXT TRANSFER. IN THIS CASE,
; 5085 ; ONLY ONE READ NEED BE PERFORMED, AND THAT DATA MAY BE STORED FOR EACH
; 5086 ; TRANSFER. THUS THE COMMON USE OF BLT TO CLEAR CORE IS SPEEDED UP.
; 5087

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 139
BLT

```

; 5088 .BIN
; 5089
; 5090 ;HERE TO SETUP FOR A BLT/UBABLT
; 5091
U 3035, 1272,3770,0604,4344,4007,0700,0000,0000,0000 ; 5092 SETBLT: [ARX]_[BRX] SWAP ;COPY TO ARX (SRC IN RH)
; 5093 =0 VMA [ARX], ;ADDRESS OF FIRST WORD
; 5094 START READ,
; 5095 PXCT BLT SRC,
; 5096 CALL [CLARXL] ;CLEAR THE LEFT HALF OF
; 5097 [BRX]_O, ; BOTH SRC AND DEST
U 1272, 3565,3443,0400,4174,4007,0700,0210,0004,0712 ; 5098 HOLD RIGHT
U 1273, 3036,4221,0006,4174,0007,0700,0000,0000,0000 ; 5099 Q_[AR]-[BRX] ;NUMBER OF WORDS TO MOVE
U 3036, 3037,2112,0306,4174,4007,0700,4000,0000,0000 ; 5100 [BR]_Q+1 ;LENGTH +1
U 3037, 3040,0001,0705,4174,4007,0700,0000,0000,0000 ; 5101 [BR]_[BR] SWAP, ;COPY TO BOTH HALFS
; 5102 HOLD RIGHT
U 3040, 3041,3770,0505,4344,0007,0700,0000,0000,0000 ; 5103 [BR]_AC+[BR], ;FINAL AC
U 3041, 3042,0551,0505,0274,4407,0701,0000,0000,0000 ; 5104 INH CRY18 ;KEEP AC CORRECT IF DEST IS 777777
U 3042, 0002,3771,0013,4370,4004,1700,0000,0000,0001 ; 5105 STATE_[BLT],RETURN [2] ;SET PAGE FAIL FLAGS
; 5106
; 5107 .DCODE
D 0251, 0000,1640,2100 ; 5108 251: I, J/BLT
; 5109 .UCODE
; 5110
; 5111 1640:
U 1640, 3035,3771,0006,0276,6007,0700,0010,0000,0000 ; 5112 BLT: [BRX]_AC,CALL [SETBLT] ;FETCH THE AC (DEST IN RH)
; 5113 1642: AC [BR], ;STORE BACK IN AC
U 1642, 3564,3440,0505,0174,4007,0700,0410,0000,0000 ; 5114 CALL [LOADQ] ;LOAD FIRST WORD INTO Q
; 5115 1643: [BR]_[ARX]+1000001, ;SRC+1
; 5116 3T,
U 1643, 3043,0551,0405,4370,4007,0701,0000,0000,0001 ; 5117 HOLD LEFT
; 5118 [BR]-[BRX], 3T, ;IS THIS THE CORE CLEAR CASE
U 3043, 1274,2113,0506,4174,4007,0331,4000,0000,0000 ; 5119 SKIP ADR.EQ.O
; 5120 =0
; 5121 BLTLP1: VMA [BRX], ;NO, GET DEST ADR
; 5122 START WRITE, ;START TO STORE NEXT WORD
; 5123 PXCT BLT DEST, ;WHERE TO STORE
; 5124 J/BLTGO
U 1274, 3046,3443,0600,4174,4007,0700,0200,0003,0312 ; 5125
; 5126
; 5127 ;SKIP TO NEXT PAGE IF CLEARING CORE
```

Produced on Advanced Information Services Electronic Laser Printer, RK011556, DTN: 223-7881

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 140
BLT

```
U 1275, 3044,3443,0600,4174,4007,0700,0200,0003,0312
U 3044, 1276,3223,0000,4174,4007,0671,0200,0000,0002
U 1276, 3047,4443,0000,4174,4007,0700,0000,0000,0000
U 1277, 1300,2113,0603,4174,4007,0521,4000,0000,0000
U 1300, 1400,4221,0013,4170,4007,0700,0000,0000,0000
U 1301, 1302,0111,0704,4174,4007,0370,0000,0000,0000
U 1302, 3044,0111,0706,4170,4007,0700,0200,0003,0312
U 1303, 3046,0111,0706,4170,4007,0700,0200,0003,0312
U 3045, 1274,3772,0000,4365,5007,0700,0200,0000,0002
U 3046, 3047,3223,0000,4174,4007,0701,0200,0000,0002
U 3047, 1304,2113,0603,4174,4007,0521,4000,0000,0000
U 1304, 1400,4221,0013,4170,4007,0700,0000,0000,0000
U 1305, 3050,0111,0706,4174,4007,0700,0000,0000,0000
U 3050, 3045,0111,0704,4170,4007,0700,0200,0004,0712
U 3051, 3052,3770,0303,4344,4007,0700,0000,0000,0000
U 3052, 3053,3771,0003,7270,4007,0701,0000,0000,0214
U 3053, 1100,3440,0303,0174,4007,0700,0400,0000,0000

; 5128 ;CLEAR CORE CASE
; 5129 VMA [BRX],
; 5130 START WRITE,
; 5131 PXCT BLT DEST
; 5132 BLTCLR: MEM WRITE, ;STORE WORD
; 5133 MEM Q,
; 5134 SKIP/-1 MS ;1 MS TIMER UP
; 5135 =0 J/BLTGOT ;GO TAKE INTERRUPT
; 5136 [BRX]-[AR], ;BELOW E?
; 5137 3T,
; 5138 SKIP DPO
; 5139 =0 END BLT, ;NO--STOP BLT
; 5140 J/DONE
; 5141 [ARX][ARX]+1, ;FOR PAGE FAIL LOGIC
; 5142 SKIP IRPT
; 5143 =0 VMA [BRX]+1,
; 5144 LOAD VMA,
; 5145 PXCT BLT DEST,
; 5146 START WRITE, ;YES--KEEP STORING
; 5147 J/BLTCLR
; 5148 VMA [BRX]+1, ;INTERRUPT
; 5149 LOAD VMA,
; 5150 PXCT BLT DEST,
; 5151 START WRITE,
; 5152 J/BLTGOT
; 5153
; 5154 ;HERE FOR NORMAL BLT
; 5155 BLTLP: MEM READ, ;FETCH
; 5156 Q MEM,
; 5157 J/BLTLP1
; 5158 BLTGO: MEM WRITE, ;STORE
; 5159 MEM Q
; 5160 BLTGOT: [BRX]-[AR], ;BELOW E?
; 5161 3T,
; 5162 SKIP DPO
; 5163 =0 END BLT, ;NO--STOP BLT
; 5164 J/DONE
; 5165 [BRX][BRX]+1 ;UPDATE DEST ADDRESS
; 5166 VMA [ARX]+1,
; 5167 LOAD VMA,
; 5168 PXCT BLT SRC,
; 5169 START READ, ;YES--MOVE 1 MORE WORD
; 5170 J/BLTLP
; 5171
; 5172 ;HERE TO CLEAN UP AFTER BLT PAGE FAILS
; 5173 BLT-CLEANUP:
; 5174 [AR][AR] SWAP ;PUT SRC IN LEFT HALF
; 5175 [AR]_WORK[SV.BRX],
; 5176 HOLD LEFT
; 5177 AC [AR], ;STORE THE AC AND RETURN
; 5178 J/CLEANED
; 5179
```

; T10KI.MCR[10,1141]
; SIMPLE.MIC[10,1141]

11:45 11-AUG-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 141
BLT

Produced on Advanced Information Services Electronic Laser Printer, PK011E6, DTN: 223-7881

```
; 5180 .IF/UBABLT
; 5181 .TOC "UBABLT - BLT BYTES TO/FROM UNIBUS FORMAT"
; 5182
; 5183 ;THESE INSTRUCTION MOVE WORDS FROM BYTE TO UNIBUS AND UNIBUS TO BYTE
; 5184 ;FORMAT. FORMATS ARE:
; 5185 ;
; 5186 ;BYTE FORMAT:
; 5187 ;
; 5188 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5189 ; :: BYTE 0 ;; BYTE 1 ;; BYTE 2 ;; BYTE 3 ;; 4 BITS ;;
; 5190 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5191 ;
; 5192 ;UNIBUS FORMAT:
; 5193 ;
; 5194 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5195 ; :: 2 BITS ;; BYTE 1 ;; BYTE 0 ;; 2 BITS ;; BYTE 3 ;; BYTE 2 ;;
; 5196 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5197 ;
; 5198
; 5199 =0*
; 5200 BLTX: [BRX] AC, ;FETCH THE AC (DEST IN RH)
; 5201 CALL [SETBLT] ;DO THE REST OF THE SETUP
; 5202 AC[BR] ;STORE THE FINAL AC IN AC
; 5203
; 5204 BLTXLP: MEM READ, ;READ THE SOURCE WORD
; 5205 Q_MEM, ;FROM MEMORY
; 5206 B_DISP ;SKIP IF BLTUB (OPCODE 717)
; 5207 =110 Q_Q*.5, ;BLTBU (OPCODE 716) - SHIFT RIGHT 1

; 5208 J/BLTBU1 ;CONTINUE INSTRUCTION
; 5209
; 5210 AD/D.AND.Q,DBUS/DBM, ;BLTUB - MASK LOW BYTES, SHIFT LEFT
; 5211 DBM/#,#/377,DEST/AD*2,B/T1 ;AND STORE RESULT
; 5212 =00 FE_S#,S#/1767, ;-9 MORE BITS TO PUT LOW BYTE OF LH
; 5213 CALL [T1LSH] ; IN TOP OF LH SHIFT LEFT
; 5214 =01 FE_S#,S#/1772, ;-6 BITS TO PUT HI BYTE TO RIGHT
; 5215 CALL [Q_RSH] ; OF LOW BYTE.
; 5216 =11 Q_Q.AND.#,#/001774 ;KEEP ONLY HI BYTES
; 5217 =
; 5218 AD/A.OR.Q,A/T1,DEST/AD, ;MERGE PAIRS OF BYTES. NOW SWAPPED,
; 5219 B/T1 ;BUT STILL IN HALF-WORDS
; 5220 AD/57,RSRC/OA,A/T1, ;CLEAR LH OF Q WHILE LOADING
; 5221 DEST/Q_AD ;RH WITH LOW WORD
; 5222 Q_Q*2 ;SHIFT LOW WORD ACROSS 1/2 WORD
; 5223 Q_Q*2 ;AND INTO FINAL POSITION
; 5224 [T1][T1].AND.# CLR RH, ;CLEAR ALL BUT HIGH 16-BIT WORD
; 5225 #/777774,J/BLTXV ;FROM T1 AND CONTINUE
; 5226
```

U 0670, 3035,3771,0006,0276,6007,0700,0010,0000,0000
U 0672, 3054,3440,0505,0174,4007,0700,0400,0000,0000

U 3054, 0006,3772,0000,4365,5003,7700,0200,0000,0002

IT
U 0006, 3062,3446,1200,4174,4007,0700,0000,0000,0000

U 0007, 0570,4665,0017,4374,4007,0700,0000,0000,0377

U 0570, 3067,4443,0000,4174,4007,0700,1010,0071,1767

U 0571, 3070,4443,0000,4174,4007,0700,1010,0071,1772
U 0573, 3055,4662,0000,4374,4007,0700,0000,0000,1774

U 3055, 3056,3001,1717,4174,4007,0700,0000,0000,0000

U 3056, 3057,5742,1700,4174,4007,0700,0000,0000,0000
U 3057, 3060,3444,0012,4174,4007,0700,0000,0000,0000
U 3060, 3061,3444,0012,4174,4007,0700,0000,0000,0000

U 3061, 3066,4521,1717,4374,4007,0700,0000,0077,7774

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984 MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 142
UBABLT - BLT BYTES TO/FROM UNIBUS FORMAT

; 5227
U 3062, 3063,3446,1200,4174,4007,0700,0000,0000,0000 ; 5228
U 3063, 3064,3446,1200,4170,4007,0700,0000,0000,0000 ; 5229
U 3064, 3065,3446,1200,4170,4007,0700,0000,0000,0000 ; 5230
; 5231
U 3065, 0610,4665,0017,4374,4007,0700,0000,0000,0377
ION
; 5232
U 0610, 3067,4443,0000,4174,4007,0700,1010,0071,1771 ; 5233
; 5234
U 0611, 3070,4443,0000,4174,4007,0700,1010,0071,1770 ; 5235
U 0613, 3066,4662,0000,4374,4007,0700,0000,0000,0377 ; 5236
; 5237
; 5238
; 5239
U 3066, 3071,3002,1700,4174,4007,0700,0000,0000,0000 ; 5240
; 5241
U 3067, 0001,3445,1717,4174,4004,1700,1020,0041,0001 ; 5242
U 3070, 0002,3446,1200,4174,4004,1700,1020,0041,0001 ; 5243
; 5244
; 5245
U 3071, 3072,3443,0600,4174,4007,0700,0200,0003,0312 ; 5246
U 3072, 3073,3223,0000,4174,4007,0701,0200,0000,0002 ; 5247
U 3073, 1306,2113,0603,4174,4007,0521,4000,0000,0000 ; 5248
U 1306, 1400,4221,0013,4170,4007,0700,0000,0000,0000 ; 5249
U 1307, 3074,0111,0706,4174,4007,0700,0000,0000,0000 ; 5250
; 5251
; 5252
U 3074, 3054,0111,0704,4170,4007,0700,0200,0004,0712 ; 5253
; 5254
; 5255

BLTBU1: Q_Q*.5 ;NOW IN 1/2 WORDS
Q_Q*.5,HOLD LEFT ;INSERT A NULL BIT IN RH
Q_Q*.5,HOLD LEFT ;ONE MORE - NOW IN HALF WORDS
AD/D.AND.Q,DBUS/DBM, ;BUT NOT SWAPPED. COPY RIGHT BYTE
DBM/#,#/377,DEST/AD*2,B/T1 ;TO T1 AND SHIFT LEFT 1 POSI
=00 FE_S#,S#/1771, ; -7 BITS MORE
CALL [T1LSH] ;TO FINAL RESTING PLACE
=01 FE_S#,S#/1770, ; -8. LEFT BYTES MOVE RIGHT
CALL [Q_RSH] ;TO FINAL RESTING PLACE
=11 Q_Q.AND.#,#/377 ;WANT ONLY THE NEW BYTES
=
BLTXV: Q [T1].OR.Q, ;MERGE RESULTS
J/BLTXW ;AND STUFF IN MEMORY
T1LSH: [T1]_[T1]*2,SHIFT,RETURN [1]
Q_RSH: Q_Q*.5,SHIFT,RETURN [2]
BLTXW: VMA [BRX],START WRITE, ;DEST TO VMA
PXCT BLT DEST
MEM WRITE, MEM_Q ;STORE
[BRX]-[AR],3T,SKIP DPO ;DONE?
=0 END BLT,J/DONE ;YES
[BRX]_[BRX]+1 ;NO, INC DEST
VMA [ARX]+1,LOAD VMA, ; AND SOURCE (LOADING VMA)
PXCT BLT SRC,START READ, ;START UP MEMORY
J/BLTXLP ;AND CONTINUE WITH NEXT WORD
.ENDIF/UBABLT

; T1OKI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 143
FLOATING POINT -- FAD, FSB

```
; 5256 .TOC "FLOATING POINT -- FAD, FSB"
; 5257
; 5258 .DCODE
; 5259 140: FL-R, FL-AC, J/FAD
; 5260 142: FL-RW, FL-MEM, J/FAD
; 5261 FL-RW, FL-BOTH, J/FAD
; 5262 FL-R, FL-AC, ROUND, J/FAD
; 5263 FL-I, FL-AC, ROUND, J/FAD
; 5264 FL-RW, FL-MEM, ROUND, J/FAD
; 5265 FL-RW, FL-BOTH, ROUND, J/FAD
; 5266
; 5267 150: FL-R, FL-AC, J/FSB
; 5268 152: FL-RW, FL-MEM, J/FSB
; 5269 FL-RW, FL-BOTH, J/FSB
; 5270 FL-R, FL-AC, ROUND, J/FSB
; 5271 FL-I, FL-AC, ROUND, J/FSB
; 5272 FL-RW, FL-MEM, ROUND, J/FSB
; 5273 FL-RW, FL-BOTH, ROUND, J/FSB
; 5274 .UCODE
; 5275
; 5276 ;BOTH FAD & FSB ARE ENTERED WITH THE MEMORY OPERAND IN AR
; 5277 ; SIGN SMEARED. THE EXPONENT IN BOTH SC AND FE.
; 5278 1576:
; 5279 FSB: [AR]_[AR] ;MAKE MEMOP NEGATIVE
; 5280
; 5281 1577:
; 5282 FAD: [BR]_AC, SC_SC-EXP-1, 3T, SCAD DISP
; 5283 =0*
; 5284 FAS1: READ [BR], SKIP DPO, J/FAS2 ;BR .LE. AR
; 5285 [ARX]_[AR] ;SWAP AR AND BR
; 5286 [AR]_[BR], SC_EXP
; 5287 [BR]_[ARX], SC_SC-FE-1 ;NUMBER OF SHIFT STEPS
; 5288 READ [AR], FE_EXP, 2T, SKIP DPO
; 5289 =0 [AR]_+SIGN, J/FAS3
; 5290 [AR]_-SIGN, J/FAS3
; 5291
; 5292 =0 ;SIGN SMEAR BR AND UNNORMALIZE
; 5293 FAS2: [BR]_+SIGN, J/FAS3
; 5294 [BR]_-SIGN, J/FAS3
; 5295
; 5296 FAS3: Q_O, STEP SC
; 5297 =0
; 5298 FAS4: [BR]_[BR]*.5 LONG, STEP SC, ASHC, J/FAS4
; 5299 [AR]_[AR]+[BR], NORM DISP, J/SNORM
; 5300
```

D 0140, 0701, 1577, 1100
D 0142, 0702, 1577, 1700
D 0143, 0703, 1577, 1700
D 0144, 0711, 1577, 1100
D 0145, 0611, 1577, 0100
D 0146, 0712, 1577, 1700
D 0147, 0713, 1577, 1700

D 0150, 0701, 1576, 1100
D 0152, 0702, 1576, 1700
D 0153, 0703, 1576, 1700
D 0154, 0711, 1576, 1100
D 0155, 0611, 1576, 0100
D 0156, 0712, 1576, 1700
D 0157, 0713, 1576, 1700

U 1576, 1577, 2441, 0303, 4174, 4007, 0700, 4000, 0000, 0000

U 1577, 0674, 3771, 0005, 0276, 6006, 7701, 2000, 0020, 2000

U 0674, 1312, 3333, 0005, 4174, 4007, 0520, 0000, 0000, 0000
U 0676, 3075, 3441, 0304, 4174, 4007, 0700, 0000, 0000, 0000
U 3075, 3076, 3441, 0503, 4174, 4007, 0700, 2000, 0041, 2000
U 3076, 3077, 3441, 0405, 4174, 4007, 0700, 2000, 0020, 0000
U 3077, 1310, 3333, 0003, 4174, 4007, 0520, 1000, 0041, 2000
U 1310, 3100, 4551, 0303, 4374, 0007, 0700, 0000, 0000, 0777
U 1311, 3100, 3551, 0303, 4374, 0007, 0700, 0000, 0077, 7000

U 1312, 3100, 4551, 0505, 4374, 0007, 0700, 0000, 0000, 0777
U 1313, 3100, 3551, 0505, 4374, 0007, 0700, 0000, 0077, 7000

U 3100, 1314, 4222, 0000, 4174, 4007, 0630, 2000, 0060, 0000

U 1314, 1314, 3446, 0505, 4174, 4047, 0630, 2000, 0060, 0000
U 1315, 0420, 0111, 0503, 4174, 4003, 4701, 0000, 0000, 0000

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 144
FLAOTING POINT -- FMP

```

; 5301 .TOC "FLAOTING POINT -- FMP"
; 5302
; 5303 .DCODE
D 0160, 0701,1570,1100 ; 5304 160: FL-R, FL-AC, J/FMP
D 0162, 0702,1570,1700 ; 5305 162: FL-RW, FL-MEM, J/FMP
D 0163, 0703,1570,1700 ; 5306 FL-RW, FL-BOTH, J/FMP
; 5307
D 0164, 0711,1570,1100 ; 5308 FL-R, FL-AC, ROUND, J/FMP
D 0165, 0611,1570,0100 ; 5309 FL-I, FL-AC, ROUND, J/FMP
D 0166, 0712,1570,1700 ; 5310 FL-RW, FL-MEM, ROUND, J/FMP
D 0167, 0713,1570,1700 ; 5311 FL-RW, FL-BOTH, ROUND, J/FMP
; 5312 .UCODE
; 5313
; 5314 1570:
; 5315 FMP: [BRX]_AC, ;GET AC
; 5316 FE_SC+EXP, 3T, ;EXPONENT OF ANSWER
; 5317 SKIP DPO ;GET READY TO SMEAR SIGN
; 5318 =0 [BRX]_+SIGN, J/FMP1 ;POSITIVE
; 5319 [BRX]_-SIGN, J/FMP1 ;NEGATIVE
; 5320 FMP1: Q_[AR], SC 27. ;GET MEMORY OPERAND
; 5321 =01* [BRX]_[BRX]*.5 LONG, ;SHIFT RIGHT
; 5322 CALL [MULSUB] ;MULTIPLY
; 5323 Q_Q.AND.#, #/777000, ;WE ONLY COMPUTED
; 5324 HOLD LEFT ; 27 BITS
; 5325 [AR]_[ARX], FE_FE+2 ;SET SHIFT PATHS
; 5326 [AR]_[AR]*.5 LONG, ;SHIFT OVER
; 5327 FE_FE-200, ;ADJUST EXPONENT
; 5328 NORM DISP, J/SNORM ;NORMALIZE & EXIT
; 5329

U 1570, 1316,3771,0006,0276,6007,0521,1000,0040,2000
U 1316, 3101,4551,0606,4374,0007,0700,0000,0000,0777
U 1317, 3101,3551,0606,4374,0007,0700,0000,0077,7000
U 3101, 0163,3442,0300,4174,4007,0700,2000,0071,0033

U 0163, 2662,3446,0606,4174,4007,0700,0010,0000,0000

U 0167, 3102,4662,0000,4370,4007,0700,0000,0077,7000
U 3102, 3103,3441,0403,4174,4007,0700,1000,0041,0002

U 3103, 0420,3446,0303,4174,4003,4701,1000,0041,1600
```

; T1OKI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 145
FLOATING POINT -- FDV

```
; 5330 .TOC "FLOATING POINT -- FDV"
; 5331
; 5332 .DCODE
D 0170, 0701, 1574, 1100 ; 5333 170: FL-R, FL-AC, J/FDV
D 0172, 0702, 1574, 1700 ; 5334 172: FL-RW, FL-MEM, J/FDV
D 0173, 0703, 1574, 1700 ; 5335 FL-RW, FL-BOTH, J/FDV
; 5336
D 0174, 0711, 1574, 1100 ; 5337 FL-R, FL-AC, ROUND, J/FDV
D 0175, 0611, 1574, 0100 ; 5338 FL-I, FL-AC, ROUND, J/FDV
D 0176, 0712, 1574, 1700 ; 5339 FL-RW, FL-MEM, ROUND, J/FDV
D 0177, 0713, 1574, 1700 ; 5340 FL-RW, FL-BOTH, ROUND, J/FDV
; 5341 .UCODE
; 5342
; 5343
; 5344 1574:
U 1574, 1320, 3441, 0305, 0174, 4007, 0621, 0000, 0000, 0000 ; 5345 FDV: [BR]_[AR], SKIP AD.EQ.O, AC ;COPY DIVSOR SEE IF 0
; 5346 =0
; 5347 [AR]_AC, FE_SC-EXP, SKIP DPO, ;GET AC & COMPUTE NEW
U 1320, 1322, 3771, 0003, 0276, 6007, 0520, 1000, 0030, 2000 ; 5348 J/FDVO ; EXPONENT
U 1321, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0071, 1000 ; 5349 FL NO DIVIDE ;DIVIDE BY ZERO
; 5350 =0
U 1322, 3104, 4551, 0303, 4374, 0007, 0700, 0000, 0000, 0777 ; 5351 FDV0: [AR]_+SIGN, J/FDV1
U 1323, 3105, 3551, 0303, 4374, 0007, 0700, 0000, 0077, 7000 ; 5352 [AR]_-SIGN, J/FDV2
U 3104, 3106, 3441, 0304, 4174, 4007, 0700, 1000, 0031, 0200 ; 5353 FDV1: [ARX]_[AR], FE_-FE+200, J/FDV3 ;COMPUTE 2*DVND
U 3105, 3106, 2441, 0304, 4174, 4007, 0700, 5000, 0031, 0200 ; 5354 FDV2: [ARX]_-[AR], FE_-FE+200, J/FDV3 ;ABSOLUTE VALUE
U 3106, 1324, 3445, 0506, 4174, 4007, 0520, 0000, 0000, 0000 ; 5355 FDV3: [BRX]_[BR]*2, SKIP DPO ;ABSOLUTE VALUE
; 5356 =0
U 1324, 1326, 2113, 0406, 4174, 4007, 0311, 4000, 0000, 0000 ; 5357 FDV4: [ARX]_-[BRX], SKIP CRYO, 3T, J/FDV5 ;FLOATING NO DIV?
U 1325, 1324, 2445, 0506, 4174, 4007, 0700, 4000, 0000, 0000 ; 5358 [BRX]_-[BR]*2, J/FDV4 ;FORCE ABSOLUTE VALUE
; 5359 =0
U 1326, 1330, 3447, 0606, 4174, 4007, 0700, 0000, 0000, 0000 ; 5360 FDV5: [BRX]_[BRX]*.5, J/FDV6 ;SHIFT BACK ARX
U 1327, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0071, 1000 ; 5361 FL NO DIVIDE ;UNNORMALIZED INPUT
; 5362 =0
; 5363 FDV6: [AR]_[AR]*2, ;DO NOT DROP A BIT
U 1330, 3567, 3445, 0303, 4174, 4007, 0700, 0010, 0000, 0000 ; 5364 CALL [SBRL] ;AT FDV7+1
U 1331, 0144, 2113, 0604, 4174, 4007, 0421, 4000, 0000, 0000 ; 5365 [BRX]_-[ARX], SKIP AD.LE.O ;IS ANSWER .LE. 1?
; 5366 =00100
U 0144, 2725, 4222, 0000, 4174, 4007, 0700, 2010, 0071, 0033 ; 5367 FDV7: Q_O, SC_27., CALL [DIVSGN] ;DIVIDE
U 0145, 0144, 3447, 0303, 4174, 4007, 0700, 1000, 0041, 0001 ; 5368 =00101 [AR]_[AR]*.5, FE_FE+1, J/FDV7 ;SCALE DV'END
; 5369 =01100
U 0154, 3107, 3227, 0003, 4174, 4007, 0700, 0000, 0000, 0000 ; 5370 FDV8: [AR]_Q*.5, J/FDV9 ;PUT ANSWER IN AR
; 5371 =01101 READ [AR], SKIP AD.EQ.O, ;-VE ANSWER, LOOK AT RMDR
; 5372 CALL [SETSN] ; SEE HOW TO NEGATE
; 5373 =01110 READ [AR], SKIP AD.EQ.O, ;-VE ANSWER, LOOK AT RMDR
; 5374 CALL [SETSN] ; SEE HOW TO NEGATE
U 0156, 2036, 3333, 0003, 4174, 4007, 0621, 0010, 0000, 0000 ; 5375 =01111 [AR]_Q*.5, J/FDV9 ;PUT ANSWER IN AR
U 0157, 3107, 3227, 0003, 4174, 4007, 0700, 0000, 0000, 0000 ; 5376 =11111 [AR]_-Q*.5, J/FDV9 ;ZERO RMDR
U 0177, 3107, 2227, 0003, 4174, 4007, 0700, 4000, 0000, 0000 ; 5377
; 5378 FDV9: Q_O, J/SNORMO ;GO NORMALIZE
; 5379
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; FLT.MIC[10,1141] 01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 146
FLOATING POINT -- FLTR, FSC

D 0127, 0011, 1616, 1100
D 0132, 0001, 1621, 2100

U 1616, 1332, 4553, 0300, 4374, 4007, 0321, 0000, 0077, 7000
U 1332, 1334, 2441, 0305, 4174, 4007, 0521, 4000, 0000, 0000
U 1333, 1341, 4222, 0000, 4174, 4007, 0700, 1000, 0071, 0233

U 1334, 1336, 4553, 0500, 4374, 4007, 0321, 0000, 0077, 7000
U 1335, 3110, 4222, 0000, 4174, 4007, 0700, 1000, 0071, 0244

U 1336, 3110, 4222, 0000, 4174, 4007, 0700, 1000, 0071, 0244
U 1337, 1341, 4222, 0000, 4174, 4007, 0700, 1000, 0071, 0233

U 3110, 1340, 3446, 0303, 4174, 4047, 0700, 2000, 0071, 0006

U 1340, 1340, 3446, 0303, 4174, 4047, 0630, 2000, 0060, 0000
U 1341, 0420, 3333, 0003, 4174, 4003, 4701, 0000, 0000, 0000

U 1621, 3111, 3333, 0003, 4174, 4007, 0700, 2000, 0041, 4000
U 3111, 3112, 4222, 0000, 0174, 4007, 0700, 0000, 0000, 0000
U 3112, 1342, 3771, 0003, 0276, 6007, 0520, 1000, 0040, 2000
U 1342, 1341, 4551, 0303, 4374, 0007, 0700, 0000, 0000, 0777
U 1343, 1341, 3551, 0303, 4374, 0007, 0700, 0000, 0077, 7000

```
; 5380 .TOC "FLOATING POINT -- FLTR, FSC"
; 5381
; 5382 .DCODE
; 5383 127: R, FL-AC, ROUND, J/FLTR
; 5384 132: I, FL-AC, J/FSC
; 5385 .UCODE
; 5386
; 5387 1616:
; 5388 FLTR: [AR].AND.#, #/777000, 3T, SKIP ADL.EQ.O ;SMALL POS NUMBER?
; 5389 =0 [BR]_[AR], SKIP DPO, 3T, J/FLTR1 ;NO--SEE IF MINUS
; 5390 Q_O, FE_S#, S#/233, J/SNORMO ;FITS IN 27 BITS
; 5391 =0
; 5392 FLTR1: [BR].AND.#, #/777000, 3T,
; 5393 SKIP ADL.EQ.O, J/FLTR1A ;SMALL NEGATIVE NUMBER
; 5394 Q_O, FE_S#, S#/244, J/FLTR2 ;LARGE POS NUMBER
; 5395 =0
; 5396 FLTR1A: Q_O, FE_S#, S#/244, J/FLTR2 ;BIG NUMBER
; 5397 Q_O, FE_S#, S#/233, J/SNORMO ;FITS IN 27 BITS
; 5398 ;AT THIS POINT WE KNOW THE NUMBER TAKES MORE THAN 27 BITS. WE JUST
; 5399 ; SHIFT 8 PLACES RIGHT AND NORMALIZE. WE COULD BE MORE CLEVER BUT
; 5400 ; THIS IS THE RARE CASE ANYWAY.
; 5401 FLTR2: [AR]_[AR]*.5 LONG, ASHC, SC_6 ;SHOVE OVER TO THE RIGHT
; 5402 =0
; 5403 FLTR3: [AR]_[AR]*.5 LONG, ASHC, ;SHIFT RIGHT 9 PLACES
; 5404 STEP SC, J/FLTR3 ; SO IT WILL FIT
; 5405 SNORMO: READ [AR], NORM DISP, J/SNORM ;NORMALIZE ANSWER
; 5406
; 5407
; 5408 1621:
; 5409 FSC: READ [AR], SC_SHIFT
; 5410 Q_O, AC ;DON'T SHIFT IN JUNK
; 5411 [AR]_AC, FE_SC+EXP, SKIP DPO ;SIGN SMEAR
; 5412 =0 [AR]_+SIGN, J/SNORMO
; 5413 [AR]_-SIGN, J/SNORMO
; 5414
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 147
FLOATING POINT -- FIX AND FIXR

```

; 5415 .TOC "FLOATING POINT -- FIX AND FIXR"
; 5416
; 5417 .DCODE
D 0122, 0701,1626,1100 ; 5418 122: FL-R, FL-AC, J/FIX
D 0126, 0711,1626,1100 ; 5419 126: FL-R, FL-AC,ROUND, J/FIX
; 5420 .UCODE
; 5421
; 5422 1626:
; 5423 FIX: Q_O, SCAD/A+B, SCADA/S#, ;CLEAR Q, SEE IF
; 5424 S#/1534, SCADB/FE, 3T, ; ANSWER FITS IN
; 5425 SCAD DISP ; 35 BITS.
U 1626, 0720,4222,0000,4174,4006,7701,0000,0041,1534 ; 5426 =0* SET AROV, J/NIDISP ;TOO BIG
U 0720, 0603,4443,0000,4174,4467,0700,0000,0041,1000 ; 5427 SC FE+S#, S#/1544, 3T, SCAD DISP ;NEED TO MOVE LEFT?
U 0722, 0724,4443,0000,4174,4006,7701,2000,0041,1544 ; 5428 =0* STEP SC, J/FIXL
U 0724, 1346,4443,0000,4174,4007,0630,2000,0060,0000 ; 5429 SC_S#-FE, S#/232 ;NUMBER OF PLACES TO SHIFT
U 0726, 3113,4443,0000,4174,4007,0700,2000,0031,0232 ; 5430 ; RIGHT
; 5431 STEP SC ;ALREADY THERE
; 5432 =0
; 5433 FIXR: [AR]_[AR]*.5 LONG, ASHC, ;SHIFT BINARY POINT
; 5434 STEP SC, J/FIXR ; TO BIT 35.5
U 1344, 1344,3446,0303,4174,4047,0630,2000,0060,0000 ; 5435 [BR]_[ONE]*.5, B DISP, J/FIXX ;WHICH KIND OF FIX?
U 1345, 0063,3447,0705,4174,4003,7700,0000,0000,0000 ; 5436
; 5437 =0
; 5438 FIXL: [AR]_[AR]*2, STEP SC, J/FIXL ;SHIFT LEFT
U 1346, 1346,3445,0303,4174,4007,0630,2000,0060,0000 ; 5439 AC_[AR], NEXT INST ;WE ARE NOW DONE
U 1347, 0100,3440,0303,0174,4156,4700,0400,0000,0000 ; 5440
; 5441 =0*11
; 5442 FIXX: READ [AR], SKIP DPO, J/FIXT ;FIX--SEE IF MINUS
U 0063, 1350,3333,0003,4174,4007,0520,0000,0000,0000 ; 5443 FIXX1: [AR]_[AR]+[BR], FL-EXIT ;FIXR--ROUND UP
U 0073, 1514,0111,0503,4174,4003,7700,0200,0003,0001 ; 5444 =0
; 5445 FIXT: AC [AR], NEXT INST ;FIX & +, TRUNCATE
U 1350, 0100,3440,0303,0174,4156,4700,0400,0000,0000 ; 5446 READ Q, SKIP AD.EQ.O ;NEGATIVE--ANY FRACTION?
U 1351, 1352,3223,0000,4174,4007,0621,0000,0000,0000 ; 5447 =0 [AR]_[AR]+1, FL-EXIT ;YES--ROUND UP
U 1352, 1514,0111,0703,4174,4003,7700,0200,0003,0001 ; 5448 [BR]_.NOT.[MASK], ;MAYBE--GENERATE .75
; 5449 J/FIXX1 ;ROUND UP IF BIT 36 OR
; 5450 ; 37 SET
; 5451
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 148
FLOATING POINT -- SINGLE PRECISION NORMALIZE

```
; 5452 .TOC "FLOATING POINT -- SINGLE PRECISION NORMALIZE"
; 5453
; 5454 ;NORMALIZE DISPATCH IS A 9-WAY DISPATCH. THE HARDWARE LOOKS AT
; 5455 ; 4 SIGNALS: DP=0, DP BIT 8, DP BIT 9, DP BIT -2. THE 9 CASES
; 5456 ; ARE:
; 5457
; 5458 ;          DP=0    DPO8    DPO9    DPO0    ACTION TO TAKE
; 5459 ;          0      0      0      0      SHIFT LEFT
; 5460 ;
; 5461 ;          0      0      0      1      NEGATE AND RETRY
; 5462 ;
; 5463 ;          0      0      1      0      ALL DONE
; 5464 ;
; 5465 ;          0      0      1      1      NEGATE AND RETRY
; 5466 ;
; 5467 ;          0      1      0      0      SHIFT RIGHT
; 5468 ;
; 5469 ;          0      1      0      1      NEGATE AND RETRY
; 5470 ;
; 5471 ;          0      1      1      0      SHIFT RIGHT
; 5472 ;
; 5473 ;          0      1      1      1      NEGATE AND RETRY
; 5474 ;
; 5475 ;          1      -      -      -      LOOK AT Q BITS
; 5476
; 5477 ;ENTER HERE WITH UNNORMALIZED NUMBER IN AR!Q. FE HOLDS THE NEW
; 5478 ; EXPONENT. CALL WITH NORM DISP
; 5479 =0000 ;9-WAY DISPATCH
; 5480 SNORM: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNORM
; 5481 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5482 READ [AR], NORM DISP, CALL [SROUND]
; 5483 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5484 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
; 5485 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5486 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
; 5487 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5488 READ Q, SKIP AD.EQ.O, J/SNORM1
; 5489 =1110 [AR]_EXP, J/FLEX
; 5490 =
; 5491 =0
; 5492 SNORM1: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNORM
; 5493 FLEX: FL-EXIT
; 5494

U 0420, 0420,3444,0303,4174,4063,4701,1000,0041,1777
U 0421, 1356,2222,0000,4174,4007,0311,4000,0000,0000
U 0422, 0262,3333,0003,4174,4003,4701,0010,0000,0000
U 0423, 1356,2222,0000,4174,4007,0311,4000,0000,0000
U 0424, 0262,3447,0303,4174,4007,0700,1010,0041,0001
U 0425, 1356,2222,0000,4174,4007,0311,4000,0000,0000
U 0426, 0262,3447,0303,4174,4007,0700,1010,0041,0001
U 0427, 1356,2222,0000,4174,4007,0311,4000,0000,0000
U 0430, 1354,3223,0000,4174,4007,0621,0000,0000,0000
U 0436, 1355,3770,0303,4324,0457,0700,0000,0041,0000

U 1354, 0420,3444,0303,4174,4063,4701,1000,0041,1777
U 1355, 1514,4443,0000,4174,4003,7700,0200,0003,0001
```

```

; T10KI.MCR[10,1141]      11:45 11-AUG-1984      MICRO 31(254)  KS10 MICROCODE V124, 27-JUL-84  Page 149
; FLT.MIC[10,1141]        01:46 20-MAR-1981      FLOATING POINT -- SINGLE PRECISION NORMALIZE

U 1356, 0440,7441,0303,4174,4003,4701,0000,0000,0000 ; 5495 =0
; 5496 SNNEG: [AR]_.NOT.[AR], NORM DISP, J/SNNORM ;NEGATE HIGH WORD
; 5497 ; (NO CARRY)
U 1357, 0440,2441,0303,4174,4003,4701,4000,0000,0000 ; 5498 [AR]_[AR], NORM DISP, J/SNNORM ;NEGATE HIGH WORD (W/CARRY)
; 5499 =0000
U 0440, 0440,3444,0303,4174,4063,4701,1000,0041,1777 ; 5500 SNNORM: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNNORM
U 0442, 0262,3333,0003,4174,4003,4701,0010,0000,0000 ; 5501 =0010 READ [AR], NORM DISP, CALL [SROUND]
U 0444, 0262,3447,0303,4174,4007,0700,1010,0041,0001 ; 5502 =0100 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
U 0446, 0262,3447,0303,4174,4007,0700,1010,0041,0001 ; 5503 =0110 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
U 0450, 0440,3444,0303,4174,4063,4701,1000,0041,1777 ; 5504 =1000 [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNNORM
U 0456, 0327,3770,0303,4324,0453,7700,0000,0041,0000 ; 5505 =1110 [AR]_EXP, B DISP
; 5506 =
U 0327, 1360,4553,1300,4374,4007,0321,0000,0000,2000 ; 5507 =0111 TL [FLG], FLG.SN/1, J/SNNOT
; 5508 [AR]_[AR].AND.[MASK], ;CLEAR ANY LEFT OVER BITS
; 5509 J/SNNOT1
U 0337, 1363,4111,1203,4174,4007,0700,0000,0000,0000 ; 5510 =0
; 5511 SNNOT: [AR]_.NOT.[AR], J/SNNOT2
U 1360, 3114,7441,0303,4174,4007,0700,0000,0000,0000 ; 5512 READ Q, SKIP AD.EQ.0
U 1361, 1362,3223,0000,4174,4007,0621,0000,0000,0000 ; 5513 [AR]_.NOT.[AR], J/SNNOT2
U 1362, 3114,7441,0303,4174,4007,0700,0000,0000,0000 ; 5514 SNNOT1: [AR]_[AR], J/SNNOT2 ;NORMAL NEGATE AND EXIT
U 1363, 3114,2441,0303,4174,4007,0700,4000,0000,0000 ; 5515 SNNOT2: [FLG]_O, FL-EXIT
U 3114, 1514,4221,0013,4174,4003,7700,0200,0003,0001 ; 5516
; 5517
; 5518
; 5519 .TOC "FLOATING POINT -- ROUND ANSWER"
; 5520
; 5521 =*01*
U 0262, 0407,3447,0705,4174,4003,7700,0000,0000,0000 ; 5522 SROUND: [BR]_[ONE]*.5, B DISP, J/SRND1
U 0266, 0262,3447,0303,4174,4007,0700,1000,0041,0001 ; 5523 [AR]_[AR]*.5, FE_FE+1, J/SROUND ;WE WENT TOO FAR
; 5524 =0111
U 0407, 0016,4443,0000,4174,4004,1700,0000,0000,0000 ; 5525 SRND1: RETURN [16] ;NOT ROUNDING INSTRUCTION
U 0417, 0302,0111,0503,4174,4003,4701,0000,0000,0000 ; 5526 [AR]_[AR]+[BR], NORM DISP
U 0302, 0016,4443,0000,4174,4004,1700,0000,0000,0000 ; 5527 =*01* RETURN [16]
U 0306, 0016,3447,0303,4174,4004,1700,1000,0041,0001 ; 5528 [AR]_[AR]*.5, FE_FE+1, RETURN [16]
; 5529
; 5529

```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 150
FLOATING POINT -- DFAD, DFSB

D 0110, 1100,1637,1100
D 0111, 1100,1635,1100

U 1635, 3115,2441,0404,4174,4007,0700,4000,0000,0000
U 3115, 1637,2441,0303,4174,4007,0700,0040,0000,0000
U 1637, 3116,4557,0006,1274,4007,0701,0000,0000,1441
U 3116, 1364,3777,0005,0274,4007,0521,2000,0020,2000
U 1364, 1366,5547,0505,0374,4007,0631,0000,0077,7400
U 1365, 1366,3547,0505,0374,4007,0631,0000,0077,7400
U 1366, 1370,3442,0600,4174,4007,0700,0000,0000,0000
U 1367, 3120,3771,0016,0276,6007,0700,2000,0041,2000

```
; 5530 .TOC "FLOATING POINT -- DFAD, DFSB"
; 5531
; 5532 .DCODE
; 5533 110: DBL FL-R, J/DFAD
; 5534 111: DBL FL-R, J/DFSB
; 5535 .UCODE
; 5536
; 5537 ;ENTER FROM A-READ CODE WITH:
; 5538 ;FE/ EXP
; 5539 ;SC/ EXP
; 5540 ;AR/ C(E) SHIFT RIGHT 2 PLACES
; 5541 ;ARX/ C(E+1) SHIFTED RIGHT 1 PLACE
; 5542 1635:
; 5543 DFSB: [ARX]_[ARX] ;NEGATE LOW WORD
; 5544 [AR]_[AR]-.25, MULTI PREC/1
; 5545 1637:
; 5546 DFAD: [BRX]_(AC[1].AND.[MAG])* .5, 3T ;GET LOW WORD
; 5547 [BR]_AC*.5, 3T, ;GET AC AND START TO SHIFT
; 5548 SC_SC-EXP-1, ;NUMBER OF PLACES TO SHIFT
; 5549 SKIP DPO ;SEE WHAT SIGN
; 5550 =0 [BR]_+SIGN*.5, 3T, ;SIGN SMEAR
; 5551 AC, SKIP/SC, J/DFAS1 ;SEE WHICH IS BIGGER
; 5552 [BR]_-SIGN*.5, 3T, ;SIGN SMEAR
; 5553 AC, SKIP/SC, J/DFAS1 ;SEE WHICH IS BIGGER
; 5554 =0
; 5555 DFAS1: Q [BRX], ;AR IS BIGGER
; 5556 J/DFAS2 ;ADJUST BR!Q
; 5557 [TO]_AC, ;BR IS BIGGER OR EQUAL
; 5558 SC_EXP, 2T, J/DFAS3 ;SET SC TO THAT EXPONENT
; 5559
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 151
FLOATING POINT -- DFAD, DFSB

U 1370, 0153,3441,0516,4174,4007,0700,0010,0000,0000
U 1371, 3117,3441,1605,4174,4007,0700,0000,0000,0000
U 3117, 3123,0002,0400,4174,4007,0700,0000,0000,0000

U 3120, 3121,3442,0400,4174,4007,0700,2000,0020,0000
U 3121, 1372,3333,0016,4174,4007,0700,1000,0041,2000
U 1372, 0153,3441,0316,4174,4007,0700,0010,0000,0000
U 1373, 3122,3441,1603,4174,4007,0700,0000,0000,0000
U 3122, 3123,0002,0600,4174,4007,0700,0000,0000,0000

U 3123, 3124,0116,0503,4174,4047,0700,0040,0000,0000
U 3124, 0433,3444,0303,4174,4046,2700,0000,0000,0000
U 0433, 3125,5111,1217,4174,4007,0700,0000,0000,0000
U 0437, 0433,0222,0000,4174,4007,0700,4000,0000,0000
U 3125, 3126,3444,0303,4174,4047,0700,1000,0041,1777
U 3126, 3127,3444,0303,4174,4047,0700,1000,0041,1777
U 3127, 2005,3002,1700,4170,4007,0700,0000,0000,0000

```
; 5560 ;HERE IF AR!ARX IS GREATER THAN BR!BRX
; 5561 =0
; 5562 DFAS2: [TO]_[BR], CALL [DFADJ] ;ADJUST BR!Q
; 5563 [BR]_[TO] ;PUT ANSWER BACK
; 5564 Q_Q+[ARX], J/DFAS5 ;ADD LOW WORDS
; 5565
; 5566 ;HERE IS BR!BRX IF GREATER THAN OR EQUAL TO AR!ARX
; 5567 DFAS3: Q_[ARX], ;SETUP TO SHIFT AR!ARX
; 5568 SC_SC-FE-1 ;COMPUTE # OF PLACES
; 5569 READ [TO], FE_EXP ;EXPONENT OF ANSWER
; 5570 =0 [TO]_[AR], CALL [DFADJ] ;ADJUST AR!Q
; 5571 [AR]_[TO] ;PUT ANSWER BACK
; 5572 Q_Q+[BRX], J/DFAS5 ;ADD LOW WORDS
; 5573
; 5574 ;BIT DIDDLE TO GET THE ANSWER (INCLUDING 2 GUARD BITS) INTO
; 5575 ; AR!Q
; 5576 DFAS5: [AR]_([AR]+[BR])* .5 LONG, ;ADD HIGH WORDS
; 5577 MULTI PREC/1, ASHC ;INJECT SAVED CRY2
; 5578 [AR]_[AR]*2 LONG, ;SHIFT BACK LEFT
; 5579 ASHC, MUL DISP ;SEE IF WE LOST A 1
; 5580 =1011
; 5581 DFAS6: [T1]_[T1].AND.NOT.[MASK], J/DFAS7
; 5582 Q_Q+.25, J/DFAS6
; 5583 DFAS7: [AR]_[AR]*2 LONG, ASHC, ;PUT IN GUARD BITS
; 5584 FE_FE-1
; 5585 [AR]_[AR]*2 LONG, ASHC,
; 5586 FE_FE-1
; 5587 Q_[T1].OR.Q, HOLD LEFT, J/DNORMO
; 5588
```


; T1OKI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 152
FLOATING POINT -- DFAD, DFSB

```
; 5589 ;SUBROUTINE TO ADJUST NUMBER IN TO!Q
; 5590 ;RETURNS 1 WITH
; 5591 ; TO!Q ADJUSTED
; 5592 ; FLG.SN=1 IF WE SHIFTED OUT ANY 1 BITS (STICKY BIT)
; 5593 ; T1 HAS Q TWO STEPS PRIOR TO BEING DONE
; 5594 DFADJ "STEP SC, ASHC, MUL DISP"
; 5595
; 5596 =O**11
; 5597 DFADJ: [TO]_[TO]*2 LONG, DIV, ;MOVE EVERYTHING 2 PLACES
; 5598 CALL [CLRSN]
; 5599 [TO]_[TO]*2 LONG, DIV
; 5600 [TO]_[TO]*2 LONG, DIV
; 5601 [TO]_[TO]*.5 LONG, ASHC, ;SHIFT AT LEAST 1 PLACE
; 5602 STEP SC
; 5603 =1010
; 5604 DFADJ1: [TO]_[TO]*.5 LONG, ;UNNORMALIZE TO!Q
; 5605 DFADJ, J/DFADJ1 ;LOOP TILL DONE
; 5606 DFADJ2: [T1]_Q, ;SAVE GUARD BITS
; 5607 MUL DISP, J/DFADJ5 ;LOOK AT LAST BIT
; 5608 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ3
; 5609 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ4
; 5610
; 5611 =0
; 5612 DFADJ3: [TO]_[TO]*.5 LONG, ASHC, STEP SC, J/DFADJ3
; 5613 DFADJ4: [T1]_Q ;SAVE 2 GUARD BITS
; 5614 =1011
; 5615 DFADJ5: [TO]_[TO]*.5 LONG, ASHC, J/DFADJ6
; 5616 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ5
; 5617 DFADJ6: [TO]_[TO]*.5 LONG, ASHC, RETURN [1]
; 5618
```

U 0153, 2037,3444,1616,4174,4067,0700,0010,0000,0000
U 0173, 3130,3444,1616,4174,4067,0700,0000,0000,0000
U 3130, 3131,3444,1616,4174,4067,0700,0000,0000,0000
U 3131, 0472,3446,1616,4174,4047,0630,2000,0060,0000
U 0472, 0472,3446,1616,4174,4046,2630,2000,0060,0000
U 0473, 0453,3221,0017,4174,4006,2700,0000,0000,0000
U 0476, 1374,3551,1313,4374,0007,0700,0000,0000,2000
U 0477, 1375,3551,1313,4374,0007,0700,0000,0000,2000
U 1374, 1374,3446,1616,4174,4047,0630,2000,0060,0000
U 1375, 0453,3221,0017,4174,4007,0700,0000,0000,0000
U 0453, 3132,3446,1616,4174,4047,0700,0000,0000,0000
U 0457, 0453,3551,1313,4374,0007,0700,0000,0000,2000
U 3132, 0001,3446,1616,4174,4044,1700,0000,0000,0000

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; FLT.MIC[10,1141] 01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 153
FLOATING POINT -- DFMP

D 0112, 1105, 1631, 1100

U 1631, 1376, 3442, 0400, 4174, 4007, 0700, 2000, 0071, 0006

U 1376, 1376, 3444, 0303, 4174, 4047, 0630, 2000, 0060, 0000
U 1377, 3133, 3446, 1200, 4174, 4007, 0700, 0000, 0000, 0000
U 3133, 3134, 4662, 0000, 4374, 0007, 0700, 0000, 0007, 7777
U 3134, 3135, 3221, 0005, 4174, 4007, 0700, 0000, 0000, 0000

U 3135, 0730, 4557, 0006, 1274, 4007, 0700, 0000, 0000, 1441
U 0730, 2663, 3447, 0606, 4174, 4007, 0700, 2010, 0071, 0043

U 0734, 0732, 3442, 0300, 4174, 4007, 0700, 2000, 0071, 0043
U 0732, 2664, 4443, 0000, 4174, 4007, 0700, 0010, 0000, 0000
U 0736, 3136, 3441, 0416, 4174, 4007, 0700, 0000, 0000, 0000
U 3136, 3137, 3227, 0004, 4174, 4007, 0700, 2000, 0011, 0000

U 3137, 2000, 3777, 0006, 0274, 4007, 0521, 1000, 0040, 2000
U 2000, 1012, 5547, 0606, 4374, 4007, 0701, 0000, 0077, 7400
U 2001, 1012, 3547, 0606, 4374, 4007, 0701, 0000, 0077, 7400

U 1012, 2664, 3442, 0500, 4174, 4007, 0700, 2010, 0071, 0043
U 1016, 3140, 3221, 0017, 4174, 4007, 0700, 0000, 0000, 0000
U 3140, 1062, 0111, 1604, 4174, 4007, 0700, 0000, 0000, 0000

U 1062, 2664, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043

```
; 5619 .TOC "FLOATING POINT -- DFMP"
; 5620
; 5621 .DCODE
; 5622 112: DBL FL-R, DAC, J/DFMP
; 5623 .UCODE
; 5624
; 5625 ;SAME ENTRY CONDITIONS AS DFAD/DFSB
; 5626 1631:
; 5627 DFMP: Q_[ARX], SC_6 ;SHIFT MEM OP 8 PLACES
; 5628 =0
; 5629 DFMP1: [AR]_[AR]*2 LONG, ASHC, ;SHIFT
; 5630 STEP SC, J/DFMP1
; 5631 Q_Q*.5
; 5632 Q_Q.AND.#, #/077777, HOLD RIGHT
; 5633 [BR]_Q ;COPY LOW WORD
; 5634 ;
; 5635 ; BRX * BR ==> C(E+1) * C(AC+1)
; 5636 ;
; 5637 [BRX]_(AC[1].AND.[MAG])*5 ;GET LOW AC
; 5638 =0** [BRX]_[BRX]*.5, SC_35., CALL [MULSB1]
; 5639 ;
; 5640 ; BRX * Q ==> C(E) * C(AC+1)
; 5641 ;
; 5642 Q_[AR], SC_35. ;GO MULT NEXT HUNK
; 5643 =0** CALL [MULTIPLY]
; 5644 [TO]_[ARX] ;SAVE PRODUCT
; 5645 [ARX]_Q*.5, SC_FE ;PUT IN NEXT STEP
; 5646 ;
; 5647 ; BRX * BR ==> C(AC) * C(E+1)
; 5648 ;
; 5649 [BRX]_AC*.5, ;PREPARE TO DO HIGH HALF
; 5650 FE_SC+EXP, ;EXPONENT ON ANSWER
; 5651 SKIP DPO, 3T
; 5652 =0 [BRX]_+SIGN*.5, 3T, J/DFMP2
; 5653 [BRX]_-SIGN*.5, 3T
; 5654 =0**
; 5655 DFMP2: Q_[BR], SC_35., CALL [MULTIPLY] ;GO MULTIPLY
; 5656 [T1]_Q ;SAVE FOR ROUNDING
; 5657 [ARX]_[ARX]+[TO] ;PREPARE FOR LAST MUL
; 5658 ;
; 5659 ; BRX * Q ==> C(AC) * C(E)
; 5660 ;
; 5661 =0** Q_[AR], SC_35., ;DO THE LAST MULTIPLY
; 5662 CALL [MULTIPLY] ;...
; 5663
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45.11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 154
FLOATING POINT -- DFMP

U 1066, 0243,3446,0403,4174,4007,0700,1000,0041,1576

U 0243, 2036,3333,0017,4174,4007,0621,0010,0000,0000
U 0263, 3141,3444,0303,4174,4047,0700,0000,0000,0000
U 3141, 3142,3447,0705,4174,4007,0700,0000,0000,0000
U 3142, 2002,4553,1700,4374,4007,0321,0000,0020,0000
U 2002, 2003,0002,0500,4174,4007,0700,0000,0000,0000
U 2003, 3143,3444,0303,4174,4047,0700,0000,0000,0000
U 3143, 2004,4553,1700,4374,4007,0321,0000,0010,0000
U 2004, 2005,0002,0500,4174,4007,0700,0000,0000,0000

U 2005, 0520,3333,0003,4174,4003,4701,1000,0041,0002

```
; 5664 ;OK, WE NOW HAVE THE PRODUCT IN ARX!Q!T1. ALL WE NEED TO DO
; 5665 ; IS SOME BIT DIDDLES TO GET EVERYTHING IN THE RIGHT PLACE
; 5666 [AR]_[ARX]*.5 LONG, ;SHIFT THE ANSWER
; 5667 FE_FE+S#, S#/1576 ;CORRECT EXPONENT
; 5668 =O**11 READ [T1], SKIP AD.EQ.O, ;SEE IF LOW ORDER 1
; 5669 CALL [SETSNI] ; BITS AROUND SOMEPLACE
; 5670 [AR]_[AR]*2 LONG, ASHC ;SHIFT LEFT
; 5671 [BR]_[ONE]*.5 ;PLACE TO INSTERT BITS
; 5672 TL [T1], #/200000 ;ANYTHING TO INJECT?
; 5673 =O Q_Q+[BR] ;YES--PUT IT IN
; 5674 [AR]_[AR]*2 LONG, ASHC ;MAKE ROOM FOR MORE
; 5675 TL [T1], #/100000 ;ANOTHER BIT NEEDED
; 5676 =O Q_Q+[BR] ;YES--PUT IN LAST BIT
; 5677 DNORMO: READ [AR], NORM DISP, ;SEE WHAT WE NEED TO DO
; 5678 FE_FE+S#, S#/2, J/DNORM ;ADJUST FOR INITIAL SHIFTS
; 5679
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 155
FLOATING POINT -- DFDV

```

; 5680 .TOC "FLOATING POINT -- DFDV"
; 5681
; 5682 .DCODE
D 0113, 1105,1636,1100 ; 5683 113: DBL FL-R, DAC, J/DFDV
; 5684 .UCODE
; 5685 1636:
U 1636, 0132,3441,0406,4174,4007,0700,0000,0000,0000 ; 5686 DFDV: [BRX]_[ARX] ;COPY OPERAND (COULD SAVE TIME
; 5687 ; WITH SEPERATE A-READ FOR DFDV)
U 0132, 2037,4221,0017,4174,4007,0700,0010,0000,0000 ; 5688 =1**10 [T1]_O, CALL [CLRSN] ;CLEAR FLAG
; 5689 [BR]_[AR], SKIP AD.LE.O, ;SEE IF POSITIVE
U 0133, 2006,3441,0305,1174,4007,0421,0000,0000,1441 ; 5690 AC[1] ;WARM UP RAM
; 5691 =0
; 5692 DFDV1: [ARX]_(AC[1].AND.[MAG])*0.5, ;POSITIVE--GET AC
U 2006, 3146,4557,0004,1274,4007,0700,0000,0000,1441 ; 5693 J/DFDV2 ; AND CONTINUE BELOW
U 2007, 3144,7441,1717,4174,4007,0700,0000,0000,0000 ; 5694 [T1]_.NOT.[T1] ;DV'SOR NEGATIVE (OR ZERO)
U 3144, 3145,2441,0606,4174,4007,0700,4000,0000,0000 ; 5695 [BRX]_-[BRX] ;NEGATE LOW WORD
; 5696 AD/-B-.25, B/BR, DEST/AD, ;NEGATE HIGH WORD
; 5697 MULTI PREC/1, 3T, ;ADDING IN CRYO2
; 5698 SKIP DPO, AC[1], ;SEE IF STILL NEGATIVE
; 5699 J/DFDV1
; ...
; 5700 DFDV2: [AR]_AC*.5, ;GET AC AND SHIFT
; 5701 FE SC-EXP, 3T, ;COMPUTE NEW EXPONENT
; 5702 SKIP DPO ;SEE IF NEGATIVE
U 3146, 2010,3777,0003,0274,4007,0521,1000,0030,2000 ; 5703 =0 [AR]_+SIGN*.5, 3T, J/DFDV3 ;POSITIVE
U 2010, 2012,5547,0303,4374,4007,0701,0000,0077,7400 ; 5704 [T1]_.NOT.[T1] ;NEGATIVE OR ZERO
U 2011, 3147,7441,1717,4174,4007,0700,0000,0000,0000 ; 5705 [AR]_-SIGN*.5, 3T ;SIGN SMEAR
U 3147, 3150,3547,0303,4374,4007,0701,0000,0077,7400 ; 5706 Q_-[ARX] ;NEGATE OPERAND
U 3150, 3151,2442,0400,4174,4007,0700,4000,0000,0000 ; 5707 [AR]_(-[AR]-.25)*.5 LONG, ;NEGATE HIGH WORD
; 5708 MULTI PREC/1, ;USE SAVED CARRY
; 5709 ASHC, J/DFDV4 ;CONTINUE BELOW
; 5710 =0
; 5711 DFDV3: Q_[ARX], ;COPY OPERAND
; 5712 CALL [DDIVS] ;SHIFT OVER
U 2012, 2724,3442,0400,4174,4007,0700,0010,0000,0000 ; 5713 DFDV4: [AR]_-[BR], 3T, SKIP DPO ;SEE IF OVERFLOW
U 2013, 2014,2113,0305,4174,4007,0521,4000,0000,0000 ; 5714 =0 FL NO DIVIDE
U 2014, 0603,4443,0000,4174,4467,0700,0000,0071,1000 ; 5715 [ARX]_Q ;START DIVISION
U 2015, 1054,3221,0004,4174,4007,0700,0000,0000,0000 ; 5716 =0* Q_O, SC_26., CALL [DBLDIV]
U 1054, 1240,4222,0000,4174,4007,0700,2010,0071,0032 ; 5717 [TO]_Q, SC_35.
U 1056, 2020,3221,0016,4174,4007,0700,2000,0071,0043 ; 5718 =0* Q_Q.AND.NOT.[MAG], ;SEE IF ODD
; 5719 SKIP AD.EQ.O, ;SKIP IF EVEN
; 5720 CALL [DBLDIV] ;GO DIVIDE
; 5721 Q_Q*.5 ;MOVE ANSWER OVER
; 5722 =
; 5723 [TO]_[TO]*2 LONG, ASHC, ;DO FIRST NORM STEP
; 5724 MUL DISP ; SEE IF A 1 FELL OUT
; 5725 =1011
; 5726 DFDV4A: READ [T1], SKIP DPO, ;SHOULD RESULT BE NEGATIVE
; 5727 FE S#-FE, S#/202, ;CORRECT EXPONENT
; 5728 J/DFDV4B ;LOOK BELOW
; 5729 Q_Q+.25, J/DFDV4A ;PUT BACK THE BIT
; 5730 =0
; 5731 DFDV4B: [AR]_[TO], NORM DISP, J/DNORM ;PLUS
U 2016, 0520,3441,1603,4174,4003,4701,0000,0000,0000 ; 5732 [AR]_[TO], NORM DISP, J/DNNORM ;MINUS
U 2017, 0200,3441,1603,4174,4003,4701,0000,0000,0000 ; 5733
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 156
FLOATING POINT -- DOUBLE PRECISION NORMALIZE

U 0520, 0520,3444,0303,4174,4043,4701,1000,0041,1777
U 0521, 2026,4553,1300,4374,4007,0321,0000,0000,2000

U 0522, 0322,3333,0003,4174,4003,4701,0010,0000,0000
U 0523, 2026,4553,1300,4374,4007,0321,0000,0000,2000

U 0524, 0322,3446,0303,4174,4047,0700,1010,0041,0001
U 0525, 2026,4553,1300,4374,4007,0321,0000,0000,2000

U 0526, 0322,3446,0303,4174,4047,0700,1010,0041,0001
U 0527, 2026,4553,1300,4374,4007,0321,0000,0000,2000

U 0530, 3154,4002,0000,4174,0007,0700,0000,0000,0000
U 0536, 3153,4221,0013,4174,4007,0700,0000,0000,0000

U 3153, 1515,4113,0400,1174,4007,0700,0400,0000,1441

U 3154, 2024,3223,0000,4174,4007,0621,0000,0000,0000

U 2024, 0520,3444,0303,4174,4043,4701,1000,0041,1777
U 2025, 1515,3440,0303,1174,4007,0700,0400,0000,1441

```
; 5734 .TOC "FLOATING POINT -- DOUBLE PRECISION NORMALIZE"  
; 5735  
; 5736 ;NORMALIZE AR!Q  
; 5737 ;DNORMO: READ [AR], NORM DISP, ;SEE WHAT WE NEED TO DO  
; 5738 ; FE_FE+S#, S#/2, J/DNORM ;ADJUST FOR INITIAL SHIFTS  
; 5739 =0000  
; 5740 DNORM: [AR][AR]*2 LONG, ;SHIFT LEFT  
; 5741 FE_FE-1, ASHC, ;ADJUST EXPONENT  
; 5742 NORM DISP, J/DNORM ;TRY AGAIN  
; 5743 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5744 READ [AR], NORM DISP, ;SEE IF WE WENT TOO FAR  
; 5745 CALL [DROUND] ; AND ROUND ANSWER  
; 5746 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5747 [AR][AR]*.5 LONG, ASHC,  
; 5748 FE_FE+1, CALL [DROUND]  
; 5749 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5750 [AR][AR]*.5 LONG, ASHC,  
; 5751 FE_FE+1, CALL [DROUND]  
; 5752 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5753 Q [MAG].AND.Q, ;HIGH WORD IS ZERO  
; 5754 HOLD RIGHT, J/DNORM1 ;GO TEST LOW WORD  
; 5755 =1110 [FLG]_O ;[122] CLEAR FLAG WORD  
; 5756 =  
; 5757 AC[1]_[ARX].AND.[MAG], ;STORE LOW WORD  
; 5758 J/STAC ;GO DO HIGH WORD  
; 5759  
; 5760  
; 5761 DNORM1: READ Q, SKIP AD.EQ.O ;TEST LOW WORD  
; 5762 =0 [AR][AR]*2 LONG, ;LOW WORD IS NON-ZERO  
; 5763 FE_FE-1, ASHC, ;ADJUST EXPONENT  
; 5764 NORM DISP, J/DNORM ;KEEP LOOKING  
; 5765 AC[1]_[AR], J/STAC ;WHOLE ANSWER IS ZERO  
; 5766
```

; T10KI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 157
FLOATING POINT -- DOUBLE PRECISION NORMALIZE

```
U 2026, 3155,7222,0000,4174,4007,0700,0000,0000,0000
U 2027, 2030,2222,0000,4174,4007,0511,4000,0000,0000
U 3155, 2030,4221,0013,4174,4007,0700,0000,0000,0000

U 2030, 0200,7441,0303,4174,4003,4701,0000,0000,0000
U 2031, 0200,2441,0303,4174,4003,4701,4000,0000,0000

U 0200, 0200,3444,0303,4174,4043,4701,1000,0041,1777
U 0202, 0322,3333,0003,4174,4003,4701,0010,0000,0000
U 0204, 0322,3446,0303,4174,4047,0700,1010,0041,0001
U 0206, 0322,3446,0303,4174,4047,0700,1010,0041,0001
U 0210, 3160,4002,0000,4174,0007,0700,0000,0000,0000
U 0216, 0630,4111,1204,4174,4007,0700,0000,0000,0000

U 0630, 3161,4111,0004,4174,4007,0700,0010,0000,0000
U 0632, 3156,6111,0004,4174,4007,0700,0000,0000,0000
U 0633, 2032,2441,0404,4174,4007,0561,4000,0000,0000
U 3156, 2032,4221,0013,4174,4007,0700,0000,0000,0000
U 2032, 3157,7333,0003,0174,4007,0700,0400,0000,0000
U 2033, 3157,2443,0300,0174,4007,0701,4400,0000,0000
U 3157, 0100,4113,0400,1174,4156,4700,0400,0000,1441
U 3160, 2034,3223,0000,4174,4007,0621,0000,0000,0000

U 2034, 0200,3444,0303,4174,4043,4701,1000,0041,1777
U 2035, 1515,3440,0303,1174,4007,0700,0400,0000,1441

; 5767 ;HERE TO NORMALIZE NEGATIVE D.P. RESULTS
; 5768 =0
; 5769 DNEG: Q_.NOT.Q, J/DNEG1 ;ONES COMP
; 5770 Q_-Q, SKIP CRY2, J/DNEG2
; 5771 DNEG1: [FLG]_O
; 5772 =0
; 5773 DNEG2: [AR]_.NOT.[AR], ;NO CARRY
; 5774 NORM DISP, J/DNNORM ;GO NORMALIZE
; 5775 [AR]_[AR], ;CARRY
; 5776 NORM DISP, J/DNNORM ;NORMALIZE
; 5777
; 5778 =000*
; 5779 DNNORM: [AR][AR]*2 LONG, ;SHIFT 1 PLACE
; 5780 FE FE-1, ASHC, ;ADJUST EXPONENT
; 5781 NORM DISP, J/DNNORM ;LOOP TILL DONE
; 5782 =001* READ [AR], NORM DISP, ;SEE IF WE WENT TOO FAR
; 5783 CALL [DROUND] ; AND ROUND ANSWER
; 5784 =010* [AR][AR]*.5 LONG, ASHC,
; 5785 FE FE+1, CALL [DROUND]
; 5786 =011* [AR]_[AR]*.5 LONG, ASHC,
; 5787 FE FE+1, CALL [DROUND]
; 5788 =100* Q [MAG].AND.Q, ;HIGH WORD IS ZERO
; 5789 HOLD RIGHT, J/DNNRM1 ;GO TEST LOW WORD
; 5790 =111* [ARX]_[ARX].AND.[MASK] ;REMOVE ROUNDING BIT
; 5791 =
; 5792 =00 [ARX]_[ARX].AND.[MAG], ;ALSO CLEAR SIGN
; 5793 CALL [CHKSN] ;ONES COMP?
; 5794 =10 [ARX]_[ARX].XOR.[MAG], ;YES--ONES COMP
; 5795 J/DNN1 ;CONTINUE BELOW
; 5796 =11 [ARX]_[ARX], 3T, ;NEGATE RESULT
; 5797 SKIP CRY1, J/DNN2
; 5798 =
; 5799 DNN1: [FLG]_O ;CLEAR FLAG
; 5800 =0
; 5801 DNN2: AC_.NOT.[AR], J/DNORM2
; 5802 AC_-[AR], 3T
; 5803 DNORM2: AC[1]_[ARX].AND.[MAG], ;STORE LOW WORD
; 5804 NEXT INST ;ALL DONE
; 5805
; 5806 DNNRM1: READ Q, SKIP AD.EQ.O ;TEST LOW WORD
; 5807 =0 [AR]_[AR]*2 LONG, ;LOW WORD IS NON-ZERO
; 5808 FE FE-1, ASHC, ;ADJUST EXPONENT
; 5809 NORM DISP, J/DNNORM ;KEEP LOOKING
; 5810 AC[1]_[AR], J/STAC ;WHOLE ANSWER IS ZERO
; 5811
```

; T1OKI.MCR[10,1141]
; FLT.MIC[10,1141]

11:45 11-AUG-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 158
FLOATING POINT -- DOUBLE PRECISION NORMALIZE

U 3161, 0002,4553,1300,4374,4004,1321,0000,0000,2000

; 5812 CHKSN: TL [FLG], FLG.SN/1, RETURN [2]
; 5813
; 5814 ;SUBROUTINE TO SET/CLEAR FLG.SN
; 5815 ;CALL WITH:
; 5816 ; CALL [SETSN], SKIP IF WE SHOULD CLEAR
; 5817 ;RETURNS 23
; 5818 =0

U 2036, 0023,3551,1313,4374,0004,1700,0000,0000,2000
U 2037, 0023,5551,1313,4374,0004,1700,0000,0000,2000

; 5819 SETSN: [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, RETURN [23]
; 5820 CLRSN: [FLG]_[FLG].AND.NOT.#, FLG.SN/1, HOLD RIGHT, RETURN [23]
; 5821
; 5822

; 5823 ;SUBROUTINE TO ROUND A FLOATING POINT NUMBER
; 5824 ;CALL WITH:
; 5825 ; NUMBER IN AR!Q AND NORM DISP
; 5826 ;RETURNS 16 WITH ROUNDED NUMBER IN AR!ARX
; 5827 ;

U 0322, 0462,0007,0704,4174,4007,0511,0000,0000,0000

; 5828 =*O1*
; 5829 DROUND: [ARX]_(Q+1)*.5, ;ROUND AND SHIFT
; 5830 SKIP CRY2, ;SEE IF OVERFLOW
; 5831 J/DRND1 ;COMPLETE ROUNDING
; 5832 [AR]_[AR]*.5 LONG, ;WE WENT TOO FAR
; 5833 FE_FE+1, ASHC, J/DROUND ;SHIFT BACK AND ROUND
; 5834 =*O10

U 0326, 0322,3446,0303,4174,4047,0700,1000,0041,0001

; 5835 DRND1: [AR]_EXP, RETURN [16] ;NO OVERFLOW
; 5836 =O11 [AR]_[AR]+.25, ;ADD CARRY (BITS 36 AND 37
; 5837 ; ARE COPIES OF Q BITS)
; 5838 ;SEE IF OVERFLOW
; 5839 NORM DISP,
; 5840 J/DRND1 ; ..

U 0463, 0462,0441,0303,4174,4003,4701,4000,0000,0000

; 5841 [AR]_[AR]*.5, ;SHIFT RIGHT
; 5842 FE_FE+1, ;KEEP EXP RIGHT
; 5843 J/DRND1 ;ALL SET NOW
; 5844 =

U 0466, 0462,3447,0303,4174,4007,0700,1000,0041,0001

; 5843 =
; 5844

; T10KI.MCR[10,1141]
; EXTEND.MIC[10,1141]

11:45 11-AUG-1984
11:35 26-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 159
EXTEND -- DISPATCH ROM ENTRIES

```

; 5845 .TOC "EXTEND -- DISPATCH ROM ENTRIES"
; 5846
; 5847 .DCODE
D 0001, 0001, 1740, 2100 ; 5848 001: I, SJCL, J/L-CMS
D 0002, 0002, 1740, 2100 ; 5849 I, SJCE, J/L-CMS
D 0003, 0003, 1740, 2100 ; 5850 I, SJCLE, J/L-CMS
D 0004, 0002, 1741, 2100 ; 5851 I, B/2, J/L-EDIT
D 0005, 0005, 1740, 2100 ; 5852 I, SJCGE, J/L-CMS
D 0006, 0006, 1740, 2100 ; 5853 I, SJCN, J/L-CMS
D 0007, 0007, 1740, 2100 ; 5854 I, SJCG, J/L-CMS
; 5855
D 0010, 0001, 1742, 2100 ; 5856 010: I, B/1, J/L-DBIN ;CVTDBO
D 0011, 0004, 1742, 2100 ; 5857 I, B/4, J/L-DBIN ;CVTDBT
D 0012, 0001, 1743, 2100 ; 5858 I, B/1, J/L-BDEC ;CVTBDO
D 0013, 0000, 1743, 2100 ; 5859 I, B/O, J/L-BDEC ;CVTBDT
; 5860
D 0014, 0001, 1744, 2100 ; 5861 014: I, B/1, J/L-MVS ;MOVSO
D 0015, 0000, 1744, 2100 ; 5862 I, B/O, J/L-MVS ;MOVST
D 0016, 0002, 1744, 2100 ; 5863 I, B/2, J/L-MVS ;MOVSLJ
D 0017, 0003, 1744, 2100 ; 5864 I, B/3, J/L-MVS ;MOVSRJ
; 5865
D 0020, 0000, 1746, 2100 ; 5866 020: I, J/L-XBLT ;XBLT
D 0021, 0000, 1747, 2100 ; 5867 I, J/L-SPARE-A ;GSNGL
D 0022, 0000, 1750, 2100 ; 5868 I, J/L-SPARE-B ;GDBLE
D 0023, 0000, 1751, 2100 ; 5869 I, B/O, J/L-SPARE-C ;GDFIX
D 0024, 0001, 1751, 2100 ; 5870 I, B/1, J/L-SPARE-C ;GDFIXR
D 0025, 0002, 1751, 2100 ; 5871 I, B/2, J/L-SPARE-C ;GDFIXR
D 0026, 0004, 1751, 2100 ; 5872 I, B/4, J/L-SPARE-C ;GDFIXR
D 0027, 0010, 1751, 2100 ; 5873 I, B/10, J/L-SPARE-C ;DGFLTR
; 5874 ;30: ;GFLTR
; 5875 ;GFSC
; 5876 .UCODE
; 5877
```


; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 160
EXTEND -- DISPATCH ROM ENTRIES

U 1740, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5878 1740:
; 5879 L-CMS: LUUD
U 1741, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5880 1741:
; 5881 L-EDIT: LUUD
U 1742, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5882 1742:
; 5883 L-DBIN: LUUD
U 1743, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5884 1743:
; 5885 L-BDEC: LUUD
U 1744, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5886 1744:
; 5887 L-MVS: LUUD
U 1746, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5888 1746:
; 5889 L-XBLT: LUUD
U 1747, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5890 1747:
; 5891 L-SPARE-A: LUUD
U 1750, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5892 1750:
; 5893 L-SPARE-B: LUUD
U 1751, 0400,4751,1203,4374,4007,0700,0000,0000,0040 ; 5894 1751:
; 5895 L-SPARE-C: LUUD
; 5896
; 5897 ;NOTE: WE DO NOT NEED TO RESERVE 3746 TO 3751 BECAUSE THE CODE
; 5898 ; AT EXTEND DOES A RANGE CHECK.
; 5899

```
; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984
```

```
MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 161
EXTEND -- INSTRUCTION SET DECODING
```

```
D 0123, 0000, 1467, 3100

U 1467, 2040, 3771, 0005, 4365, 5007, 0700, 0200, 0000, 0002
U 2040, 3556, 4553, 0500, 4374, 4007, 0321, 0010, 0076, 0740
U 2044, 3162, 4521, 0206, 4374, 4007, 0700, 0000, 0000, 0740
U 3162, 3163, 3111, 0605, 4174, 0417, 0700, 0000, 0000, 0000
U 3163, 3164, 3333, 0005, 4174, 4217, 0700, 0000, 0000, 0500
U 3164, 3165, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0240
U 3165, 0170, 4443, 0000, 2174, 4006, 6700, 0000, 0000, 0000
U 0170, 0172, 0551, 0505, 2270, 4007, 0700, 0000, 0000, 0000
U 0172, 0556, 5741, 0505, 4174, 4003, 7700, 0200, 0000, 0010
U 0174, 3166, 0551, 0505, 2270, 4007, 0700, 0200, 0004, 0512
U 0176, 3166, 3443, 0500, 4174, 4007, 0700, 0200, 0004, 0512
U 3166, 3165, 3771, 0005, 4361, 5217, 0700, 0200, 0000, 0502

U 0556, 3400, 3333, 0005, 7174, 4001, 2700, 0400, 0000, 0241
U 0557, 2042, 3333, 0005, 4174, 4007, 0530, 0000, 0000, 0000
U 2042, 3400, 3333, 0005, 7174, 4001, 2700, 0400, 0000, 0241
U 2043, 0556, 3771, 0005, 4374, 0007, 0700, 0000, 0077, 7777
```

```
; 5900 .TOC "EXTEND -- INSTRUCTION SET DECODING"
; 5901
; 5902 ;EACH INSTRUCTION IN THE RANGE 1-23 GOES TO 1 OF 2 PLACES
; 5903 ; 1740-1747 IF NOT UNDER EXTEND
; 5904 ; 3740-3747 IF UNDER EXTEND
; 5905
; 5906 .DCODE
; 5907 123: I,READ/1, J/EXTEND
; 5908 .UCODE
; 5909
; 5910 1467:
; 5911 EXTEND: MEM READ, [BR]_MEM ;FETCH INSTRUCTION
; 5912 =0** TL [BR], #/760740, ;IN RANGE 0-17 (AND.AC#=0)
; 5913 CALL [BITCHK] ;TRAP IF NON-ZERO BITS FOUND
; 5914 [BRX]_[HR].AND.# CLR RH, ;SPLIT OUT AC NUMBER
; 5915 #/000740 ; FROM EXTEND INSTRUCTION
; 5916 [BR]_[BR].OR.[BRX], ;LOAD IR AND AC #
; 5917 HOLD RIGHT, LOAD IR ; ..
; 5918 READ [BR], LOAD BYTE EA, ;LOAD XR #
; 5919 J/EXTEAO ;COMPUTE E1
; 5920
; 5921 EXTEAO: WORK[EO]_[AR]
; 5922 EXTEA1: EA MODE DISP
; 5923 =100*
; 5924 EXTEA: [BR]_[BR]+XR
; 5925 EXTDSP: [BR]_EA FROM [BR], LOAD VMA,
; 5926 B DISP, J/EXTEXT
; 5927 [BR]_[BR]+XR, START READ, PXCT EXTEND EA, LOAD VMA, J/EXTIND
; 5928 VMA_[BR], START READ, PXCT EXTEND EA
; 5929
; 5930 EXTIND: MEM READ, [BR]_MEM, HOLD LEFT, LOAD BYTE EA, J/EXTEA1
; 5931
; 5932 ;HERE TO EXTEND SIGN FOR OFFSET MODES
; 5933 =1110
; 5934 EXTEXT: WORK[E1]_[BR], ;SAVE E1
; 5935 DISP/DROM, J/3400 ;GO TO EXTENDED EXECUTE CODE
; 5936 READ [BR], SKIP DP18 ;NEED TO EXTEND SIGN
; 5937 =0 WORK[E1]_[BR], ;POSITIVE
; 5938 DISP/DROM, J/3400
; 5939 [BR]_#, #/777777, HOLD RIGHT, ;NEGATIVE
; 5940 J/EXTEXT
; 5941
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 162
EXTEND -- MOVE STRING -- SETUP

```
; 5942 .TOC "EXTEND -- MOVE STRING -- SETUP"
; 5943
; 5944 ;HERE TO MOVE A STRING
; 5945 ;COME HERE WITH:
; 5946 ; AR/ EO
; 5947 ; BR/ E1
; 5948 ;
; 5949 3744:
; 5950 MVS: [AR]_[AR]+1, ;GO FETCH FILL
; 5951 LOAD VMA, ; BYTE
; 5952 START READ, ; ..
; 5953 CALL [GTFILL] ;SUBROUTINE TO COMPLETE
; 5954 3754: [BR]_AC[DLEN] ;GET DEST LENGTH AND FLAGS
; 5955 =O** TL [BR], #/777000, ;ANY FLAGS SET?
; 5956 CALL [BITCHK] ;SEE IF ILLEGAL
; 5957 [AR]_AC ;GET SRC LENGTH AND FLAGS
; 5958 =O [BRX]_[AR].AND.# CLR RH, ;COPY FLAGS TO BRX
; 5959 #/777000, ; ..
; 5960 CALL [CLRFLG] ;CLEAR FLAGS IN AR
; 5961 ;NEW DLEN IS <SRC LEN>--<DST LEN>
; 5962 AC[DLEN]_[AR]-[BR], 3T, ;COMPUTE DIFFERENCE
; 5963 SKIP DPO ;WHICH IS SHORTER?
; 5964 =O [AR]_.NOT.[BR], ;DESTINATION
; 5965 J/MVS1 ;GET NEGATIVE LENGTH
; 5966 [AR]_.NOT.[AR] ;SOURCE
; 5967 MVS1: WORK[SLEN]_[AR], ; ..
; 5968 B DISP ;SEE WHAT TYPE OF MOVE
; 5969 ;SLEN NOW HAS <LEN OF SHORTER STRING>-1
; 5970 =1100
; 5971 STATE_[SRC], J/MOVELP ;TRANSLATE--ALL SET
; 5972 [BR]_AC[DSTP], J/MVSO ;OFFSET BUILD MASK
; 5973 [ARX]_[AR], ;LEFT JUSTIFY
; 5974 J/MOVSTO ; ..
; 5975 [ARX]_AC[DLEN], ;RIGHT JUSTIFY
; 5976 SKIP DPO, 4T, ;WHICH IS SHORTER?
; 5977 J/MOVRJ
; 5978
; 5979 MVSO: READ [BR], FE_S+2 ;GET DST BYTE SIZE
; 5980 Q_O, BYTE STEP ;BUILD AN S BIT MASK
; 5981 =O*
; 5982 MVS01: GEN MSK [AR], BYTE STEP, J/MVS01
; 5983 [AR]_.NOT.Q ;BITS WHICH MUST NOT BE SET
; 5984 WORK[MSK]_[AR].AND.[MASK], ;SAVE FOR SRCMOD
; 5985 J/MOVLPO ;GO ENTER LOOP
; 5986
```

U 3744, 3361,0111,0703,4174,4007,0700,0210,0004,0012
U 3754, 2041,3771,0005,1276,6007,0701,0000,0000,1443
U 2041, 3556,4553,0500,4374,4007,0321,0010,0077,7000
U 2045, 2046,3771,0003,0276,6007,0700,0000,0000,0000
U 2046, 3363,4521,0306,4374,4007,0700,0010,0077,7000
U 2047, 2050,2113,0305,1174,4007,0521,4400,0000,1443
U 2050, 3167,7441,0503,4174,4007,0700,0000,0000,0000
U 2051, 3167,7441,0303,4174,4007,0700,0000,0000,0000
U 3167, 0574,3333,0003,7174,4003,7700,0400,0000,0242
U 0574, 0500,3771,0013,4370,4007,0700,0000,0000,0003
U 0575, 3170,3771,0005,1276,6007,0701,0000,0000,1444
U 0576, 3210,3441,0304,4174,4007,0700,0000,0000,0000
U 0577, 0650,3771,0004,1276,6007,0522,0000,0000,1443
U 3170, 3171,3333,0005,4174,4007,0700,1000,0041,6020
U 3171, 2021,4222,0000,4174,4006,7701,1000,0041,1770
U 2021, 2021,4224,0003,4174,4026,7701,1000,0041,1770
U 2023, 3172,7221,0003,4174,4007,0700,0000,0000,0000
U 3172, 0507,4113,0312,7174,4007,0700,0400,0000,0243

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 163
EXTEND -- MOVE STRING -- OFFSET/TRANSLATE

```
; 5987 .TOC "EXTEND -- MOVE STRING -- OFFSET/TRANSLATE"
; 5988
; 5989 ;HERE IS THE LOOP FOR OFFSET AND TRANSLATED MOVES
; 5990 =000
; 5991 MOVLP: [AR]_WORK[SLEN]+1, ;UPDATE STRING LENGTH
; 5992 CALL [SRCMOD] ;GET A SOURCE BYTE
; 5993 =001 [ARX]_[AR], SKIP DPO, ;(1) LENGTH EXHAUSTED
; 5994 J/MOVST2 ; SEE IF FILL IS NEEDED
; 5995 =100 [AR]_WORK[SLEN], ;(4) ABORT
; 5996 J/MVABT ;
; 5997 STATE [SRC+DST], ;(5) NORMAL--STORE DST BYTE
; 5998 CALL [PUTDST] ;
; 5999 =111
; 6000 MOVLP: STATE_[SRC], J/MOVLP ;(7) DPB DONE
; 6001 =
; 6002
; 6003 ;HERE TO ABORT A STRING MOVE DUE TO TRANSLATE OR OFFSET FAILURE
; 6004
; 6005 MVABT: [BR]_AC[DLEN], ;WHICH STRING IS LONGER
; 6006 SKIP DPO, 4T
; 6007 =0
; 6008 MVABT1: AC[DLEN]_[AR], J/MVABT2 ;PUT AWAY DEST LEN
; 6009 [AR]_[AR]-[BR], ;DEST LEN WAS GREATER
; 6010 J/MVABT1 ;STICK BACK IN AC
; 6011
; 6012 MVABT2: [AR]_.NOT.WORK[SLEN] ;GET UNDECREMENTED SLEN
; 6013 READ [BR], SKIP DPO ;NEED TO FIXUP SRC?
; 6014 =0 [AR]_[AR]+[BR] ;SRC LONGER BY (DLEN)
; 6015 MVEND: [AR]_[AR].OR.[BRX] ;PUT BACK SRC FLAGS
; 6016 END STATE, J/STAC ;ALL DONE
; 6017
```

U 0500, 1040,0551,0703,7274,4007,0701,0010,0000,0242
U 0501, 1000,3441,0304,4174,4007,0520,0000,0000,0000
U 0504, 3173,1771,0003,7274,4007,0701,4000,0000,0242
U 0505, 3353,3771,0013,4370,4007,0700,0010,0000,0005
U 0507, 0500,3771,0013,4370,4007,0700,0000,0000,0003

U 3173, 2052,3771,0005,1276,6007,0522,0000,0000,1443
U 2052, 3174,3440,0303,1174,4007,0700,0400,0000,1443
U 2053, 2052,1111,0503,4174,4007,0700,4000,0000,0000

U 3174, 3175,7771,0003,7274,4007,0701,0000,0000,0242
U 3175, 2054,3333,0005,4174,4007,0520,0000,0000,0000
U 2054, 2055,0111,0503,4174,4007,0700,0000,0000,0000
U 2055, 3176,3111,0603,4174,4007,0700,0000,0000,0000
U 3176, 1515,4221,0013,4170,4007,0700,0000,0000,0000

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 164
EXTEND -- MOVE STRING -- MOVSRJ

```

; 6018 .TOC "EXTEND -- MOVE STRING -- MOVSRJ"
; 6019
; 6020 =00
U 0650, 3177,3771,0003,1276,6007,0701,0000,0000,1441 ;MOVRJ: [AR]_AC[SRCP], J/MVSKP ;SRC LONGER, SKIP OVER SOME
; 6021 STATE [DSTF], ;DST LONGER, FILL IT
; 6022 CALL [MOVFIL] ;...
; 6023 =11 [ARX]_WORK[SLEN]+1, ;DONE FILLING
; 6024 J/MOVST1 ;GO MOVE STRING
; 6025
; 6026
; 6027 ;HERE TO SKIP OVER EXTRA SOURCE BYTES
U 3177, 2056,3440,0303,1174,4007,0670,0400,0000,1441 ;MVSKP: AC[SRCP]_[AR], SKIP -1MS ;[121] Is there a timer interrupt?
U 2056, 3202,3333,0003,7174,4007,0700,0400,0000,0211 ; 6028 =0 WORK[SV.AR]_[AR], J/MVSK2 ;[121][123] Yes, save regs for int
rrupt. ; 6029
; 6030 [ARX]_[ARX]-1, 3T, ;DONE SKIPPING?
; 6031 SKIP DPO
; 6032 =0 IBP DP, IBP SCAD, ;NO--START THE IBP
; 6033 SCAD DISP, SKIP IRPT, ;4-WAY DISPATCH
; 6034 3T, J/MVSKP1 ;GO BUMP POINTER
; 6035 AC[DLEN]_O, ;LENGTHS ARE NOW EQUAL
; 6036 J/MOVST4 ;GO MOVE STRING
; 6037
; 6038 =00
U 0750, 3177,3441,0503,4174,4007,0700,0000,0000,0000 ;MVSKP1: [AR]_[BR], J/MVSKP ;NO OVERFLOW
; 6039 [AR]_.NOT.WORK[SLEN], ;INTERRUPT
; 6040 J/MVSK3 ;...
; 6041 SET P TO 36-S, ;WORD OVERFLOW
; 6042 J/MVSKP2 ;FIXUP Y
; 6043 [AR]_.NOT.WORK[SLEN] ;[121] INTERRUPT or timer.
; 6044 MVSK3: AC[DLEN]_[AR] ;RESET DLEN
; 6045 =0 [AR]_[AR]+[ARX],
; 6046 CALL [INCAR] ;ADD 1 TO AR
; 6047 AC [AR].OR.[BRX], ;PUT BACK FLAGS
; 6048 J/ITRAP ;DO INTERRUPT TRAP
; 6049
; 6050
; 6051 MVSKP2: [AR]_[AR]+1, HOLD LEFT, ;BUMP Y
; 6052 J/MVSKP ;KEEP GOING
; 6053
; 6054
; 6055 MVSK2: WORK[SV.BR]_[BR] ;BEGIN EDIT [123]
; 6056 WORK[SV.ARX]_[ARX] ;SAVE ALL
; 6057 WORK[SV.BRX]_[BRX] ;THE REGISTERS
; 6058 =0* CALL [TICK] ;FOR THE TICK
; 6059 [AR]_WORK[SV.AR] ;UPDATE CLOCK AND SET INTERUPT
; 6060 [BR]_WORK[SV.BR] ;NOW PUT
; 6061 [ARX]_WORK[SV.ARX] ;THEM ALL
; 6062 [BRX]_WORK[SV.BRX], ;BACK SO WE
; 6063 J/MVSKP ;CAN CONTINUE
; 6064
; 6065 ;END EDIT [123]
; 6066
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 165
EXTEND -- MOVE STRING -- SIMPLE MOVE LOOP

```

; 6066 .TOC "EXTEND -- MOVE STRING -- SIMPLE MOVE LOOP"
; 6067
; 6068 ;HERE FOR NO-MODIFICATION STRING MOVES
U 3210, 3211,0111,0704,4174,4007,0700,0000,0000,0000 ; 6069 MOVST0: [ARX]_[ARX]+1 ;CANT DO [ARX]_[AR]+1
U 3211, 0540,3771,0013,4370,4007,0700,0000,0000,0003 ; 6070 MOVST1: STATE_[SRC] ;PREPARE FOR PAGE FAIL
; 6071 =000
; 6072 WORK[SLEN]_[ARX], ;GO GET A SOURCE BYTE
; 6073 SKIP DPO, CALL [GSR] ; ..
U 0540, 2256,3333,0004,7174,4007,0520,0410,0000,0242 ; 6074 MOVSTX: [ARX]_[AR], ;SHORT STRING RAN OUT
U 0541, 1000,3441,0304,4174,4007,0520,0000,0000,0000 ; 6075 SKIP DPO, J/MOVST2 ;GO SEE IF FILL NEEDED
; 6076 =010 STATE [SRC+DST], ;WILL NEED TO BACK UP BOTH POINTERS
U 0542, 3353,3771,0013,4370,4007,0700,0010,0000,0005 ; 6077 CALL [PUTDST] ;STORE BYTE
; 6078 =110
; 6079 MOVST4: [ARX]_WORK[SLEN]+1, ;COUNT DOWN LENGTH
; 6080 J/MOVST1 ;LOOP OVER STRING
; 6081 =
; 6082 =00
U 0546, 3211,0551,0704,7274,4007,0701,0000,0000,0242 ; 6083 MOVST2: AC[DLEN]_0, J/MOVST3 ;CLEAR DEST LEN, REBUILD SRC
U 1000, 3212,4223,0000,1174,4007,0700,0400,0000,1443 ; 6084 STATE [DST], CALL [MOVFIL] ;FILL OUT DEST
U 1001, 2245,3771,0013,4370,4007,0700,0010,0000,0004 ; 6085 =11 AC_[BRX], J/ENDSKP ;ALL DONE
U 1003, 2125,3440,0606,0174,4007,0700,0400,0000,0000 ; 6086
U 3212, 3213,3113,0406,0174,4007,0700,0400,0000,0000 ; 6087 MOVST3: AC_[ARX].OR.[BRX] ;REBUILD SRC
U 3213, 0252,4221,0013,4170,4007,0700,0000,0000,0000 ; 6088 END STATE, J/SKIPE ; ..
; 6089
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 166
EXTEND -- COMPARE STRING

```

; 6090 .TOC "EXTEND -- COMPARE STRING"
; 6091
; 6092 3740:
U 3740, 2070,3771,0004,1276,6007,0701,0000,0000,1443 ; 6093 CMS: [ARX]_AC[DLEN] ;GET DEST LEN
U 2070, 3556,4553,0400,4374,4007,0321,0010,0077,7000 ; 6094 =O** TL [ARX], #/777000, CALL [BITCHK]
U 2074, 2071,3771,0006,0276,6007,0700,0000,0000,0000 ; 6095 [BRX]_AC ;GET SRC LEN
U 2071, 3556,4553,0600,4374,4007,0321,0010,0077,7000 ; 6096 =O** TL [BRX], #/777000, CALL [BITCHK]
U 2075, 2072,2113,0604,4174,4007,0521,4000,0000,0000 ; 6097 [BRX]-[ARX], 3T, SKIP DPO ;WHICH STRING IS LONGER?
U 2072, 2073,0111,0703,4174,4007,0700,0000,0000,0000 ; 6098 =0 [AR]_[AR]+1 ;SRC STRING IS LONGER
U 2073, 2076,0111,0703,4170,4007,0700,0200,0004,0012 ; 6099 VMA [AR]+1, START READ ;DST STRING
; 6100 =0 [AR]_0, ;FORCE FIRST COMPARE TO BE
; 6101 ;EQUAL
U 2076, 3564,4221,0003,4174,4007,0700,0010,0000,0000 ; 6102 CALL [LOADQ] ;PUT FILL INTO Q
; 6103 WORK[FILL]_Q, ;SAVE FILLER
U 2077, 3223,3223,0000,7174,4007,0700,0400,0000,0244 ; 6104 J/CMS2 ;ENTER LOOP
; 6105
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 167
EXTEND -- COMPARE STRING

```

; 6106 ;HERE IS THE COMPARE LOOP.
; 6107 ; ARX/ CONTAINS REMAINING DEST LENGTH
; 6108 ; BRX/ CONTAINS REMAINING SOURCE LENGTH
; 6109 =0
; 6110 CMS3: ;BYTES ARE NOT EQUAL
; 6111 END STATE, ;NO MORE SPECIAL PAGE FAIL ACTION
; 6112 SKIP-COMP DISP ;SEE SKIP-COMP-TABLE
U 2100, 0250,4221,0013,4170,4003,7700,0000,0000,0000 ; 6113 CMS4: [AR]_AC[SRCP] ;GET BYTE POINTER
U 2101, 3214,3771,0003,1276,6007,0701,0000,0000,1441 ; 6114 READ [BRX], SKIP DPO ;MORE IN SOURCE STRING?
U 3214, 1004,3333,0006,4174,4007,0520,0000,0000,0000 ; 6115 =00 STATE_[EDIT-SRC], ;PREPARE FOR PAGE FAIL
; 6116 CALL [GETSRC] ; GO GET BYTE
U 1004, 2257,3771,0013,4370,4007,0700,0010,0000,0011 ; 6117 READ [ARX], SKIP DPO, ;NO MORE SRC--SEE IF MORE DEST
; 6118 J/CMS5 ;
; 6119 WORK[CMS]_[AR] ;SAVE SRC BYTE
; 6120 =
; 6121 AC_[BRX] ;PUT BACK SRC LEN
; 6122 STATE_[COMP-DST] ;HAVE TO BACK UP IF DST FAILS
U 1005, 2102,3333,0004,4174,4007,0520,0000,0000,0000 ; 6123 READ [ARX], SKIP DPO ;ANY MORE DEST?
U 1006, 3215,3333,0003,7174,4007,0700,0400,0000,0245 ; 6124 =00
; 6125 CMS6: CALL [CMPDST] ;MORE DEST BYTES
; 6126 [AR]_WORK[FILL], ;OUT OF DEST BYTES
; 6127 J/CMS7 ;GO DO COMPARE
U 1020, 2104,4443,0000,4174,4007,0700,0010,0000,0000 ; 6128 AC[DLEN]_[ARX] ;GOT A BYTE, UPDATE LENGTH
; 6129 =
; 6130 CMS7: [AR]_[AR].AND.[MASK], ;MAKE MAGNITUDES
; 6131 WORK[CMS] ;WARM UP RAM
; 6132 [BR]_[MASK].AND.WORK[CMS], 2T ;GET SRC MAGNITUDE
; 6133 [AR]_[BR]-[AR] REV ;UNSIGNED COMPARE
U 3220, 3221,4111,1203,7174,4007,0700,0000,0000,0245 ; 6134 CMS2: [ARX]_[ARX]-1 ;UPDATE LENGTHS
U 3221, 3222,4551,1205,7274,4007,0700,0000,0000,0245 ; 6135 [BRX]_[BRX]-1 ;
U 3222, 3223,2111,0503,4174,4007,0700,4000,0000,0000 ; 6136 READ [AR], SKIP AD.EQ.O, J/CMS3 ;SEE IF EQUAL
U 3223, 3224,1111,0704,4174,4007,0700,4000,0000,0000 ; 6137
U 3224, 3225,1111,0706,4174,4007,0700,4000,0000,0000 ; 6138 =0
U 3225, 2100,3333,0003,4174,4007,0621,0000,0000,0000 ; 6139 CMS5: Q_WORK[FILL], J/CMS8 ;MORE DST--GET SRC FILL
; 6140 [AR]_O, J/CMS3 ;STRINGS ARE EQUAL
U 2102, 3226,3772,0000,7274,4007,0701,0000,0000,0244 ; 6141 CMS8: STATE_[EDIT-DST] ;JUST DST POINTER ON PAGE FAIL
U 2103, 2100,4221,0003,4174,4007,0700,0000,0000,0000 ; 6142 WORK[CMS]_Q, J/CMS6 ;MORE DST--SAVE SRC FILL
U 3226, 3227,3771,0013,4370,4007,0700,0000,0000,0012 ; 6143
U 3227, 1020,3223,0000,7174,4007,0700,0400,0000,0245 ; 6144 =0
; 6145 CMPDST: [AR]_AC[DSTP], ;GET DEST POINTER
; 6146 CALL [IDST] ;UPDATE IT
; 6147 READ [AR], ;LOOK AT BYTE POINTER
; 6148 FE_FE.AND.S#, S#/0770, ;MASK OUT BIT 6
U 2104, 3354,3771,0003,1276,6007,0701,0010,0000,1444 ; 6149 BYTE DISP, J/LDB1 ;GO LOAD BYTE
; 6150
```

Produced on Advanced Information Services Electronic Laser Printer, PNO11E66, DTN: 223-7881

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 168
EXTEND -- DECIMAL TO BINARY CONVERSION

```
; 6151 .TOC "EXTEND -- DECIMAL TO BINARY CONVERSION"
; 6152
; 6153 3742:
; 6154 DBIN: [AR]_[777777] XWD 0 ;IF WE ARE IN OFFSET MODE
; 6155 WORK[MSK]_[AR] ; ONLY ALLOW 18 BITS
; 6156 ; RANGE CHECKED (0-10) LATER
; 6157 [AR]_AC ;GET SRC LENGTH
; 6158 [BRX]_[AR].AND.# CLR RH, ;SPLIT OUT FLAGS
; 6159 #/777000 ;...
; 6160 =0* [ARX]_AC[BIN1], ;GET LOW WORD
; 6161 CALL [CLARXO] ;CLEAR BIT 0 OF ARX
; 6162 AC[BIN1]_[ARX] ;STORE BACK
; 6163 =0 READ [BRX], SKIP DPO. ;IS S ALREADY SET?
; 6164 CALL [CLRBIN] ;GO CLEAR BIN IF NOT
; 6165 [AR]_[AR].AND.#, ;CLEAR FLAGS FROM LENGTH
; 6166 #/000777, HOLD RIGHT, ;...
; 6167 B DISP ;SEE IF OFFSET OR TRANSLATE
; 6168 =1110
; 6169 DBIN1: STATE_[CVTDB], J/DBIN2 ;TRANSLATE--LEAVE S ALONE
; 6170 [BRX]_[BRX].OR.#, ;OFFSET--FORCE S TO 1
; 6171 #/400000, HOLD RIGHT,
; 6172 J/DBIN1
; 6173 DBIN2: WORK[SLEN]_.NOT.[AR] ;STORE -SLEN-1
; 6174
; 6175 ;HERE IS THE MAIN LOOP
; 6176 =0*0
; 6177 DBINLP: [AR]_WORK[SLEN]+1, CALL [SRCMOD] ;(0) GET MODIFIED SRC BYTE
; 6178 TL [BRX], #/100000, ;(1) DONE, IS M SET?
; 6179 J/DBXIT
; 6180 [AR]_.NOT.WORK[SLEN], ;(4) ABORT
; 6181 J/DBABT ;...
; 6182 [AR]-#, #/10., ;(5) NORMAL--SEE IF 0-9
; 6183 4T, SKIP DP18 ;...
; 6184 =0 [AR]_.NOT.WORK[SLEN], ;DIGIT TOO BIG
; 6185 J/DBABT ;GO ABORT CVT
; 6186
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 169
EXTEND -- DECIMAL TO BINARY CONVERSION

```
U 2111, 1030,3771,0005,1276,6007,0622,0000,0000,1443
; 6187 ;HERE TO ADD IN A DIGIT
; 6188 [BR]_AC[BINO], 4T, ;GET HIGH BINARY
; 6189 SKIP AD.EQ.O ;SEE IF SMALL
; 6190 =00
; 6191 DBSLO: [ARX]_AC[BIN1], ;TOO BIG
; 6192 CALL [DBSLOW] ;GO USE DOUBLE PRECISION PATHS
; 6193 [BR]_AC[BIN1], ;GET LOW WORD
; 6194 J/DBFAST ;MIGHT FIT IN 1 WORD
; 6195 J/DBINLP ;RETURN FROM DBSLOW
; 6196 ;GO DO NEXT DIGIT
; 6197 =
; 6198 DBFAST: TL [BR], #/760000 ;WILL RESULT FIT IN 36 BITS?
; 6199 =0 J/DBSLO ;MAY NOT FIT--USE DOUBLE WORD
; 6200 [BR]_AC[BIN1]*2 ;COMPUTE AC*2
; 6201 [BR]_[BR]*2, AC[BIN1] ;COMPUTE AC*4
; 6202 =0 [BR]_[BR]+AC[BIN1], 2T, ;COMPUTE AC*5
; 6203 CALL [SBRL] ;COMPUTE AC*10
; 6204 AC[BIN1]_[AR]+[BR], 3T, ;NEW BINARY RESULT
; 6205 J/DBINLP ;DO NEXT DIGIT
; 6206
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 170
EXTEND -- DECIMAL TO BINARY CONVERSION

```

; 6207 ;HERE IF NUMBER DOES NOT FIT IN ONE WORD
; 6208
; 6209 =000
; 6210 DBSL0W: [BR]_AC[BINO], ;FETCH HIGH WORD
; 6211 CALL [MULBY4] ;MULTIPLY BY 4
; 6212 [ARX]_[ARX]+AC[BIN1], ;COMPUTE VALUE * 5
; 6213 SKIP CRY1, 4T, ;SEE IF OVERFLOW
; 6214 CALL [ADDCRY] ;GO ADD CARRY
; 6215 =101 [BR]_[BR]+AC[BINO] ;ADD IN HIGH WORD
; 6216 =
; 6217 =000 CALL [DBLDBL] ;MAKE * 10
; 6218 [ARX]_[ARX]+[AR], 3T, ;ADD IN NEW DIGIT
; 6219 SKIP CRY1, ;SEE IF OVERFLOW
; 6220 CALL [ADDCRY] ;ADD IN THE CARRY
; 6221 =101 AC[BIN1]_[ARX] ;PUT BACK ANSWER
; 6222 =
; 6223 AC[BINO]_[BR], ;
; 6224 RETURN [2] ;GO DO NEXT BYTE
; 6225
; 6226 ;HERE TO DOUBLE BR!ARX
; 6227 =000
; 6228 MULBY4: CALL [DBLDBL] ;DOUBLE TWICE
; 6229 DBLDBL: [BR]_[BR]+[BR] ;DOUBLE HIGH WORD FIRST
; 6230 ;(SO WE DON'T DOUBLE CARRY)
; 6231 [ARX]_[ARX]+[ARX], ;DOUBLE LOW WORD
; 6232 SKIP CRY1, 3T, ;SEE IF CARRY
; 6233 CALL [ADDCRY] ;ADD IN CARRY
; 6234 =110 RETURN [1] ;ALL DONE
; 6235 =
; 6236
; 6237 ;HERE TO ADD THE CARRY
; 6238 =0
; 6239 ADDCRY: RETURN [4] ;NO CARRY
; 6240 CLEAR [ARX]0 ;KEEP LOW WORD POSITIVE
; 6241 [BR]_[BR]+1, ;ADD CARRY
; 6242 RETURN [4] ;ALL DONE
; 6243

U 0560, 0620,3771,0005,1276,6007,0701,0010,0000,1443
U 0561, 2116,0551,0404,1274,4007,0562,0010,0000,1444
U 0565, 0600,0551,0505,1274,4007,0701,0000,0000,1443
U 0600, 0621,4443,0000,4174,4007,0700,0010,0000,0000
U 0601, 2116,0111,0304,4174,4007,0561,0010,0000,0000
U 0605, 3236,3440,0404,1174,4007,0700,0400,0000,1444
U 3236, 0002,3440,0505,1174,4004,1700,0400,0000,1443
U 0620, 0621,4443,0000,4174,4007,0700,0010,0000,0000
U 0621, 0622,0111,0505,4174,4007,0700,0000,0000,0000
U 0622, 2116,0111,0404,4174,4007,0561,0010,0000,0000
U 0626, 0001,4443,0000,4174,4004,1700,0000,0000,0000
U 2116, 0004,4443,0000,4174,4004,1700,0000,0000,0000
U 2117, 3237,4551,0404,4374,0007,0700,0000,0037,7777
U 3237, 0004,0111,0705,4174,4004,1700,0000,0000,0000
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 171
EXTEND -- DECIMAL TO BINARY CONVERSION

U 3240, 3241, 3111, 0306, 4174, 4007, 0700, 0000, 0000, 0000
U 3241, 2121, 1111, 0701, 4170, 4007, 0700, 4000, 0000, 0000

U 2120, 3244, 3771, 0004, 1276, 6007, 0701, 0000, 0000, 1444
U 2121, 3242, 3771, 0003, 1276, 6007, 0701, 0000, 0000, 1444

U 3242, 2122, 3771, 0005, 1276, 6007, 0522, 0000, 0000, 1443
U 2122, 3243, 4551, 0303, 4374, 0007, 0700, 0000, 0037, 7777
U 2123, 3243, 3551, 0303, 4374, 0007, 0700, 0000, 0040, 0000
U 3243, 2124, 3440, 0303, 1174, 4007, 0700, 0400, 0000, 1444

U 2124, 2132, 3770, 0606, 0174, 4007, 0520, 0410, 0000, 0000
U 2125, 0372, 4221, 0013, 4170, 4007, 0700, 0000, 0000, 0000

U 3244, 3245, 4551, 0404, 4374, 0007, 0700, 0000, 0037, 7777

U 3245, 2126, 2441, 0404, 1174, 4007, 0621, 4000, 0000, 1443
U 2126, 2131, 7771, 0003, 1274, 4007, 0700, 0000, 0000, 1443

U 2127, 2130, 1771, 0003, 1274, 4007, 0621, 4000, 0000, 1443
U 2130, 2131, 4571, 1204, 4374, 4007, 0700, 0000, 0040, 0000

U 2131, 3246, 3440, 0303, 1174, 4007, 0700, 0400, 0000, 1443
U 3246, 2121, 3440, 0404, 1174, 4007, 0700, 0400, 0000, 1444

U 2132, 3247, 4223, 0000, 1174, 4007, 0700, 0400, 0000, 1443
U 2133, 0001, 4443, 0000, 4174, 4004, 1700, 0000, 0000, 0000
U 3247, 0001, 4223, 0000, 1174, 4004, 1700, 0400, 0000, 1444

; 6244 ;HERE TO ABORT CONVERSION
; 6245 DBABT: [BRX][BRX].OR.[AR] ;PUT BACK UNUSED LENGTH
; 6246 [PC][PC]-1, HOLD LEFT, ;DO NOT SKIP
; 6247 J/DBDONE ;GO FIX UP SIGN COPY
; 6248
; 6249 ;HERE AT END
; 6250 =0
; 6251 DBXIT: [ARX]_AC[BIN1], ;GET LOW WORD
; 6252 J/DBNEG ;GO NEGATE
; 6253 DBDONE: [AR]_AC[BIN1] ;FETCH LOW WORD
; 6254 [BR]_AC[BINO], 4T, ;GET HIGH WORD
; 6255 SKIP DPO ;WHAT SIGN
; 6256 =0 CLEAR [AR]0, J/DBDN1 ;POSITIVE
; 6257 [AR][AR].OR.#, #/400000, HOLD RIGHT
; 6258 DBDN1: AC[BIN1]_[AR] ;STORE AC BACK
; 6259 =0 AC [BRX] TEST, ;RETURN FLAGS
; 6260 SKIP DPO, CALL [CLRBIN] ;CLEAR BIN IS S=0
; 6261 ENDSKP: END STATE, J/SKIP ;NO--ALL DONE
; 6262
; 6263 DBNEG: CLEAR ARX0 ;CLEAR EXTRA SIGN BIT
; 6264 [ARX]_[ARX], 3T, ;NEGATE AND SEE IF
; 6265 SKIP AD.EQ.O, AC[BINO] ; ANY CARRY
; 6266 =0 [AR]_.NOT.AC[BINO], 2T, J/STAC34 ;NO CARRY
; 6267 [AR]_[AC[BINO], 3T, ;CARRY
; 6268 SKIP AD.EQ.O ;SEE IF ALL ZERO
; 6269 =0 [ARX]_[400000] XWD 0 ;MAKE COPY OF SIGN
; 6270 ; UNLESS HIGH WORD IS ZERO
; 6271 STAC34: AC[BINO]_[AR] ;PUT BACK ANSWER
; 6272 AC[BIN1]_[ARX], J/DBDONE ; ..
; 6273
; 6274 ;HELPER SUBROUTINE TO CLEAR AC[BINO] AND AC[BIN1] IF S=0
; 6275 ;CALL WITH:
; 6276 ; READ [BRX], SKIP DPO, CALL [CLRBIN]
; 6277 ;RETURNS 1 ALWAYS
; 6278 =0
; 6279 CLRBIN: AC[BINO]_0, J/CLRB1
; 6280 RETURN [1]
; 6281 CLRB1: AC[BIN1]_0, RETURN [1]
; 6282

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 172
EXTEND -- BINARY TO DECIMAL CONVERSION

```

; 6283 .TOC "EXTEND -- BINARY TO DECIMAL CONVERSION"
; 6284
; 6285 3743:
; 6286 BDEC: [BRX]_AC[DLEN], ;GET LENGTH AND FLAGS
; 6287 SKIP FPD ;CONTINUE FROM INTERRUPT?
; 6288 =0 [BRX]_[BRX].AND.#, ;JUST KEEP THE FLAGS
; 6289 #/777000, ;
; 6290 J/BDECO ;COMPUTE NEW FLAGS
; 6291 DOCVT: [AR]_AC, J/DOCVT1 ;ALL SET PRIOR TO TRAP
; 6292 BDECO: [ARX]_AC[1] ;GET LOW BINARY
; 6293 [AR]_AC, SC 20. ;GET HIGH WORD, SET STEP COUNT
; 6294 =0* WORK[BDL]_[ARX], ;SAVE IN CASE OF ABORT
; 6295 CALL [CLARXO] ;MAKE SURE BIT 0 IS OFF
; 6296 WORK[BDH]_[AR], ;SAVE HIGH WORD AND
; 6297 SKIP DPO ; TEST SIGN
; 6298 =0
; 6299 BDEC1: [BRX]_O, HOLD LEFT, ;POSITIVE, CLEAR RH OF BRX
; 6300 J/BDEC3 ;COMPUTE # OF DIGITS REQUIRED
; 6301 [BRX]_[BRX].OR.#, ;NEGATIVE, SET M
; 6302 #/100000, HOLD RIGHT ;
; 6303 =0*
; 6304 BDEC2: CLEAR ARXO, CALL [DBLNG1] ;NEGATE ARIARX
; 6305 AC [AR] TEST, ;PUT BACK ANSWER
; 6306 SKIP DPO ;IF STILL MINUS WE HAVE
; 6307 ; 1BO, AND NO OTHER BITS
; 6308 =0 AC[1]_[ARX], J/BDEC1 ;POSITIVE NOW
; 6309 [ARX]_[ARX]+1 ;JUST 1BO--ADD 1
; 6310 [BRX]_[BRX].OR.#, ;AND REMEMBER THAT WE DID
; 6311 #/040000, HOLD RIGHT, ; IN LEFT HALF OF AC+3
; 6312 J/BDEC2 ; NEGATE IT AGAIN
; 6313 =0
; 6314 BDEC3: [AR]_AC, J/BDEC4 ;GET HIGH AC
; 6315 [BRX]_[BRX].OR.#, ;NO LARGER POWER OF 10 FITS
; 6316 #/200000, ;SET N FLAG (CLEARLY NOT 0)
; 6317 HOLD RIGHT, J/BDEC5 ;SETUP TO FILL, ETC.
; 6318 =001
; 6319 BDEC4: [ARX]_AC[1], ;GET HIGH WORD
; 6320 CALL [BDSUB] ;SEE IF 10**C(BRX) FITS
; 6321 =011 [BRX]_[BRX]+1, ;NUMBER FITS--TRY A LARGER ONE
; 6322 STEP SC, J/BDEC3 ;UNLESS WE ARE OUT OF NUMBERS
; 6323 =111 TR [BRX], #/777777 ;ANY DIGITS REQUIRED?
; 6324 =
; 6325 =0 [BRX]_[BRX].OR.#, ;SOME DIGITS NEEDED,
; 6326 #/200000, HOLD RIGHT, ; SET N FLAG
; 6327 J/BDEC5 ;CONTINUE BELOW
; 6328 [BRX]_[BRX]+1 ;ZERO--FORCE AT LEAST 1 DIGIT
; 6329
```

U 3743, 2134,3771,0006,1276,6007,0351,0000,0000,1443

U 2134, 3250,4551,0606,4374,4007,0700,0000,0077,7000
U 2135, 3266,3771,0003,0276,6007,0700,0000,0000,0000
U 3250, 3251,3771,0004,1276,6007,0701,0000,0000,1441
U 3251, 2140,3771,0003,0276,6007,0700,2000,0071,0024

U 2140, 2202,3333,0004,7174,4007,0700,0410,0000,0250

U 2142, 2136,3333,0003,7174,4007,0520,0400,0000,0247

U 2136, 2146,4221,0006,4170,4007,0700,0000,0000,0000

U 2137, 2141,3551,0606,4374,0007,0700,0000,0010,0000

U 2141, 2753,4551,0404,4374,0007,0700,0010,0037,7777

U 2143, 2144,3770,0303,0174,4007,0520,0400,0000,0000

U 2144, 2136,3440,0404,1174,4007,0700,0400,0000,1441

U 2145, 3252,0111,0704,4174,4007,0700,0000,0000,0000

U 3252, 2141,3551,0606,4374,0007,0700,0000,0004,0000

U 2146, 0441,3771,0003,0276,6007,0700,0000,0000,0000

U 2147, 2152,3551,0606,4374,0007,0700,0000,0020,0000

U 0441, 2172,3771,0004,1276,6007,0701,0010,0000,1441

U 0443, 2146,0111,0706,4174,4007,0630,2000,0060,0000

U 0447, 2150,4553,0600,4374,4007,0331,0000,0077,7777

U 2150, 2152,3551,0606,4374,0007,0700,0000,0020,0000

U 2151, 2152,0111,0706,4174,4007,0700,0000,0000,0000

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 173
EXTEND -- BINARY TO DECIMAL CONVERSION

```

; 6330 =0
; 6331 BDEC5: [AR]_AC[DLEN], ;GET LENGTH
; 6332 CALL [CLRFLG] ;REMOVE FLAGS FROM AR
; 6333 [BR]_O
; 6334 [BR]_[BRX], HOLD LEFT ;GET # OF DIGITS NEEDED
; 6335 [BR]_[BR]-[AR], ;NUMBER OF FILLS NEEDED
; 6336 SKIP AD.LE.O ;SEE IF ENOUGH ROOM
; 6337 =0 [ARX]_WORK[BDL], ;DOES NOT FIT IN SPACE ALLOWED
; 6338 J/BDABT ; DO NOT DO CONVERT
; 6339 READ [BRX], SKIP DPO ;IS L ALREADY SET
; 6340 =0 AC[DLEN]_[BRX], ;NO--NO FILLERS
; 6341 J/DOCVT ;GO CHURN OUT THE NUMBER
; 6342
; 6343
; 6344 ;HERE TO STORE LEADING FILLERS
; 6345 [AR]_[BRX], HOLD RIGHT ;MAKE SURE THE FLAGS GET SET
; 6346 AC[DLEN]_[AR] ; BEFORE WE PAGE FAIL
; 6347 [AR]_WORK[EO] ;ADDRESS OF FILL (-1)
; 6348 [AR]_[AR]+1, LOAD VMA, ;FETCH FILLER
; 6349 START READ
; 6350 MEM READ, [TO]_MEM ;GET FILLER INTO AR
; 6351 STATE_[EDIT-DST] ;PAGE FAILS BACKUP DST
; 6352 WORK[SLEN]_[BR]-1, 3T ;SAVE # OF FILLERS
; 6353 BDFILL: [AR]_[TO], WORK[SLEN] ;RESTORE FILL BYTE AND
; 6354 ; WARM UP RAM FILE
; 6355 [BR]_WORK[SLEN]+1, 3T, ;MORE FILLERS NEEDED?
; 6356 SKIP DPO
; 6357 =000 AC[DLEN]_[BRX], J/DOCVT ;ALL DONE FIX FLAGS AND CONVERT
; 6358 =001 WORK[SLEN]_[BR], ;SAVE UPDATED LENGTH
; 6359 CALL [PUTDST] ; AND STORE FILLER
; 6360 =111 [BR]_AC[DLEN]-1 ;COUNT DOWN STRING LENGTH
; 6361 =
; 6362 AC[DLEN]_[BR], J/BDFILL ;KEEP FILLING
; 6363
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 174
EXTEND -- BINARY TO DECIMAL CONVERSION

```
; 6364 ;HERE TO STORE THE ANSWER
; 6365
; 6366 DOCVT1: [ARX]_AC[1], ;GET LOW WORD
; 6367 J/DOCVT2 ;ENTER LOOP FROM BOTTOM
; 6368 =010
; 6369 BDECLP: [BR]_[BR]+1, ;COUNT DIGITS
; 6370 CALL [BDSUB] ;KEEP SUBTRACTING 10**C(BRX)
; 6371 =110 WORK[BDH]_[AR] ;SAVE BINARY
; 6372 =
; 6373 [AR]_[BR]+WORK[E1], ;OFFSET DIGIT
; 6374 B DISP ;SEE WHICH MODE
; 6375 =1110 READ [AR], LOAD VMA, ;TRANSLATE, START READING TABLE
; 6376 START READ, J/BDTBL ; GO GET ENTRY FROM TABLE
; 6377 BDSET: WORK[BDL]_[ARX] ;SAVE LOW BINARY
; 6378 =00* STATE [EDIT-DST], CALL [PUTDST]
; 6379 =11* [BR]_AC[DLEN]-1 ;UPDATE STRING LENGTH
; 6380 [AR]_WORK[BDH]
; 6381 [ARX]_WORK[BDL]
; 6382 TL [BR], #/040000 ;ARE WE CONVERTING 1BO?
; 6383 =0 [ARX]_[ARX]+1, J/BDCFLG ;YES--FIX THE NUMBER AND CLEAR FLAG
; 6384 DOCVT3: AC [AR]
; 6385 AC[1] [ARX]
; 6386 AC[DLEN] [BR] ;STORE BACK NEW STRING LENGTH
; 6387 DOCVT2: [BRX]_[BRX]-1, 3T, SKIP DP18
; 6388 =0 [BR]_-1, SET FPD, 3T, J/BDECLP
; 6389 END STATE, CLR FPD, J/SKIP
; 6390
; 6391 ;HERE TO TRANSLATE 1 DIGIT
; 6392 =0
; 6393 BDTBL: END STATE, ;DON'T CHANGE BYTE POINTER IF
; 6394 ; THIS PAGE FAILS
; 6395 CALL [LOADAR] ;GO PUT WORD IN AR
; 6396 TR [BRX], #/777777 ;LAST DIGIT
; 6397 =0 [AR]_0, HOLD RIGHT, J/BDSET
; 6398 TL [BRX], #/100000 ;AND NEGATIVE
; 6399 =0 [AR]_[AR] SWAP ;LAST AND MINUS, USE LH
; 6400 [AR]_0, HOLD RIGHT, J/BDSET
; 6401
; 6402 BDABT: [AR]_WORK[BDH], J/DAC
; 6403
; 6404 BDCFLG: [BR]_[BR].AND.NOT.#, ;CLEAR FLAG THAT TELLS US
; 6405 #/040000, HOLD RIGHT, ; TO SUBTRACT 1 AND
; 6406 J/DOCVT3 ; CONTINUE CONVERTING
; 6407

U 3266, 3275,3771,0004,1276,6007,0701,0000,0000,1441
U 0562, 2172,0111,0705,4174,4007,0700,0010,0000,0000
U 0566, 3267,3333,0003,7174,4007,0700,0400,0000,0247

U 3267, 0636,0551,0503,7274,4003,7701,0000,0000,0241

U 0636, 2164,3333,0003,4174,4007,0700,0200,0004,0012
U 0637, 0510,3333,0004,7174,4007,0700,0400,0000,0250
U 0510, 3353,3771,0013,4370,4007,0700,0010,0000,0012
U 0516, 3270,2551,0705,1274,4007,0701,4000,0000,1443
U 3270, 3271,3771,0003,7274,4007,0701,0000,0000,0247
U 3271, 3272,3771,0004,7274,4007,0701,0000,0000,0250
U 3272, 2160,4553,0500,4374,4007,0321,0000,0004,0000
U 2160, 3277,0111,0704,4174,4007,0700,0000,0000,0000
U 2161, 3273,3440,0303,0174,4007,0700,0400,0000,0000
U 3273, 3274,3440,0404,1174,4007,0700,0400,0000,1441
U 3274, 3275,3440,0505,1174,4007,0700,0400,0000,1443
U 3275, 2162,1111,0706,4174,4007,0531,4000,0000,0000
U 2162, 0562,2441,0705,4174,4467,0701,4000,0003,0000
U 2163, 0372,4221,0013,4170,4467,0700,0000,0005,0000

U 2164, 3562,4221,0013,4170,4007,0700,0010,0000,0000
U 2165, 2166,4553,0600,4374,4007,0331,0000,0077,7777
U 2166, 0637,4221,0003,4174,0007,0700,0000,0000,0000
U 2167, 2170,4553,0600,4374,4007,0321,0000,0010,0000
U 2170, 2171,3770,0303,4344,4007,0700,0000,0000,0000
U 2171, 0637,4221,0003,4174,0007,0700,0000,0000,0000

U 3276, 1505,3771,0003,7274,4007,0701,0000,0000,0247

U 3277, 2161,5551,0505,4374,0007,0700,0000,0004,0000
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 175
EXTEND -- BINARY TO DECIMAL CONVERSION

U 2172, 2174,0551,0616,4374,4007,0701,0000,0000,0344
U 2173, 2565,4443,0000,4174,4007,0700,0000,0000,0000

U 2174, 2202,3441,1617,4174,4007,0700,0210,0000,0010

U 2176, 2200,1551,0404,6274,4007,0561,4000,0000,0000
U 2200, 2201,1111,0703,4174,4007,0700,4000,0000,0000
U 2201, 3300,0551,0616,4374,4007,0701,0000,0000,0373
U 3300, 3301,3333,0016,4174,4007,0700,0200,0000,0010

U 3301, 2202,1551,0303,6274,4007,0522,4000,0000,0000

U 2202, 0002,4551,0404,4374,0004,1700,0000,0037,7777
U 2203, 3302,0551,0303,6274,4007,0700,0000,0000,0000
U 3302, 3303,3333,0017,4174,4007,0700,0200,0000,0010
U 3303, 2204,0551,0404,6274,4007,0561,0000,0000,0000

U 2204, 0006,4551,0404,4374,0004,1700,0000,0037,7777

U 2205, 2204,0111,0703,4174,4007,0700,0000,0000,0000

```
; 6408 ;SUBROUTINE TO SUBTRACT A POWER OF 10 FROM AR!ARX
; 6409 ;CALL WITH:
; 6410 ; AR!ARX/ NUMBER TO BE CONVERTED
; 6411 ; BRX(RIGHT)/ POWER OF 10
; 6412 ;RETURNS:
; 6413 ; 2 RESULT IS STILL POSITIVE
; 6414 ; 6 RESULT WOULD HAVE BEEN NEGATIVE (RESTORE DONE)
; 6415 =0
; 6416 BDSUB: [TO]_ [BRX]+#, 3T, WORK/DECLO, ;ADDRESS OF LOW WORD
; 6417 J/BDSUB1 ;NO INTERRUPT
; 6418 J/FIXPC ;INTERRUPT
; 6419 =0*
; 6420 BDSUB1: [T1]_ [TO], LOAD VMA, ;PUT IN VMA,
; 6421 CALL [CLARXO] ;FIX UP SIGN OF LOW WORD
; 6422 [ARX]_ [ARX]-RAM, 3T, ;SUBTRACT
; 6423 SKIP CRY1 ;SEE IF OVERFLOW
; 6424 =0 [AR]_ [AR]-1 ;PROCESS CARRY
; 6425 [TO]_ [BRX]+#, 3T, WORK/DECHI ;ADDRESS OF HIGH WORD
; 6426 READ [TO], LOAD VMA ;PLACE IN VMA
; 6427 [AR]_ [AR]-RAM, 4T, ;SUBTRACT
; 6428 SKIP DPO ;SEE IF IT FIT
; 6429 =0
; 6430 CLARXO: CLEAR ARXO, ;IT FIT, KEEP LOW WORD +
; 6431 RETURN [2] ; AND RETURN
; 6432 [AR]_ [AR]+RAM ;RESTORE
; 6433 READ [T1], LOAD VMA
; 6434 [ARX]_ [ARX]+RAM, 3T, SKIP CRY1
; 6435 =0
; 6436 BDSUB2: CLEAR ARXO, ;KEEP LOW WORD +
; 6437 RETURN [6] ;RETURN OVERFLOW
; 6438 [AR]_ [AR]+1, ;ADD BACK THE CARRY
; 6439 J/BDSUB2 ;COMPLETE SUBTRACT
; 6440
```


; T1OKI.MCR[10,1141]
; EXTEND.MIC[10,1141]

11:45 11-AUG-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 176
EXTEND -- EDIT -- MAIN LOOP

```
; 6441 .TOC "EXTEND -- EDIT -- MAIN LOOP"
; 6442
; 6443 ;HERE FOR EDIT INSTRUCTION
; 6444 ;CALL WITH:
; 6445 ; AR/ EO ADDRESS OF FILL, FLOAT, AND MESSAGE TABLE
; 6446 ; BR/ E1 TRANSLATE TABLE
; 6447 ;
; 6448 3741:
; 6449 EDIT: VMA_[AR]+1, START READ, ;FIRST GET FILL BYTE
U 3741, 3361,0111,0703,4170,4007,0700,0210,0004,0012 ; 6450 CALL [GTFILL] ;GO GET IT
U 3751, 2210,3771,0006,0276,6007,0700,0000,0000,0000 ; 6451 [BRX]_AC ;GET PATTERN POINTER
; 6452 =0** TL [BRX], #/047777, ;MAKE SURE SECTION 0
; 6453 CALL [BITCHK] ; ..
U 2210, 3556,4553,0600,4374,4007,0321,0010,0004,7777 ; 6454 EDITLP: VMA_[BRX], START READ ;FETCH PATTERN WORD
U 2214, 3304,3443,0600,4174,4007,0700,0200,0004,0012 ; 6455 END STATE ;NO SPECIAL PAGE FAIL ACTION
U 3304, 3305,4221,0013,4170,4007,0700,0000,0000,0000 ; 6456 [BR]_[BRX] SWAP ;GET PBN IN BITS 20 & 21
U 3305, 2206,3770,0605,4344,4007,0700,0000,0000,0000 ; 6457 =0 [BR]_[BR]*4, ; ..
; 6458 CALL [LOADAR] ;GET PATTERN WORD
U 2206, 3562,0115,0505,4174,4007,0700,0010,0000,0000 ; 6459 READ [BR], 3T, DISP/DP LEFT
U 2207, 0654,3333,0005,4174,4003,1701,0000,0000,0000 ; 6460 =1100
; 6461 [AR]_[AR] SWAP, SC_7, J/MOVPAT ;(0) BITS 0-8
U 0654, 2212,3770,0303,4344,4007,0700,2000,0071,0007 ; 6462 [AR]_[AR] SWAP, J/MSKPAT ;(1) BITS 9-17
U 0655, 2213,3770,0303,4344,4007,0700,0000,0000,0000 ; 6463 [AR]_[AR]*.5, SC_6, J/MOVPAT ;(2) BITS 18-27
U 0656, 2212,3447,0303,4174,4007,0700,2000,0071,0006 ; 6464 [AR]_[AR].AND.#, #/777, J/EDISP ;(3) BITS 28-35
U 0657, 3306,4551,0303,4374,4007,0700,0000,0000,0777 ; 6465 =0
; 6466 MOVPAT: [AR]_[AR]*.5, STEP SC, J/MOVPAT ;SHIFT OVER
U 2212, 2212,3447,0303,4174,4007,0630,2000,0060,0000 ; 6467 MSKPAT: [AR]_[AR].AND.#, #/777
U 2213, 3306,4551,0303,4374,4007,0700,0000,0000,0777 ; 6468
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 177
EXTEND -- EDIT -- MAIN LOOP

U 3306, 2216,3447,0305,4174,4007,0700,2000,0071,0002
U 2216, 2216,3447,0505,4174,4007,0630,2000,0060,0000
U 2217, 0661,3333,0005,4174,4003,5701,0000,0000,0000

U 0661, 2220,1553,0300,4374,4007,0532,4000,0000,0005

U 0663, 2232,3333,0006,4174,4007,0520,0000,0000,0000
U 0665, 3326,4443,0000,4174,4007,0700,0000,0000,0000
U 0667, 3326,4443,0000,4174,4007,0700,0000,0000,0000
U 0671, 3326,4443,0000,4174,4007,0700,0000,0000,0000

U 0673, 2236,4553,0600,4374,4007,0321,0000,0010,0000

U 0675, 2236,4553,0600,4374,4007,0321,0000,0020,0000
U 0677, 2236,4443,0000,4174,4007,0700,0000,0000,0000

U 2220, 3326,4443,0000,4174,4007,0700,0000,0000,0000
U 2221, 0710,3333,0003,4174,4003,5701,0000,0000,0000
U 0710, 3307,0111,0701,4174,4007,0700,0000,0000,0000

U 0711, 2175,3771,0013,4370,4007,0700,0000,0000,0011
U 0712, 0246,3333,0006,4174,4007,0520,0000,0000,0000

U 0713, 3326,4551,0606,4374,0007,0700,0000,0007,7777
U 0714, 0715,3771,0005,1276,6007,0701,0000,0000,1443
U 0715, 2224,3443,0500,4174,4007,0700,0200,0004,0012

; 6469 ;HERE WITH PATTERN BYTE RIGHT ADJUSTED IN AR
; 6470 EDISP: [BR]_[AR]*.5, SC_2 ;SHIFT OVER
; 6471 =0
; 6472 EDISP1: [BR]_[BR]*.5, STEP SC, J/EDISP1
; 6473 READ [BR], 3T, DISP/DP ;LOOK AT HIGH 3 BITS
; 6474 =0001 ;(0) OPERATE GROUP
; 6475 [AR]-#, #/5, 4T, ; SEE IF 0-4
; 6476 SKIP DP18, J/EDOPR
; 6477 ;(1) MESSAGE BYTE
; 6478 READ [BRX], SKIP DPO,
; 6479 J/EDMSG ;(2) UNDEFINED
; 6480 ;(3) UNDEFINED
; 6481 J/EDNOP ;(4) UNDEFINED
; 6482 ;(5) SKIP IF M SET
; 6483 J/EDNOP ;(6) SKIP IF N SET
; 6484 ;(7) SKIP ALWAYS
; 6485 J/EDNOP
; 6486 ;(5) SKIP IF M SET
; 6487 TL [BRX], #/100000,
; 6488 J/EDSKP ;(6) SKIP IF N SET
; 6489 ;(7) SKIP ALWAYS
; 6490 TL [BRX], #/200000,
; 6491 J/EDSKP
; 6492 ;(7) SKIP ALWAYS
; 6493 J/EDSKP
; 6494
; 6495 .TOC "EXTEND -- EDIT -- DECODE OPERATE GROUP"
; 6496
; 6497 ;HERE FOR OPERATE GROUP. SKIP IF IN RANGE
; 6498 =0
; 6499 EDOPR: J/EDNOP ;OUT OF RANGE
; 6500 READ [AR], 3T, DISP/DP ;DISPATCH ON TYPE
; 6501 =1000 [PC]_[PC]+1, J/EDSTOP ;(0) STOP EDIT
; 6502 STATE_[EDIT-SRC], ;(1) SELECT SOURCE BYTE
; 6503 J/EDSEL
; 6504 READ [BRX], SKIP DPO, ;(2) START SIGNIFICANCE
; 6505 J/EDSSIG
; 6506 [BRX]_[BRX].AND.#, ;(3) FIELD SEPERATOR
; 6507 #/77777, HOLD RIGHT,
; 6508 J/EDNOP
; 6509 [BR]_AC[MARK] ;(4) EXCHANGE MARK AND DEST
; 6510 VMA [BR], START READ,
; 6511 J/EDEXMD
; 6512 =
; 6513

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 178
EXTEND -- EDIT -- STOP EDIT

```
; 6514 .TOC "EXTEND -- EDIT -- STOP EDIT"
; 6515
; 6516 ;HERE TO END AN EDIT OPERATION. PC IS SET TO SKIP IF NORMAL END
; 6517 ; OR NON-SKIP IF ABORT
; 6518 EDSTOP: [BR]_NOT.[BRX], ;AD WILL NOT DO D.AND.NOT.A
; 6519 FE S#, S#/10 ;PRESET FE
U 3307, 3310,7441,0605,4174,4007,0700,1000,0071,0010
U 3310, 3311,3441,0603,4174,4007,0701,1000,0043,0000
; 6520 [AR]_[BRX], 3T, FE_FE+P ;MOVE POINTER, UPBATE PBN
; 6521 [BR].AND.#, 3T, ;WAS OLD NUMBER 3?
; 6522 #/O30000, SKIP ADL.EQ.O ; ..
; 6523 =0
; 6524 EDSTP1: [AR]_P, J/STAC ;NO--ALL DONE
; 6525 [AR]_[AR]+1, ;YES--BUMP WORD #
; 6526 FE_FE.AND.S#, S#/O700, ;KEEP ONLY FLAG BITS
; 6527 J/EDSTP1 ;GO STOP EDIT
; 6528
; 6529 .TOC "EXTEND -- EDIT -- START SIGNIFICANCE"
; 6530
; 6531 ;HERE WITH DST POINTER IN AR
; 6532 =110
; 6533 EDSSIG: CALL [EDFLT] ;STORE FLT CHAR
; 6534 J/EDNOP ;DO NEXT PATTERN BYTE
; 6535
; 6536 .TOC "EXTEND -- EDIT -- EXCHANGE MARK AND DESTINATION"
; 6537
; 6538 ;HERE WITH ADDRESS OF MARK POINTER IN BR
; 6539 =0
; 6540 EDEXMD: Q_AC[DSTP], ;GET DEST POINTER
; 6541 CALL [LOADAR] ;GO PUT MARK IN AR
; 6542 START WRITE ;START WRITE. SEPERATE STEP TO AVOID
; 6543 ; PROBLEM ON DPMS
; 6544 MEM WRITE, MEM_Q ;PUT OLD DEST IN MARK
U 2224, 3562,3772,0000,1275,5007,0701,0010,0000,1444
U 2225, 3312,4443,0000,4174,4007,0700,0200,0003,0002
; 6545 AC[DSTP]_[AR],_J/EDNOP ;PUT BACK DEST POINTER
; 6546
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 179
EXTEND -- EDIT -- PROCESS SOURCE BYTE

```

; 6547 .TOC "EXTEND -- EDIT -- PROCESS SOURCE BYTE"
; 6548
; 6549 =0*
; 6550 EDSEL: [AR]_AC[SRCP], ;PICK UP SRC POINTER
; 6551 CALL [GETSRC] ;GET SOURCE BYTE
; 6552 [AR]_[AR]*.5, WORK[E1] ;PREPARE TO TRANSLATE
; 6553 =000 [AR]_[AR]+WORK[E1], ;GO TRANSLATE BY HALFWORDS
; 6554 2T, CALL [TRNAR] ; ..
; 6555 =010
; 6556 EDFILL: READ [AR], ;(2) NO SIGNIFICANCE, GO FILL
; 6557 SKIP AD.EQ.O, ; SEE IF ANY FILLER
; 6558 J/EDFIL1 ; GO TO IT
; 6559 STATE [EDIT-SRC], ;(3) SIG START, DO FLOAT CHAR
; 6560 J/EDSFLT
; 6561 =100 J/EDSTOP ;(4) ABORT
; 6562 =101
; 6563 EDSPUT: STATE [EDIT-S+D], ;(5) NORMAL, STORE AT DST
; 6564 CALL [PUTDST] ; ..
; 6565 =111
; 6566 J/EDNOP ;(7) BYTE STORED
; 6567 =
; 6568
; 6569 ;HERE TO COMPLETE STORING FILL
; 6570 =0
; 6571 EDFIL1: J/EDSPUT ;STORE FILLER
; 6572 J/EDNOP ;NO FILLER TO STORE
; 6573
; 6574 ;HERE TO DO FLOAT BYTE
; 6575 =110
; 6576 EDSFLT: WORK[FSIG]_[ARX], ;SAVE SIG CHAR
; 6577 CALL [EDFLT] ;STORE FLOAT CHAR
; 6578 [AR]_WORK[FSIG] ;RESTORE CHAR
; 6579 [AR]_[AR].AND.# CLR LH, ;JUST KEEP THE BYTE IN CASE
; 6580 #/7777, ; DEST BYTE .GT. 15 BITS
; 6581 J/EDSPUT ;GO STORE CHAR WHICH STARTED THIS AL
; 6582
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 180
EXTEND -- EDIT -- PROCESS SOURCE BYTE

```
U 3315, 3316,3771,0005,1276,6007,0701,0000,0000,1443
U 3316, 3317,3443,0500,4174,4007,0700,0200,0003,0012
U 3317, 3320,3771,0005,1276,6007,0701,0000,0000,1444
U 3320, 2230,3333,0005,4175,5007,0701,0200,0000,0002

U 2230, 3322,4751,1203,4374,4007,0700,0010,0000,0002
U 2231, 0740,3771,0003,4365,5007,0621,0200,0000,0002

U 0740, 3353,3551,1313,4370,4007,0700,0010,0000,0012
U 0741, 3321,3551,0606,4374,0007,0700,0000,0040,0000
U 0746, 3321,3551,0606,4374,0007,0700,0000,0040,0000

U 3321, 0007,3440,0606,0174,4004,1700,0400,0000,0000

U 2232, 0760,3771,0003,7274,4007,0622,0000,0000,0244
U 2233, 2234,4251,0303,4374,4007,0700,0000,0000,0077
U 2234, 3322,0111,0703,7174,4007,0700,0010,0000,0240
U 2235, 0760,3771,0003,4365,5007,0700,0200,0000,0002

U 0760, 3353,3771,0013,4370,4007,0700,0010,0000,0012
U 0761, 3326,4443,0000,4174,4007,0700,0000,0000,0000
U 0766, 3326,4443,0000,4174,4007,0700,0000,0000,0000

U 3322, 3323,0551,0303,7274,4007,0701,0000,0000,0240
U 3323, 0001,3443,0300,4174,4004,1700,0200,0004,0012

; 6583 ;SUBROUTINE TO PROCESS FLOAT CHAR
; 6584 ;CALL WITH:
; 6585 ; AR/ POINTER TO STORE @ MARK
; 6586 ;RETURN 7 WITH FLOAT STORED
; 6587 EDFLT: [BR]_AC[MARK] ;ADDRESS OF MARK POINTER
; 6588 VMA [BR], START WRITE ;READY TO STORE
; 6589 [BR]_AC[DSTP] ;GET DST POINTER
; 6590 MEM WRITE, MEM [BR] ;STORE POINTER
; 6591 =0 [AR]_O XWD [2], ;FETCH FLOAT CHAR
; 6592 CALL [EDBYTE] ;GET TBL BYTE
; 6593 MEM READ, [AR]_MEM, ;GET FLOAT CHAR
; 6594 SKIP AD.EQ.O ;SEE IF NULL
; 6595 =000
; 6596 [FLG]_[FLG].OR.#, ;REMEMBER TO BACKUP DST POINTER
; 6597 STATE/EDIT-DST, ; WILL ALSO BACKUP SRC IF CALLED
; 6598 HOLD LEFT, ; FROM SELECT
; 6599 CALL [PUTDST] ; STORE FLOAT
; 6600 =001 [BRX]_[BRX].OR.#, #/400000,
; 6601 HOLD RIGHT, J/EDFLT1 ;NULL
; 6602 =110 [BRX]_[BRX].OR.#, #/400000,
; 6603 HOLD RIGHT, J/EDFLT1 ;MARK STORED
; 6604 =
; 6605 EDFLT1: AC_[BRX], ;SAVE FLAGS SO WE DON'T
; 6606 ;TRY TO DO THIS AGAIN IF
; 6607 ;NEXT STORE PAGE FAILS
; 6608 RETURN [7] ;AND RETURN
; 6609
; 6610 .TOC "EXTEND -- EDIT -- MESSAGE BYTE"
; 6611
; 6612 ;HERE WITH SKIP ON S
; 6613 =0
; 6614 EDMSG: [AR]_WORK[FILL], ;GET FILL BYTE
; 6615 SKIP AD.EQ.O, 4T, ;SEE IF NULL
; 6616 J/EDMSG1 ;GO STORE
; 6617 [AR]_[AR].AND.# CLR LH, ;GET OFFSET INTO TABLE
; 6618 #/77
; 6619 =0 [AR]_[AR]+1, WORK[EO], ;PLUS 1
; 6620 CALL [EDBYTE] ;GET TBL BYTE
; 6621 MEM READ, [AR]_MEM ;FROM MEMORY
; 6622 =000
; 6623 EDMSG1: STATE [EDIT-DST], ;WHAT TO DO ON PAGE FAILS
; 6624 CALL [PUTDST] ;STORE MESSAGE BYTE
; 6625 =001 J/EDNOP ;NULL FILLER
; 6626 =110 J/EDNOP ;NEXT BYTE
; 6627 =
; 6628
; 6629 EDBYTE: [AR]_[AR]+WORK[EO] ;GET OFFSET INTO TABLE
; 6630 VMA [AR], START READ, ;START MEMORY CYCLE
; 6631 RETURN [1] ;RETURN TO CALLER
; 6632
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 181
EXTEND --- EDIT -- SKIP

```
; 6633 .TOC "EXTEND -- EDIT -- SKIP"
; 6634
; 6635 =0
; 6636 ;HERE TO SKIP ALWAYS
; 6637 EDSKP: [AR]_[AR].AND.#, #/77, ;JUST KEEP SKIP DISTANCE
; 6638 J/EDSKP1 ;CONTINUE BELOW
; 6639 ;HERE IF WE DO NOT WANT TO SKIP
; 6640 J/EDNOP
; 6641 EDSKP1: [AR]_([AR]+1)*2 ;GIVE 1 EXTRA SKIP
; 6642 READ [AR], SCAD/A*2, ;PUT THE ADJUSTMENT
; 6643 SCADA/BYTES, 3T, LOAD SC, ; THE SC
; 6644 J/EDNOP1 ;JOIN MAIN LOOP
; 6645
; 6646
; 6647 .TOC "EXTEND -- EDIT -- ADVANCE PATTERN POINTER"
; 6648
; 6649 EDNOP: SC O ;NO SKIP
; 6650 EDNOP1: READ [BRX], 3T, FE_P ;PUT PBN IN FE
; 6651 FE FE.AND.S#, S#/30 ;JUST BYTE #
; 6652 FE_FE+SC ;ADD IN ANY SKIP DISTANCE
; 6653 FE FE+S#, S#/10 ;BUMP PBN
; 6654 [AR]_FE, ;GET NUMBER OF WORDS
; 6655 LOAD SC ;PUT MSB WHERE IT CAN BE TESTED
; 6656 ; QUICKLY
; 6657 [AR]_[AR].AND.# CLR LH, ;KEEP ONLY 1 COPY
; 6658 #/170, SKIP/SC ; ..
; 6659 =0
; 6660 EDN1A: [AR]_[AR]*.5, SC_O,
; 6661 J/EDNOP2 ;READY TO SHIFT OFF BYTE WITHIN
; 6662 ; WORD
; 6663 [AR]_[AR].OR.#, #/200, ;GET THE SIGN BIT OF THE FE
; 6664 HOLD LEFT, ; INTO THE AR. ONLY HAPPENS ON
; 6665 J/EDN1A ; SKP 76 OR SKP 77
; 6666 =0
; 6667 EDNOP2: [AR]_[AR]*.5, STEP SC, J/EDNOP2
; 6668 [BRX]_[BRX]+[AR], ;UPDATE WORD ADDRESS
; 6669 HOLD LEFT
; 6670 [AR]_P ;PUT PBN BACK IN BRX
; 6671 [BRX]_[BRX].AND.#, ;JUST KEEP FLAGS
; 6672 #/700000, ; ..
; 6673 HOLD RIGHT
; 6674 [AR]_[AR].AND.#, ;JUST KEEP PBN
; 6675 #/030000
; 6676 [BRX]_[BRX].OR.[AR], ;FINAL ANSWER
; 6677 HOLD RIGHT
; 6678 AC_[BRX], J/EDITLP ;DO NEXT FUNCTION
; 6679
```

U 2236, 3324, 4551, 0303, 4374, 4007, 0700, 0000, 0000, 0077
U 2237, 3326, 4443, 0000, 4174, 4007, 0700, 0000, 0000, 0000
U 3324, 3325, 0115, 0703, 4174, 4007, 0700, 0000, 0000, 0000
U 3325, 3327, 3333, 0003, 4174, 4007, 0701, 2000, 0007, 0000
U 3326, 3327, 4443, 0000, 4174, 4007, 0700, 2000, 0071, 0000
U 3327, 3330, 3333, 0006, 4174, 4007, 0701, 1000, 0073, 0000
U 3330, 3331, 4443, 0000, 4174, 4007, 0700, 1000, 0051, 0030
U 3331, 3332, 4443, 0000, 4174, 4007, 0700, 1000, 0040, 0000
U 3332, 3333, 4443, 0000, 4174, 4007, 0700, 1000, 0041, 0010
U 3333, 3334, 3777, 0003, 4334, 4057, 0700, 2000, 0041, 0000
U 3334, 2240, 4251, 0303, 4374, 4007, 0630, 0000, 0000, 0170
U 2240, 2242, 3447, 0303, 4174, 4007, 0700, 2000, 0071, 0000
U 2241, 2240, 3551, 0303, 4370, 4007, 0700, 0000, 0000, 0200
U 2242, 2242, 3447, 0303, 4174, 4007, 0630, 2000, 0060, 0000
U 2243, 3335, 0111, 0306, 4170, 4007, 0700, 0000, 0000, 0000
U 3335, 3336, 3770, 0303, 4334, 4017, 0700, 0000, 0041, 0000
U 3336, 3337, 4551, 0606, 4374, 0007, 0700, 0000, 0070, 0000
U 3337, 3340, 4551, 0303, 4374, 4007, 0700, 0000, 0003, 0000
U 3340, 3341, 3111, 0306, 4174, 0007, 0700, 0000, 0000, 0000
U 3341, 2214, 3440, 0606, 0174, 4007, 0700, 0400, 0000, 0000

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 182
EXTEND SUBROUTINES -- FILL OUT DESTINATION

```

; 6680 .TOC "EXTEND SUBROUTINES -- FILL OUT DESTINATION"
; 6681
; 6682 ;CALL WITH
; 6683 ; AC[DLEN]/ NEGATIVE NUMBER OF BYTES LEFT IN DEST
; 6684 ; FILL/ FILL BYTE
; 6685 ; RETURN [2] WITH FILLERS STORED
; 6686
; 6687 ;NOTE: THIS ROUTINE NEED NOT TEST FOR INTERRUPTS ON EACH BYTE
; 6688 ; BECAUSE EVERY BYTE STORE DOES A MEMORY READ.
; 6689
; 6690 =01*
; 6691 MOVF1: [AR]_WORK[FILL], 2T, ;GET FILL BYTE
; 6692 CALL [PUTDST] ;PLACE IN DEST
; 6693 [AR]_AC[DLEN] ;AMOUNT LEFT
; 6694 AC[DLEN]_[AR]+1, 3T, ;STORE UPDATED LEN
; 6695 SKIP DPO ; AND SEE IF DONE
; 6696 =0 RETURN [2] ;DONE
; 6697 MOVFIL: WORK[FILL], J/MOVF1 ;DO ANOTHER BYTE
; 6698 ;ENTERING HERE SAVES 150NS
; 6699 ; PER BYTE BUT COSTS 300NS
; 6700 ; PER FIELD MOVED. I ASSUME (BUT DO
; 6701 ; NOT KNOW) THAT THIS SPEEDS
; 6702 ; THINGS UP.
; 6703

U 0332, 3353,3771,0003,7274,4007,0700,0010,0000,0244
U 0336, 3342,3771,0003,1276,6007,0701,0000,0000,1443

U 3342, 2244,0113,0703,1174,4007,0521,0400,0000,1443
U 2244, 0002,4443,0000,4174,4004,1700,0000,0000,0000
U 2245, 0332,4443,0000,7174,4007,0700,0000,0000,0244
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
 ; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 183
 EXTEND SUBROUTINES -- GET MODIFIED SOURCE BYTE

```

; 6704 .TOC"EXTEND SUBROUTINES -- GET MODIFIED SOURCE BYTE"
; 6705
; 6706 ;CALL WITH:
; 6707 ;SLEN = MINUS LENGTH OF STRING
; 6708 ;MSK = MASK FOR BYTE SIZE (1 IF BIT MUST BE ZERO)
; 6709 ;E1 = EFFECTIVE ADDRESS OF OPERATION WORD (SIGN EXTENDED IF OFFSET)
; 6710 ; [AR]_WORK[SLEN]+1, CALL [SRCMOD]
; 6711 ;RETURNS:
; 6712 ; 1 LENGTH EXHAUSTED
; 6713 ; 2 (EDIT ONLY) NO SIGNIFICANCE
; 6714 ; 3 (EDIT ONLY) SIGNIFICANCE START:
; 6715 ; 4 ABORT: OUT OF RANGE OR TRANSLATE FAILURE
; 6716 ; 5 NORMAL: BYTE IN AR
; 6717 ;
; 6718 ;DROM B SET AS FOLLOWS:
; 6719 ; 0 TRANSLATE
; 6720 ; 1 OFFSET
; 6721 ; 2 EDIT
; 6722 ; 4 CVTDBT
; 6723 =00
; 6724 SRCMOD: WORK[SLEN]_[AR], ;PUT BACK SOURCE LENGTH
; 6725 SKIP DPO, ;SEE IF DONE
; 6726 CALL [GSRC] ;GET A SOURCE BYTE
; 6727 END STATE, RETURN [1] ;DONE
; 6728 WORK[E1], B DISP ;OFFSET OR TRANSLATE?
; 6729 =
; 6730 =1110 [AR]_[AR]* 5, J/XLATE ;TRANSLATE
; 6731 FIX [AR] SIGN, WORK[E1] ;IF WE ARE PROCESSING FULL WORD
; 6732 ; BYTES, AND THEY ARE NEGATIVE,
; 6733 ; AND THE OFFSET IS POSITIVE THEN
; 6734 ; WE HAVE TO MAKE BITS -1 AND -2
; 6735 ; COPIES OF THE SIGN BIT.
; 6736 [AR]_[AR]+WORK[E1], 2T ;OFFSET
; 6737 [AR].AND.WORK[MSK], ;VALID BYTE?
; 6738 SKIP AD.EQ.0, 4T, ;SKIP IF OK
; 6739 RETURN [4] ;RETURN 4 IF BAD, 5 IF OK
; 6740

U 1040, 2256,3333,0003,7174,4007,0520,0410,0000,0242
U 1041, 0001,4221,0013,4170,4004,1700,0000,0000,0000
U 1042, 0716,4443,0000,7174,4003,7700,0000,0000,0241

U 0716, 3345,3447,0303,4174,4007,0700,0000,0000,0000
U 0717, 3343,3770,0303,7174,0007,0700,0000,0000,0241

U 3343, 3344,0551,0303,7274,4007,0700,0000,0000,0241

U 3344, 0004,4553,0300,7274,4004,1622,0000,0000,0243

```


; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 184
EXTEND SUBROUTINES -- TRANSLATE

```

; 6741 .TOC "EXTEND SUBROUTINES -- TRANSLATE"
; 6742
; 6743 ;HERE WITH BYTE IN AR 1-36. FETCH TABLE ENTRY.
U 3345, 3346,0551,0303,7274,4007,0701,0000,0000,0241 ; 6744 XLATE: [AR]_[AR]+WORK[E1] ;COMPUTE ADDRESS
; 6745 TRNAR: READ [AR], LOAD VMA, ;FETCH WORD
U 3346, 2246,3333,0003,4174,4007,0700,0200,0004,0012 ; 6746 START READ ; ..
; 6747 =0 [AR]_[AR]*2, ;GET BACK LSB
; 6748 ;BIT 36 IS NOT PRESERVED
; 6749 ; BY PAGE FAILS
U 2246, 3563,3445,0303,4174,4007,0700,0010,0000,0000 ; 6750 CALL [LOADARX] ;PUT ENTRY IN ARX
U 2247, 2250,4553,0300,4374,4007,0331,0000,0000,0001 ; 6751 TR [AR], #/1 ;WHICH HALF?
; 6752 =0
; 6753 XLATE1: [AR]_[ARX], 3T, ;RH -- COPY TO AR
; 6754 DISP/DP LEFT, ;DISPATCH ON CODE
; 6755 J/TRNFNC ;DISPATCH TABLE
; 6756 [ARX]_[ARX] SWAP, ;LH -- FLIP AROUND
U 2250, 0721,3441,0403,4174,4003,1701,0000,0000,0000 ; 6757 J/XLATE1 ;START SHIFT
; 6758
U 2251, 2250,3770,0404,4344,4007,0700,0000,0000,0000
```

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 185
EXTEND SUBROUTINES -- TRANSLATE

U 0721, 2252,3333,0006,4174,4007,0520,0000,0000,0000
U 0723, 0004,4443,0000,4174,4004,1700,0000,0000,0000
U 0725, 0721,5551,0606,4374,0007,0700,0000,0010,0000
U 0727, 0721,3551,0606,4374,0007,0700,0000,0010,0000
U 0731, 0721,3551,0606,4374,0007,0700,0000,0020,0000
U 0733, 0004,3551,0606,4374,0004,1700,0000,0020,0000
U 0735, 0731,5551,0606,4374,0007,0700,0000,0010,0000
U 0737, 0721,3551,0606,4374,0007,0700,0000,0030,0000

```
; 6759 ;HERE ON TRANSLATE OPERATION TO PERFORM FUNCTIONS REQUIRED BY
; 6760 ; THE 3 HIGH ORDER BITS OF THE TRANSLATE FUNCTION HALFWORD. WE
; 6761 ; DISPATCH ON FUNCTION AND HAVE:
; 6762 ; BRX/ FLAGS
; 6763 ; ARX/ TABLE ENTRY IN RH
; 6764 ;
; 6765 =0001
; 6766 ;(0) NOP
; 6767 TRNFNC: READ [BRX], SKIP DPO, ;S FLAG ALREADY SET?
; 6768 J/TRNFNC ;
; 6769 ;(1) ABORT
; 6770 RETURN [4]
; 6771 ;(2) CLEAR M FLAG
; 6772 [BRX]_[BRX].AND.NOT.#,
; 6773 #/100000, HOLD RIGHT,
; 6774 J/TRNFNC
; 6775 ;(3) SET M FLAG
; 6776 [BRX]_[BRX].OR.#,
; 6777 #/100000, HOLD RIGHT,
; 6778 J/TRNFNC
; 6779 ;(4) SET N FLAG
; 6780 TRNSIG: [BRX]_[BRX].OR.#,
; 6781 #/200000, HOLD RIGHT,
; 6782 J/TRNFNC
; 6783 ;(5) SET N FLAG THEN ABORT
; 6784 [BRX]_[BRX].OR.#,
; 6785 #/200000, HOLD RIGHT,
; 6786 RETURN [4]
; 6787 ;(6) CLEAR M THEN SET N
; 6788 [BRX]_[BRX].AND.NOT.#,
; 6789 #/100000, HOLD RIGHT,
; 6790 J/TRNSIG
; 6791 ;(7) SET N AND M
; 6792 [BRX]_[BRX].OR.#,
; 6793 #/300000, HOLD RIGHT,
; 6794 J/TRNFNC
; 6795
```

Produced on Advanced Information Services Electronic Laser Printer. RKO1556, DTN: 222-7881

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 186
EXTEND SUBROUTINES -- TRANSLATE

```

; 6796 ;HERE TO COMPLETE A TRANSLATE
; 6797
; 6798 =0
; 6799 TRNRET: READ [ARX], SKIP DP18, ;S-FLAG IS ZERO
; 6800 B DISP, SKIP DP18, ;SEE IF EDIT OR SIG START
; 6801 J/TRNSS ; ..
U 2252, 0754,3333,0004,4174,4003,7530,0000,0000,0000
; 6802 TRNSS1: [AR]_[ARX].AND.# CLR LH, ;S IS SET, JUST RETURN BYTE
; 6803 #/77777, RETURN [5] ; ..
; 6804
; 6805 =1100
; 6806 TRNSS: [AR]_AC[DLEN], ;NO SIG ON MOVE OR D2B
; 6807 B DISP, J/TRNSS1 ;SEE IF D2B
; 6808 [BRX]_[BRX].OR.#, ;SIG START ON MOVE OR D2B
; 6809 #/400000, HOLD RIGHT,
; 6810 J/TRNSS1 ;RETURN BYTE
; 6811 [AR]_WORK[FILL], ;EDIT--NO SIG RETURN FILL
; 6812 RETURN [2] ; ..
; 6813 [AR]_AC[DSTP], ;EDIT--START OF SIG
; 6814 RETURN [3] ; ..
; 6815
; 6816 =1011
; 6817 TRNSS1: [AR]_[AR]-1, J/TRNSS2 ;COMPENSATE FOR IGNORING SRC
; 6818 [AR]_WORK[SLEN]+1, ;DEC TO BIN HAS NO DEST LENGTH
; 6819 J/SRCMOD ;JUST UPDATE SRC LENTH
; 6820 TRNSS2: AC[DLEN]_[AR] TEST, ;PUT BACK DLEN AND
; 6821 SKIP DPO ; SEE WHICH IS NOW SHORTER
; 6822 =0 [AR]_WORK[SLEN], ;DEST IS SHORTER. DO NOT CHANGE
; 6823 J/SRCMOD ; AMOUNT LEFT
; 6824 [AR]_WORK[SLEN]+1, ;GO LOOK AT NEXT BYTE
; 6825 J/SRCMOD
; 6826
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
 ; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 187
 EXTEND SUBROUTINES -- GET UNMODIFIED SOURCE BYTE

U 2256, 0001,3771,0003,1276,6004,1701,0000,0000,1443
 U 2257, 3350,3771,0003,1276,6007,0701,0000,0000,1441
 U 3350, 0231,3770,0305,4334,4016,7701,0000,0033,6000
 U 0231, 3352,3441,0503,4174,4007,0700,0000,0000,0000
 U 0233, 3351,3770,0503,4334,4017,0700,0000,0032,6000
 U 3351, 3352,0111,0703,4170,4007,0700,0000,0000,0000
 U 3352, 2260,3440,0303,1174,4007,0700,0400,0000,1441
 U 2260, 2760,3333,0003,4174,4217,0701,1010,0073,0500
 U 2261, 0340,3333,0003,4174,4006,5701,1000,0051,0770

```

; 6827 .TOC      "EXTEND SUBROUTINES -- GET UNMODIFIED SOURCE BYTE"
; 6828
; 6829 ;CALL:
; 6830 ;          GSRC WITH SKIP ON SOURCE LENGTH
; 6831 ;          GETSRC IF LENGHT IS OK
; 6832 ;WITH:
; 6833 ;          AC1/ SOURCE BYTE POINTER
; 6834 ;RETURNS:
; 6835 ;          1 IF LENGTH RAN OUT
; 6836 ;          2 IF OK (BYTE IN AR)
; 6837 ;
; 6838 =0
; 6839 GSRC:  [AR]_AC[DLEN],          ;LENGTH RAN OUT
; 6840          RETURN [1]           ;RESTORE AR AND RETURN
; 6841 GETSRC: [AR]_AC[SRCP]         ;GET SRC PTR
; 6842          IBP DP, IBP SCAD,    ;UPDATE BYTE POINTER
; 6843          SCAD DISP, 3T        ;SEE IF OFLOW
; 6844 =01  [AR]_[BR], J/GSRC1      ;NO OFLOW
; 6845          SET P TO 36-S       ;RESET P
; 6846          [AR]_[AR]+1, HOLD LEFT ;BUMP Y
; 6847
; 6848 GSRC1: AC[SRCP]_[AR]          ;STORE UPDATED POINTER
; 6849 =0  READ [AR], LOAD BYTE EA, ;SETUP TO FIGURE OUT
; 6850          FE P, 3T, CALL [BYTEAS] ; EFFECTIVE ADDRESS
; 6851          READ [AR],           ;LOOK AT POINTER
; 6852          BYTE DISP,           ;SEE IF 7 BIT
; 6853          FE FE.AND.S#, S#/0770, ;MASK OUT P FIELD
; 6854          J/LDB1              ;GO GET THE BYTE
; 6855

```

; T10KI.MCR[10,1141]
; EXTEND.MIC[10,1141]

11:45 11-AUG-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 188
EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING

U 3353, 2262,3441,0304,4174,4007,0700,0000,0000,0000
U 2262, 3354,3771,0003,1276,6007,0701,0010,0000,1444
U 2263, 2264,0113,0404,4174,4007,0701,1000,0077,0000
U 2264, 0360,3333,0003,4174,4006,5701,0010,0000,0000
U 2266, 0006,4443,0000,4174,4004,1700,0000,0000,0000

```
; 6856 .TOC "EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING"  
; 6857  
; 6858 ;CALL WITH:  
; 6859 ; AR/ BYTE TO STORE  
; 6860 ; AC4/ DESTINATION BYTE POINTER  
; 6861 ;RETURNS:  
; 6862 ; AR & AC4/ UPDATED BYTE POINTER  
; 6863 ; ARX/ BYTE TO STORE  
; 6864 ; BR/ WORD TO MERGE WITH  
; 6865 ; 6 ALWAYS  
; 6866  
; 6867 PUTDST: [ARX]_[AR] ;SAVE BYTE  
; 6868 =0 [AR]_AC[DSTP], ;GET DEST POINTER  
; 6869 CALL [IDST] ;BUMP DEST POINTER  
; 6870 AD/A+B, A/ARX, B/ARX, ;SHIFT 7-BIT BYTE TO  
; 6871 SCAD/A, 3T, ; NATURAL PLACE, AND PUT  
; 6872 SCADA/BYTE5, LOAD FE ; INTO FE  
; 6873 =0* READ [AR], BYTE DISP, ;GO PUT BYTE IN MEMORY  
; 6874 CALL [DPB1] ; ..  
; 6875 RETURN [6] ;ALL DONE  
; 6876
```

```
; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984
```

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 189
 EXTEND SUBROUTINES -- UPDATE DEST STRING POINTERS

```
; 6877 .TOC "EXTEND SUBROUTINES -- UPDATE DEST STRING POINTERS"
; 6878
; 6879
; 6880 ;SUBROUTINE TO BUMP DST POINTERS
; 6881 ;CALL WITH:
; 6882 ; AR/ AC[DSTP]
; 6883 ; RETURN 1 WITH UPDATED POINTER STORED
; 6884 ;
; 6885 IDST: IBP DP, IBP SCAD, SCAD DISP, 3T
; 6886 =0* [AR]_[BR], LOAD DST EA, J/IDSTX
; 6887 SET P TO 36-S
; 6888 [AR]_[AR]+1, HOLD LEFT, LOAD DST EA
; 6889 IDSTX: AC[DSTP]_[AR], 3T, ;STORE PTR BACK
; 6890 FE_P, DISP/EAMODE ;SAVE P FOR CMPDST
; 6891 =100*
; 6892 DSTE A: VMA [AR]+XR, START READ, PXCT BYTE DATA, 3T, J/BYTFET
; 6893 VMA [AR], START READ, PXCT BYTE DATA, J/BYTFET
; 6894 VMA [AR]+XR, START READ, PXCT/BIS-DST-EA, 3T, J/DSTIND
; 6895 VMA [AR], START READ, PXCT/BIS-DST-EA, J/DSTIND
; 6896
; 6897 DSTIND: MEM READ, [AR]_MEM, HOLD LEFT, LOAD DST EA
; 6898 EA MODE DISP, J/DSTE A
; 6899
; 6900
; 6901 ;HERE TO TEST ILLEGAL BITS SET
; 6902 ;CALL WITH:
; 6903 ; SKIP IF ALL BITS LEGAL
; 6904 ; RETURN [4] IF OK, ELSE DO UUU
; 6905 ;
; 6906 3556: ;EXTEND OF 0 COMES HERE
; 6907 BITCHK: UUU
; 6908 3557: RETURN [4]
; 6909
; 6910 ;HERE TO PUT FILL IN [AR] AND WORK[FILL]
; 6911 GTFILL: MEM READ, ;WAIT FOR DATA
; 6912 [AR]_MEM ;PLACE IN AR
; 6913 WORK[FILL]_[AR], ;SAVE FOR LATER
; 6914 RETURN [10] ;RETURN TO CALLER
; 6915
; 6916 ;SUBROUTINE TO CLEAR FLAGS IN AR
; 6917 CLRFLG: [AR]_[AR].AND.#, ;CLEAR FLAGS IN AR
; 6918 #/000777, ; ..
; 6919 HOLD RIGHT, RETURN [1]
; 6920
```

Produced on Advanced Information Services Electronic Laser Printer, PK01556, DTN: 223-7881

; T1OKI.MCR[10,1141] 11:45 11-AUG-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 190
EXTEND -- PAGE FAIL CLEANUP

```
; 6921 .TOC "EXTEND -- PAGE FAIL CLEANUP"
; 6922
; 6923 ;BACK UP SOURCE POINTER
; 6924 =0
; 6925 BACKS: [AR]_AC[SRCP],
; 6926 CALL [BACKBP] ;BACKUP BP
; 6927 AC[SRCP]_[BR], J/CLDISP
; 6928
; 6929 CMSDST: [AR]_WORK[SV.BRX] ;GET OLD SRC LEN
; 6930 AC [AR]+1, 3T ;BACK UP
; 6931 ;BACK UP DESTINATION POINTER
; 6932 =0
; 6933 BACKD: [AR]_AC[DSTP],
; 6934 CALL [BACKBP]
; 6935 AC[DSTP]_[BR], J/CLDISP
; 6936
; 6937 ;FAILURES DURING MOVE STRING (BACKUP LENGTHS)
; 6938 STRPF: [AR]_-WORK[SLEN] ;GET AMOUNT LEFT
; 6939 STRPFO: [BR]_AC[DLEN], 4T, ;WHICH STRING IS LONGER?
; 6940 SKIP DPO
; 6941 =0
; 6942 STRPF1: AC[DLEN]_[AR], J/STPF1A ;SRC LONGER
; 6943 [ARX]_[AR] ;COPY SRC LENGTH
; 6944 =0 [ARX]_[ARX].OR.WORK[SV.BRX], ;REBUILD FLAGS
; 6945 CALL [AC ARX] ;RESET AC[SLEN]
; 6946 [AR]_[AR]_[BR] ;MAKE DEST LEN
; 6947 STRPF3: AC[DLEN]_[AR], ;PUT BACK DEST LEN
; 6948 J/CLDISP ;DO NEXT CLEANUP
; 6949
; 6950 STPF1A: [AR]_[AR]+[BR], J/STRPF2
; 6951
; 6952 PFDBIN: [AR]_-WORK[SLEN] ;RESTORE LENGTH
; 6953 STRPF2: [AR]_[AR].OR.WORK[SV.BRX]
; 6954 PFGACO: AC_[AR], J/CLDISP ;PUT BACK SRC LEN AND FLAGS
; 6955
; 6956 STRPF4: [AR]_.NOT.WORK[SLEN], J/STRPFO
; 6957
; 6958 BACKBP: IBP DP, SCAD/A+B, SCADA/BYTE1, SCADB/SIZE, ;P_P+S
; 6959 RETURN [1]
; 6960
```

U 2270, 3376,3771,0003,1276,6007,0701,0010,0000,1441
U 2271, 2566,3440,0505,1174,4007,0700,0400,0000,1441

U 3364, 3365,3771,0003,7274,4007,0701,0000,0000,0214
U 3365, 2272,0113,0703,0174,4007,0701,0400,0000,0000

U 2272, 3376,3771,0003,1276,6007,0701,0010,0000,1444
U 2273, 2566,3440,0505,1174,4007,0700,0400,0000,1444

U 3366, 3367,1771,0003,7274,4007,0701,4000,0000,0242
U 3367, 2274,3771,0005,1276,6007,0522,0000,0000,1443

U 2274, 3371,3440,0303,1174,4007,0700,0400,0000,1443
U 2275, 2276,3441,0304,4174,4007,0700,0000,0000,0000

U 2276, 3573,3551,0404,7274,4007,0701,0010,0000,0214
U 2277, 3370,1111,0503,4174,4007,0700,4000,0000,0000

U 3370, 2566,3440,0303,1174,4007,0700,0400,0000,1443

U 3371, 3373,0111,0503,4174,4007,0700,0000,0000,0000

U 3372, 3373,1771,0003,7274,4007,0701,4000,0000,0242
U 3373, 3374,3551,0303,7274,4007,0701,0000,0000,0214
U 3374, 2566,3440,0303,0174,4007,0700,0400,0000,0000

U 3375, 3367,7771,0003,7274,4007,0701,0000,0000,0242

U 3376, 0001,3770,0305,4334,4014,1700,0000,0043,6000

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 191
TRAPS

```

; 6961          .TOC      "TRAPS"
; 6962
; 6963 TRAP:  [ARX]_PC WITH FLAGS      ;SAVE THE PC WHICH CAUSED THE
; 6964          WORK[TRAPPC]_[ARX],    ; TRAP
; 6965          SKIP KERNEL              ;SEE IF UBR OR EBR
; 6966          =0  [AR]_[AR]+[UBR],    ;ADDRESS OF INSTRUCTION
; 6967          MEM READ,                ;WAIT FOR PREFETCH TO GET INTO
; 6968          ; THE CACHE. MAY PAGE FAIL BUT
; 6969          ; THAT IS OK
; 6970          START READ,              ;START FETCH
; 6971          VMA PHYSICAL,           ;ABSOLUTE ADDRESSING
; 6972          J/TRP1                   ;JOIN COMMON CODE
; 6973
; 6974          [AR]_[AR]+[EBR],        ;WE COME HERE IN EXEC MODE
; 6975          MEM READ,                ;WAIT FOR PREFETCH TO GET INTO
; 6976          ; THE CACHE. MAY PAGE FAIL BUT
; 6977          ; THAT IS OK
; 6978          START READ,              ;START FETCH
; 6979          VMA PHYSICAL,           ;ABSOLUTE ADDRESSING
; 6980          J/TRP1                   ;JOIN COMMON CODE
; 6981
; 6982 TRP1:  MEM READ, [HR]_MEM,        ;PLACE INSTRUCTION IN HR
; 6983          LOAD INST                 ;LOAD IR, XR, @
; 6984          [HR].AND.#,              ;TEST TO SEE IF THIS
; 6985          #/700000, 3T,             ; IS A UUO
; 6986          SKIP ADL.EQ.0
; 6987          =0  CHANGE FLAGS,        ;NOT A UUO
; 6988          HOLD USER/1,            ;CLEAR TRAP FLAGS
; 6989          J/XCT1                   ;DO THE INSTRUCTION
; 6990          UUO                      ;DO THE UUO
; 6991

U 3377, 3400,3741,0104,4074,4007,0700,0000,0000,0000
U 3400, 2300,3333,0004,7174,4007,0340,0400,0000,0425

U 2300, 3401,0111,1103,4364,4007,0700,0200,0024,1016

U 2301, 3401,0111,1003,4364,4007,0700,0200,0024,1016

U 3401, 3402,3771,0002,4365,5617,0700,0200,0000,0002

U 3402, 2302,4553,0200,4374,4007,0321,0000,0070,0000

U 2302, 2600,4443,0000,4174,4467,0700,0000,0001,0000
U 2303, 2621,4551,0202,4374,0007,0700,0000,0077,7740
```


; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 192
IO -- INTERNAL DEVICES

D 0700, 1200, 1700, 4100
D 0701, 1200, 1720, 4100

U 1701, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1702, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1703, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1706, 3405, 3771, 0005, 4304, 4007, 0701, 0000, 0000, 0000
U 1707, 3403, 3771, 0005, 4304, 4007, 0701, 0000, 0000, 0000

U 1710, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1711, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1712, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1713, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1716, 3406, 3441, 1405, 4174, 4007, 0700, 0000, 0000, 0000
U 1717, 3404, 3441, 1405, 4174, 4007, 0700, 0000, 0000, 0000

U 1720, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1726, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1727, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1730, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1731, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1732, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1733, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1734, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1735, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1736, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1737, 2621, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 3403, 3404, 4251, 0505, 4374, 4007, 0700, 0000, 0000, 7770
U 3404, 0372, 4113, 0305, 4174, 4007, 0330, 0000, 0000, 0000

U 3405, 3406, 4251, 0505, 4374, 4007, 0700, 0000, 0000, 7770
U 3406, 1400, 4113, 0305, 4174, 4007, 0330, 0000, 0000, 0000

```
; 6992      .TOC      "IO -- INTERNAL DEVICES"
; 6993
; 6994      .DCODE
; 6995      700:      IOT,AC DISP,      J/GRP700
; 6996      IOT,AC DISP,      J/GRP701
; 6997      .UCODE
; 6998
; 6999      1701:     UUU          ;DATAI APR,
; 7000      1702:     UUU          ;BLKO APR,
; 7001      1703:     UUU          ;DATAO APR,
; 7002      1706:     [BR]_APR, J/APRSZ ;CONSZ APR,
; 7003      1707:     [BR]_APR, J/APRSO ;CONSO APR,
; 7004      1710:
; 7005      RDERA:    UUU          ;BLKI PI,
; 7006      1711:     UUU          ;DATAI PI,
; 7007      1712:     UUU          ;BLKO PI,
; 7008      1713:     UUU          ;DATAO PI,
; 7009      1716:     [BR]_[PI], J/CONSZ ;CONSZ PI,
; 7010      1717:     [BR]_[PI], J/CONSO ;CONSO PI,
; 7011
; 7012      1720:
; 7013      GRP701:  UUU          ;BLKI PAG,
; 7014      1726:     UUU          ;CONSZ PAG,
; 7015      1727:     UUU          ;CONSO PAG,
; 7016
; 7017      ;680I AND CACHE SWEEP STUFF
; 7018      1730:     UUU
; 7019      1731:     UUU
; 7020      1732:     UUU
; 7021      1733:     UUU
; 7022      1734:     UUU
; 7023      1735:     UUU
; 7024      1736:     UUU
; 7025      1737:     UUU
; 7026
; 7027      APRSO:    [BR]_[BR].AND.# CLR LH, #/7770
; 7028      CONSO:    [BR].AND.[AR], SKIP ADR.EQ.O, J/SKIP
; 7029
; 7030      APRSZ:    [BR]_[BR].AND.# CLR LH, #/7770
; 7031      CONSZ:    [BR].AND.[AR], SKIP ADR.EQ.O, J/DONE
; 7032
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 193
IO -- INTERNAL DEVICES

U 1700, 0137,3771,0005,4374,4007,0700,0000,0001,0001

U 0137, 3513,3771,0005,4374,0007,0700,0000,0016,0124

U 1704, 3407,3771,0005,7274,4007,0701,0000,0000,0230

U 3407, 3410,5551,0505,4370,4007,0700,0000,0000,0007
U 3410, 3411,4551,0304,4374,4007,0700,0000,0000,0007
U 3411, 3412,3111,0405,4174,4007,0700,0000,0000,0000

U 3412, 3413,4551,0304,4374,4007,0700,0000,0000,7760
U 3413, 2304,4553,0300,4374,4007,0331,0000,0010,0000
U 2304, 2305,3111,0405,4174,4007,0700,0000,0000,0000
U 2305, 2306,4553,0300,4374,4007,0331,0000,0004,0000
U 2306, 2307,5111,0405,4174,4007,0700,0000,0000,0000
U 2307, 3414,3771,0006,4304,4007,0701,0000,0000,0000
U 3414, 2310,4553,0300,4374,4007,0331,0000,0002,0000
U 2310, 2311,5111,0406,4174,4007,0700,0000,0000,0000
U 2311, 2312,4553,0300,4374,4007,0331,0000,0001,0000
U 2312, 2313,3111,0406,4174,4007,0700,0000,0000,0000
U 2313, 2314,4553,0300,4374,4007,0331,0000,0003,0000

U 2314, 3417,3333,0006,4174,4007,0700,0000,0000,0000
U 2315, 3415,3333,0005,4174,4007,0700,0000,0000,0000

U 3415, 3416,3333,0005,4174,4257,0700,0000,0000,0000

U 3416, 1400,3333,0005,7174,4007,0700,0400,0000,0230

U 3417, 3420,3333,0006,4174,4237,0700,0000,0000,0000

U 3420, 3421,5551,0606,4370,4007,0700,0000,0000,2000

U 3421, 2315,3333,0006,4174,4237,0700,0000,0000,0000

```
; 7033 1700:
; 7034 GRP700:
; 7035 APRID: [BR]_#,
; 7036 #/4097.
; 7037 137: [BR]_#,
; 7038 MICROCODE OPTION(INHCST)/OPT,
; 7039 MICROCODE OPTION(NOCST)/OPT,
; 7040 MICROCODE OPTION(NONSTD)/OPT,
; 7041 MICROCODE OPTION(UBABLT)/OPT,
; 7042 MICROCODE OPTION(KIPAGE)/OPT,
; 7043 MICROCODE OPTION(KLPAGE)/OPT,
; 7044 MICROCODE VERSION/UCV,
; 7045 HOLD RIGHT,
; 7046 J/RTNREG
; 7047
; 7048 1704:
; 7049 WRAPR: [BR]_WORK[APR]
; 7050 [BR]_[BR].AND.NOT.#, ;CLEAR THE OLD PIA
; 7051 #/7, HOLD LEFT ;...
; 7052 [ARX]_[AR].AND.#, #/7 ;PUT NEW PIA IN ARX
; 7053 [BR]_[BR].OR.[ARX] ;PUT NEW PIA IN BR
; 7054 [ARX]_[AR].AND.#, ;MASK THE DATA BITS
; 7055 #/007760 ;DOWN TO ENABLES
; 7056 TR [AR], #/100000 ;WANT TO ENABLE ANY?
; 7057 =0 [BR]_[BR].OR.[ARX] ;YES--SET THEM
; 7058 TR [AR], #/40000 ;WANT TO DISABLE ANY?
; 7059 =0 [BR]_[BR].AND.NOT.[ARX] ;YES--CLEAR THEM
; 7060 [BRX]_APR ;GET CURRENT STATUS
; 7061 TR [AR], #/20000 ;WANT TO CLEAR FLAGS?
; 7062 =0 [BRX]_[BRX].AND.NOT.[ARX] ;YES--CLEAR BITS
; 7063 TR [AR], #/10000 ;WANT TO SET ANY FLAGS?
; 7064 =0 [BRX]_[BRX].OR.[ARX] ;YES--SET FLAGS
; 7065 TR [AR], #/30000 ;ANY CHANGE AT ALL?
; 7066 =0 READ [BRX], ;YES--LOAD NEW FLAGS
; 7067 J/WRAPR2 ;TURN OFF INTERRUPT 8080
; 7068 WRAPR1: READ [BR] ;FIX DPM TIMING BUG
; 7069 READ [BR], ;ENABLE CONDITIONS
; 7070 SET APR ENABLES
; 7071 WORK[APR]_[BR], ;SAVE FOR RDAPR
; 7072 J/DONE ;ALL DONE
; 7073
; 7074 WRAPR2: READ [BRX], ;LOAD NEW FLAGS
; 7075 SPEC/APR FLAGS ;...
; 7076 [BRX]_[BRX].AND.NOT.#, ;CLEAR INTERRUPT THE 8080
; 7077 #/002000, HOLD LEFT ;FLAG
; 7078 READ [BRX], ;LOAD NEW FLAGS
; 7079 SPEC/APR FLAGS, ;...
; 7080 J/WRAPR1 ;LOOP BACK
; 7081
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 194
IO -- INTERNAL DEVICES

```
U 1705, 3422,3771,0005,7274,4007,0701,0000,0000,0230 ; 7082 1705:
; 7083 RDAPR: [BR]_WORK[APR]
; 7084 [BR]_[BR] SWAP, ;PUT ENABLES IN BOTH
; 7085 HOLD_RIGHT ; HALVES
; 7086 [BR]_[BR].AND.#, ;SAVE ENABLES IN LH
; 7087 #/7760, ;
; 7088 HOLD_RIGHT
; 7089 [BR]_[BR].AND.#, ;SAVE PIA IN RH
; 7090 #/7,
; 7091 HOLD_LEFT
; 7092 [ARX]_APR ;READ THE APR FLAGS
; 7093 [ARX]_[ARX].AND.# CLR LH, ;MASK OUT JUNK
; 7094 #/007770 ;KEEP 8 FLAGS
; 7095 [BR]_[BR].OR.[ARX], ;MASH THE STUFF TOGETHER
; 7096 J/RTNREG ;RETURN
; 7097
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 195
IO -- INTERNAL DEVICES -- EBR & UBR

```

; 7098 .TOC "IO -- INTERNAL DEVICES -- EBR & UBR"
; 7099
; 7100 1723:
; 7101 WRUBR: VMA [AR], ;LOAD E INTO VMA
; 7102 START READ ;START MEMORY
; 7103 MEM READ, ;WAIT FOR DATA
; 7104 [AR]_MEM, 3T, ;PUT IT INTO THE AR
; 7105 SKIP DPO ;SEE IF WE WANT TO LOAD
; 7106 ; AC BLOCK NUMBERS
; 7107 =0 [AR]_[AR].AND.#, ;NO--CLEAR JUNK IN AR
; 7108 #/100000, ; LEAVE ONLY LOAD UBR
; 7109 HOLD RIGHT, ; IN LEFT HALF
; 7110 SKIP ADL.EQ.O, 3T, ;SEE IF WE WANT TO LOAD
; 7111 J/ACBSET ;SKIP AROUND UBR LOAD
; 7112 [UBR]_[UBR].AND.#, ;MASK OUT THE OLD
; 7113 #/770077, ; AC BLOCK NUMBERS
; 7114 HOLD RIGHT ;IN THE LEFT HALF
; 7115 [AR].AND.#, ;SEE IF WE WANT TO LOAD
; 7116 #/100000, 3T, ; UBR ALSO
; 7117 SKIP ADL.EQ.O
; 7118 =0
; 7119 ACBSET: [BR]_[AR].AND.#, ;COPY UBR PAGE NUMBER
; 7120 #/17777, ; INTO BR
; 7121 J/SETUBR ;GO LOAD UBR
; 7122 [UBR]_[UBR].OR.[AR], ;DO NOT LOAD UBR
; 7123 ; PUT AC BLOCK # IN
; 7124 HOLD RIGHT, ; THE LEFT HALF
; 7125 LOAD AC BLOCKS, ;LOAD HARDWARE
; 7126 J/DONE ;ALL DONE
; 7127 SETUBR: [BR]_O, ;CLEAR BR LEFT
; 7128 SC_7, ;PUT THE COUNT IN SC
; 7129 HOLD RIGHT
; 7130 =0
; 7131 STUBRS: [BR]_[BR]*2, ;SHIFT BR OVER
; 7132 STEP SC, ; 9 PLACES
; 7133 J/STUBRS
; 7134 [UBR]_[UBR].AND.#, ;MASK OUT OLD UBR
; 7135 #/777774, ; BITS IN
; 7136 HOLD RIGHT ; LEFT HALF
; 7137 [UBR]_O, ;CLEAR RIGHT HALF
; 7138 HOLD LEFT
; 7139 [UBR]_[UBR].OR.[BR] ;PUT IN PAGE TABLE ADDRESS
; 7140 [UBR]_[UBR].OR.[AR], ;PUT IN AC BLOCK #
; 7141 HOLD RIGHT, ; IN LEFT HALF
; 7142 LOAD AC BLOCKS, ;TELL HARDWARE
; 7143 J/SWEEP ;CLEAR CACHE
; 7144
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 196
IO -- INTERNAL DEVICES -- EBR & UBR

```
U 1724, 2324,3445,0303,4174,4007,0700,2000,0071,0006 ; 7145 1724:
; 7146 WREBR: [AR]_[AR]*2, SC_6 ;DO A SHIFT OVER 8 MORE
; 7147 =0
U 2324, 2324,3445,0303,4174,4007,0630,2000,0060,0000 ; 7148 WREBR1: [AR]_[AR]*2, STEP SC, J/WREBR1 ;SKIP WHEN = -1
; 7149 .IF/FULL ;DO NOT ENABLE PAGING IN SMALL
; 7150 ; MICROCODE.
; 7151 [BR]_WORK[APR]
U 2325, 3436,3771,0005,7274,4007,0701,0000,0000,0230 ; 7152 [BR]_[BR].AND.#, #/747777, HOLD LEFT
U 3436, 3437,4551,0505,4370,4007,0700,0000,0074,7777 ; 7153 [AR].AND.#, #/20, 3T, SKIP ADL.EQ.O ;BIT 22 - TRAP ENABL
U 3437, 2326,4553,0300,4374,4007,0321,0000,0000,0020 ; 7154 =0 [BR]_[BR].OR.#, #/030000, HOLD LEFT ;SET - ALLOW TRAPS T
U 2326, 2327,3551,0505,4370,4007,0700,0000,0003,0000 ; 7155 READ [BR], SET APR ENABLES
HAPPEN ; 7156 WORK[APR]_[BR]
U 2327, 3440,3333,0005,4174,4257,0700,0000,0000,0000 ; 7157 .ENDIF/FULL
U 3440, 3441,3333,0005,7174,4007,0700,0400,0000,0230 ; 7158 ;
; 7159 .IF/KIPAGE
; 7160 .IF/KLPAGE
; 7161 [EBR]_[AR] ;NOTE: SHIFTED LEFT 9 BITS
; 7162 [EBR].AND.#, #/40, 3T, SKIP ADL.EQ.O ;BIT 21 - KL PAGING
NABLE ;:7163 =0 [EBR]_[EBR].OR.#, #/400000, HOLD RIGHT, J/SWEEP ;YES, SET IN
ERNAL FLAG ;:7164 [EBR]_[EBR].AND.NOT.#, #/400000, HOLD RIGHT, J/SWEEP ;NO, CL
BIT 0 ;
; 7165 .ENDIF/KLPAGE
; 7166 .ENDIF/KIPAGE
; 7167
; 7168 .IFNOT/KLPAGE ;MUST BE KI ONLY
U 3441, 2372,3441,0310,4174,4007,0700,0000,0000,0000 ; 7169 [EBR]_[AR],J/SWEEP ;SO INTERNAL FLAG ISN'T USED
; 7170 .ENDIF/KLPAGE
; 7171
; 7172 .IFNOT/KIPAGE ;MUST BE KL ONLY
; 7173 [EBR]_[AR],J/SWEEP ;SO INTERNAL FLAG ISN'T USED
; 7174 .ENDIF/KIPAGE
; 7175
; 7176 1725:
U 1725, 2330,3447,1005,4174,4007,0700,2000,0071,0006 ; 7177 RDEBR: [BR]_[EBR]*.5, SC_6
; 7178 =0
; 7179 RDEBR1: [BR]_[BR]*.5, STEP SC, J/RDEBR1
U 2330, 2330,3447,0505,4174,4007,0630,2000,0060,0000 ; 7180 [BR]_[BR].AND.#, #/63777 ;MASK TO JUST EBR
U 2331, 3442,4551,0505,4374,4007,0700,0000,0006,3777 ; 7181 [BR]_O, ;CLEAR LEFT HALF
; 7182 HOLD RIGHT, ; BITS
; 7183 J/RTNREG ;RETURN ANSWER
; 7184
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 197
IO -- INTERNAL DEVICES -- EBR & UBR

U 1721, 2332,3441,1105,4174,4007,0700,0000,0000,0000.
U 2332, 2334,3447,0506,4174,4007,0700,2010,0071,0006

U 2333, 3513,4551,0505,4374,0007,0700,0000,0050,7700

U 3443, 3444,3441,1105,4174,4007,0700,0000,0000,0000
U 3444, 2334,3447,0506,4174,4007,0700,2000,0071,0006

U 2334, 2334,3447,0606,4174,4007,0630,2000,0060,0000
U 2335, 3445,4551,0606,4374,4007,0700,0000,0001,7777
U 3445, 0001,3441,0605,4170,4004,1700,0000,0000,0000

; 7185 1721:
; 7186 RDUBR: [BR]_[UBR]
; 7187 =0 [BRX]_[BR]*.5, SC_6, CALL [GTPCW1]
; 7188 [BR]_[BR].AND.#, ;JUST RETURN USEFUL
; 7189 #/507700, HOLD RIGHT, ; BITS
; 7190 J/RTNREG
; 7191
; 7192
; 7193 GETPCW: [BR]_[UBR]
; 7194 [BRX]_[BR]*.5, SC_6
; 7195 =0
; 7196 GTPCW1: [BRX]_[BRX]*.5, STEP SC, J/GTPCW1
; 7197 [BRX]_[BRX].AND.#, #/17777
; 7198 [BR]_[BRX], HOLD LEFT, RETURN [1]
; 7199

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 198
IO -- INTERNAL DEVICES -- KL PAGING REGISTERS

```
D 0702, 1216,1760,4700
; 7200 .TOC "IO -- INTERNAL DEVICES -- KL PAGING REGISTERS"
; 7201
; 7202 .DCODE
; 7203 702: IOT,AC DISP, M, J/GRP702
; 7204 .UCODE
; 7205
; 7206 1760:
; 7207 GRP702:
; 7208 RDSPB: [BR]_WORK[SBR], J/RTNREG
; 7209 1761:
; 7210 RDCSB: [BR]_WORK[CBR], J/RTNREG
; 7211 1762:
; 7212 RDPUR: [BR]_WORK[PUR], J/RTNREG
; 7213 1763:
; 7214 RDCSTM: [BR]_WORK[CSTM], J/RTNREG
; 7215 1766:
; 7216 RDHSB: [BR]_WORK[HSBADR], J/RTNREG
; 7217 1767: UUU
; 7218
; 7219 1770:
; 7220 WRSPB: START READ
; 7221 MEM READ, [AR]_MEM
; 7222 WORK[SBR]_[AR], J/DONE
; 7223 1771:
; 7224 WRCSB: START READ
; 7225 MEM READ, [AR]_MEM
; 7226 WORK[CBR]_[AR], J/DONE
; 7227 1772:
; 7228 WRPUR: START READ
; 7229 MEM READ, [AR]_MEM
; 7230 WORK[PUR]_[AR], J/DONE
; 7231 1773:
; 7232 WRCSTM: START READ
; 7233 MEM READ, [AR]_MEM
; 7234 WORK[CSTM]_[AR], J/DONE
; 7235 1776:
; 7236 WRHSB: START READ
; 7237 MEM READ, [AR]_MEM
; 7238 WORK[HSBADR]_[AR], J/DONE
; 7239 1777: UUU
; 7240
```

U 1760, 3513,3771,0005,7274,4007,0701,0000,0000,0215
U 1761, 3513,3771,0005,7274,4007,0701,0000,0000,0216
U 1762, 3513,3771,0005,7274,4007,0701,0000,0000,0220
U 1763, 3513,3771,0005,7274,4007,0701,0000,0000,0217
U 1766, 3513,3771,0005,7274,4007,0701,0000,0000,0227
U 1767, 2621,4551,0202,4374,0007,0700,0000,0077,7740

U 1770, 3446,4443,0000,4174,4007,0700,0200,0004,0002
U 3446, 3447,3771,0003,4365,5007,0700,0200,0000,0002
U 3447, 1400,3333,0003,7174,4007,0700,0400,0000,0215

U 1771, 3450,4443,0000,4174,4007,0700,0200,0004,0002
U 3450, 3451,3771,0003,4365,5007,0700,0200,0000,0002
U 3451, 1400,3333,0003,7174,4007,0700,0400,0000,0216

U 1772, 3452,4443,0000,4174,4007,0700,0200,0004,0002
U 3452, 3453,3771,0003,4365,5007,0700,0200,0000,0002
U 3453, 1400,3333,0003,7174,4007,0700,0400,0000,0220

U 1773, 3454,4443,0000,4174,4007,0700,0200,0004,0002
U 3454, 3455,3771,0003,4365,5007,0700,0200,0000,0002
U 3455, 1400,3333,0003,7174,4007,0700,0400,0000,0217

U 1776, 3456,4443,0000,4174,4007,0700,0200,0004,0002
U 3456, 3457,3771,0003,4365,5007,0700,0200,0000,0002
U 3457, 1400,3333,0003,7174,4007,0700,0400,0000,0227
U 1777, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 199
IO -- INTERNAL DEVICES -- TIMER CONTROL

```
; 7241 .TOC "IO -- INTERNAL DEVICES -- TIMER CONTROL"
; 7242
; 7243
; 7244
; 7245 TICK: [AR]_WORK[TIME1], ;BEGIN [123]
; 7246 SPEC/CLRCLK ;GET LOW WORD
; 7247 ;CLEAR CLOCK FLAG
; 7248 ;END [123]
U 3460, 3461,3771,0003,7274,4117,0701,0000,0000,0301
; 7249 TOCK: [BR]_O XWD [10000] ;2^12 UNITS PER MS
; 7250 [AR]_[AR]+[BR] ;INCREMENT THE TIMER
; 7251 FIX [AR] SIGN, SKIP DPO ;SEE IF IT OVERFLOWED
; 7252 =0
; 7253 TOCK1: WORK[TIME1]_[AR], ;STORE THE NEW TIME
; 7254 J/TOCK2 ;SKIP OVER THE OVERFLOW CODE
; 7255 [AR]_WORK[TIME0] ;GET HIGH WORD
; 7256 =0* [AR]_[AR]+1, ;BUMP IT
; 7257 CALL [WRTIM1] ;STORE BACK IN RAM
; 7258 [AR]_O, ;CAUSE LOW WORD WORD
; 7259 J/TOCK1 ; TO GET STORED
U 2336, 3464,3333,0003,7174,4007,0700,0400,0000,0301
U 2337, 2340,3771,0003,7274,4007,0701,0000,0000,0300
; 7260 TOCK2: [AR]_WORK[TTG]
; 7261 [AR]_[AR]-[BR], ;COUNT DOWN TIME TO GO
; 7262 SKIP AD.LE.O ;SEE IF IT TIMED OUT
; 7263 =0
; 7264 TOCK3: WORK[TTG]_[AR], ;SAVE NEW TIME TO GO
; 7265 RETURN [2] ;ALL DONE
; 7266 [AR]_WORK[PERIOD]
; 7267 [BR]_APR ;GET CURRENT FLAGS
; 7268 [BR]_[BR].OR.#, #/40 ;SET TIMER INTERRUPT FLAG
; 7269 READ [BR], ;PLACE ON DP AND
; 7270 SPEC/APR FLAGS, ;LOAD INTO HARDWARE
; 7271 J/TOCK3 ;ALL DONE
```


; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 200
IO -- INTERNAL DEVICES -- WRTIME & RDTIME

```
; 7272 .TOC "IO -- INTERNAL DEVICES -- WRTIME & RDTIME"
; 7273
; 7274 1774:
U 1774, 3471,4443,0000,4174,4007,0700,0200,0004,0002 ; 7275 WRTIME: START READ ;FETCH WORD AT E
; 7276 MEM READ, ;WAIT FOR DATA
U 3471, 1044,3771,0003,4365,5007,0700,0200,0000,0002 ; 7277 [AR] MEM ;PUT WORD IN AR
; 7278 =00 VMA [HR]+1, ;BUMP E
; 7279 START READ, ;START MEMORY
U 1044, 3563,0111,0702,4170,4007,0700,0210,0004,0012 ; 7280 CALL [LOADARX] ;PUT DATA IN ARX
; 7281 [ARX]_[ARX].AND.#, ;CLEAR PART HELD IN
; 7282 #/770000, ;HARDWARE COUNTER
U 1045, 3472,4551,0404,4370,4007,0700,0010,0077,0000 ; 7283 HOLD LEFT, CALL [WRTIM1]
; 7284 =11 WORK[TIME1]_[ARX], ;IN WORK SPACE
; 7285 J/DONE ;NEXT INSTRUCTION
U 1047, 1400,3333,0004,7174,4007,0700,0400,0000,0301 ; 7286 =
; 7287 WRTIM1: WORK[TIME0]_[AR], ;SAVE THE NEW VALUE
; 7288 RETURN [2]
; 7289
; 7290 1764:
U 1764, 3473,4451,1205,4324,4007,0700,0000,0000,0000 ; 7291 RDTIME: [BR]_TIME ;READ THE TIME
U 3473, 3474,4451,1204,4324,4007,0700,0000,0000,0000 ; 7292 [ARX]_TIME ; AGAIN
U 3474, 3475,4451,1206,4324,4007,0700,0000,0000,0000 ; 7293 [BRX]_TIME ; AGAIN
; 7294 [BR].XOR.[ARX], ;SEE IF STABLE
; 7295 SKIP AD.EQ.O ;...
U 3475, 2346,6113,0405,4174,4007,0621,0000,0000,0000 ; 7296 =0 [ARX]_[BRX] ;NO THEN NEXT TRY MUST BE OK
U 2346, 2347,3441,0604,4174,4007,0700,0000,0000,0000 ; 7297 [BR]_WORK[TIME0]
U 2347, 3476,3771,0005,7274,4007,0701,0000,0000,0300 ; 7298 [ARX]_[ARX]+WORK[TIME1], ;COMBINE PARTS
; 7299 SKIP/-1 MS ;SEE IF OVERFLOW HAPPENED
U 3476, 1050,0551,0404,7274,4007,0671,0000,0000,0301 ; 7300 =00 SPEC/CLRCLK, ;CLEAR CLOCK FLAG
; 7301 [AR]_WORK[TIME1], 2T, ;GET LOW WORD FOR TOCK
; 7302 CALL [TOCK] ;UPDATE CLOCKS
; 7303 READ [HR], LOAD VMA, ;DID NOT OVERFLOW
; 7304 START WRITE, J/RDTIM1 ;STORE ANSWER
; 7305 J/RDTIME ;TRY AGAIN
; 7306 =
; 7307 RDTIM1: MEM WRITE, MEM [BR]
U 3477, 3500,3333,0005,4175,5007,0701,0200,0000,0002 ; 7308 VMA [HR]+1, LOAD VMA, START WRITE
U 3500, 3501,0111,0702,4170,4007,0700,0200,0003,0012 ; 7309 MEM WRITE, MEM [ARX], J/DONE
U 3501, 1400,3333,0004,4175,5007,0701,0200,0000,0002 ; 7310
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 201
IO -- INTERNAL DEVICES -- WRINT & RDINT

U 1775, 3502,4443,0000,4174,4007,0700,0200,0004,0002
U 3502, 3503,3771,0003,4365,5007,0700,0200,0000,0002
U 3503, 3504,3333,0003,7174,4007,0700,0400,0000,0302
U 3504, 1400,3333,0003,7174,4007,0700,0400,0000,0303
U 1765, 3513,3771,0005,7274,4007,0701,0000,0000,0302

; 7311 .TOC "IO -- INTERNAL DEVICES -- WRINT & RDINT"
; 7312
; 7313
; 7314 1775:
; 7315 WRINT: START READ
; 7316 MEM READ, [AR]_MEM
; 7317 WORK[PERIOD]_[AR]
; 7318 WORK[TTG]_[AR],
; 7319 J/DONE
; 7320
; 7321 1765:
; 7322 RDINT: [BR]_WORK[PERIOD],
; 7323 J/RTNREG
; 7324

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 202
IO -- INTERNAL DEVICES -- RDPI & WRPI

```
U 1715, 3513,3441,1405,4174,4007,0700,0000,0000,0000
; 7325 .TOC "IO -- INTERNAL DEVICES -- RDPI & WRPI"
; 7326
; 7327 1715:
; 7328 RDPI: [BR]_[PI], J/RTNREG
; 7329
; 7330 1714:
; 7331 WRPI: TR [AR], PI.CLR/1
; 7332 =0 [PI]_O
; 7333 TR [AR], PI.MBZ/17
; 7334 =0 UUD
; 7335 [BR]_[AR].AND.#,#/177
; 7336 [BR]_[BR] SWAP, HOLD RIGHT
; 7337 TR [AR], PI.DIR/1
; 7338 =0 [PI]_[PI].AND.NOT.[BR], HOLD RIGHT
; 7339 TR [AR], PI.REQ/1
; 7340 =0 [PI]_[PI].OR.[BR], HOLD RIGHT
; 7341 TR [AR], PI.TSN/1
; 7342 =0 [PI]_[PI].OR.#,PI.ON/1, HOLD LEFT
; 7343 TR [AR], PI.TSF/1
; 7344 =0 [PI]_[PI].AND.NOT.#,PI.ON/1, HOLD LEFT
; 7345 TR [AR], PI.TCN/1
; 7346 =0 [PI]_[PI].OR.[BR], HOLD LEFT
; 7347 TR [AR], PI.TCF/1
; 7348 =0**0 [PI]_[PI].AND.NOT.[BR], HOLD LEFT
; 7349 PIEXIT: CALL LOAD PI
; 7350 =1**1
; 7351 DONE
; 7352 =
; 7353
; 7354 ;SUBROUTINE TO LOAD PI HARDWARE
; 7355 ;CALL WITH:
; 7356 ; CALL LOAD PI
; 7357 ;RETURNS 10 WITH PI HARDWARE LOADED
; 7358 ;
; 7359 LOADPI: [TO]_[PI] SWAP ;PUT ACTIVE CHANS IN LH
; 7360 LDPI2: [TO]_-1, HOLD LEFT ;DONT MASK RH
; 7361 [TO]_[TO].AND.[PI] ;ONLY REQUEST CHANS THAT ARE ON
; 7362 .NOT.[TO], LOAD PI, ;RELOAD HARDWARE
; 7363 RETURN [10] ;RETURN TO CALLER
; 7364

U 1714, 2350,4553,0300,4374,4007,0331,0000,0001,0000
U 2350, 2351,4221,0014,4174,4007,0700,0000,0000,0000
U 2351, 2352,4553,0300,4374,4007,0331,0000,0074,0000
U 2352, 2621,4551,0202,4374,0007,0700,0000,0077,7740
U 2353, 3505,4551,0305,4374,4007,0700,0000,0000,0177
U 3505, 3506,3770,0505,4344,0007,0700,0000,0000,0000
U 3506, 2354,4553,0300,4374,4007,0331,0000,0002,0000
U 2354, 2355,5111,0514,4174,0007,0700,0000,0000,0000
U 2355, 2356,4553,0300,4374,4007,0331,0000,0000,4000
U 2356, 2357,3111,0514,4174,0007,0700,0000,0000,0000
U 2357, 2360,4553,0300,4374,4007,0331,0000,0000,0200
U 2360, 2361,3551,1414,4370,4007,0700,0000,0000,0200
U 2361, 2362,4553,0300,4374,4007,0331,0000,0000,0400
U 2362, 2363,5551,1414,4370,4007,0700,0000,0000,0200
U 2363, 2364,4553,0300,4374,4007,0331,0000,0000,2000
U 2364, 2365,3111,0514,4170,4007,0700,0000,0000,0000
U 2365, 0304,4553,0300,4374,4007,0331,0000,0000,1000
U 0304, 0305,5111,0514,4170,4007,0700,0000,0000,0000
U 0305, 3510,3770,1416,4344,4007,0700,0010,0000,0000

U 0315, 0110,3443,0100,4174,4156,4700,0200,0014,0012
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 203
IO -- INTERNAL DEVICES -- SUBROUTINES

U 3513, 3514, 3443, 0300, 4174, 4007, 0700, 0200, 0003, 0012
U 3514, 1400, 3333, 0005, 4175, 5007, 0701, 0200, 0000, 0002

; 7365 .TOC "IO -- INTERNAL DEVICES -- SUBROUTINES"
; 7366
; 7367
; 7368 ;HERE WITH SOMETHING IN BR STORE IT @AR
; 7369 RTNREG: VMA_[AR], START WRITE
; 7370 MEM WRITE, MEM_[BR], J/DONE
; 7371

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 204
IO -- INTERNAL DEVICES -- SUBROUTINES

```
U 1722, 3515,3443,0300,4174,4147,0700,0200,0000,0010
U 3515, 2366,4221,0003,4174,4007,0700,0000,0000,0000
U 2366, 3521,3771,0003,4374,4007,0700,0010,0037,7377
U 2367, 3516,3771,0005,4374,4247,0700,0000,0000,1001
U 3516, 2370,3333,0003,4174,4247,0700,0000,0000,1000

U 2370, 2370,1111,0503,4174,4247,0630,6000,0060,1000
U 2371, 2375,3333,0003,4174,4007,0700,0000,0000,0000

U 2372, 3521,3771,0003,4374,4007,0700,0010,0037,7377
U 2373, 3517,3771,0005,4374,4347,0700,0000,0000,1001
U 3517, 2374,3333,0003,4174,4347,0700,0000,0000,1000

U 2374, 2374,1111,0503,4174,4347,0630,6000,0060,1000
U 2375, 3520,4223,0000,7174,4007,0700,0400,0000,0424
U 3520, 1400,4223,0000,7174,4007,0700,0400,0000,0423

U 3521, 0001,4443,0000,4174,4004,1700,2000,0071,0375

; 7372 ;CACHE SWEEP
; 7373
; 7374 1722:
; 7375 CLRPT: VMA [AR], ;PUT CORRECT ADDRESS IN VMA
; 7376 LOAD PAGE TABLE ;GET SET TO WRITE PAGE TABLE
; 7377 [AR]_O ;CLEAR ENTRY
; 7378 =O [AR]_#, #/377377, ;INITIAL VMA VALUE
; 7379 CALL [SSWEEP] ;LOAD THE SC
; 7380 [BR]_#, #/1001, ;CONSTANT TO KEEP ADDING
; 7381 CLRC SH ;START TO CLEAR CACHE
; 7382 READ [AR], CLRC SH ;FIRST THING TO CLEAR
; 7383 =O
; 7384 CLRPTL: [AR]_[AR]-[BR], ;UPDATE AR (AND PUT ON DP)
; 7385 CLRC SH, ;SWEEP ON NEXT STEP
; 7386 STEP SC, ;SKIP IF WE ARE DONE
; 7387 J/CLRPTL ;LOOP FOR ALL ENTRIES
; 7388 READ [AR], J/ZAPPTA ;CLEAR LAST ENTRY
; 7389
; 7390 =O
; 7391 SWEEP: [AR]_#, #/377377, ;INITIAL VMA VALUE
; 7392 CALL [SSWEEP] ;LOAD NUMBER OF STEPS INTO SC
; 7393 [BR]_#, #/1001, ;CONSTANT TO KEEP ADDING
; 7394 SWEEP ;START SWEEP
; 7395 READ [AR], SWEEP ;FIRST THING TO CLEAR
; 7396 =O
; 7397 SWEPL: [AR]_[AR]-[BR], ;UPDATE AR (AND PUT ON DP)
; 7398 SWEEP, ;SWEEP ON NEXT STEP
; 7399 STEP SC, ;SKIP IF WE ARE DONE
; 7400 J/SWEPL ;LOOP FOR ALL ENTRIES
; 7401 ;CLEAR LAST ENTRY AND
; 7402 ZAPPTA: WORK[PTA.U]_O ;FORGET PAGE TABLE ADDRESS
; 7403 WORK[PTA.E]_O, ;FORGET PAGE TABLE ADDRESS
; 7404 J/DONE ;ALL DONE
; 7405
; 7406 SSWEEP: SC S#, S#/375, ;NUMBER OF STEPS
; 7407 RETURN [1] ;RETURN
; 7408
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254) KS10 MICRDCODE V124, 27-JUL-84 Page 205
IO -- INTERNAL DEVICES -- SUBROUTINES

```
; 7409 ;WE COME HERE EITHER FROM NEXT INSTRUCTION DISPATCH OR PAGE FAIL
; 7410 ; LOGIC. IN ALL CASES, THE CURRENT INSTRUCTION IS CORRECTLY SETUP
; 7411 ; TO RESTART PROPERLY.
; 7412
; 7413 ;FIRST SET THE CORRECT PI IN PROGRESS BIT
; 7414 ; [FLG]_[FLG].OR.#,FLG.PI/1, HOLD RIGHT,
; 7415 ; J/PI ;SET PI CYCLE AND PROCESS PI
; 7416 =1000
; 7417 PI: AD/D, DBUS/PI NEW, ;LOOK AT NEW LEVEL
; 7418 DISP/DP LEFT, 3T, ;DISPATCH ON IT
; 7419 J/PI ;GO TO 1 OF NEXT 7 PLACES
U 0770, 0770,3773,0000,4074,4003,1701,0000,0000,0000
; 7420 =1001 [PI]_[PI].OR.#, #/040000, HOLD LEFT, J/PIP1
U 0771, 3522,3551,1414,4370,4007,0700,0000,0004,0000
; 7421 =1010 [PI]_[PI].OR.#, #/020000, HOLD LEFT, J/PIP2
U 0772, 3523,3551,1414,4370,4007,0700,0000,0002,0000
; 7422 =1011 [PI]_[PI].OR.#, #/010000, HOLD LEFT, J/PIP3
U 0773, 3524,3551,1414,4370,4007,0700,0000,0001,0000
; 7423 =1100 [PI]_[PI].OR.#, #/004000, HOLD LEFT, J/PIP4
U 0774, 3525,3551,1414,4370,4007,0700,0000,0000,4000
; 7424 =1101 [PI]_[PI].OR.#, #/002000, HOLD LEFT, J/PIP5
U 0775, 3526,3551,1414,4370,4007,0700,0000,0000,2000
; 7425 =1110 [PI]_[PI].OR.#, #/001000, HOLD LEFT, J/PIP6
U 0776, 3527,3551,1414,4370,4007,0700,0000,0000,1000
; 7426 =1111 [PI]_[PI].OR.#, #/000400, HOLD LEFT, J/PIP7
U 0777, 3530,3551,1414,4370,4007,0700,0000,0000,0400
; 7427 PIP1: [BRX]_O XWD [1], J/PI10 ;REMEMBER WE ARE AT LEVEL 1
U 3522, 3531,4751,1206,4374,4007,0700,0000,0000,0001
; 7428 PIP2: [BRX]_O XWD [2], J/PI10 ;REMEMBER WE ARE AT LEVEL 2
U 3523, 3531,4751,1206,4374,4007,0700,0000,0000,0002
; 7429 PIP3: [BRX]_O XWD [3], J/PI10 ;REMEMBER WE ARE AT LEVEL 3
U 3524, 3531,4751,1206,4374,4007,0700,0000,0000,0003
; 7430 PIP4: [BRX]_O XWD [4], J/PI10 ;REMEMBER WE ARE AT LEVEL 4
U 3525, 3531,4751,1206,4374,4007,0700,0000,0000,0004
; 7431 PIP5: [BRX]_O XWD [5], J/PI10 ;REMEMBER WE ARE AT LEVEL 5
U 3526, 3531,4751,1206,4374,4007,0700,0000,0000,0005
; 7432 PIP6: [BRX]_O XWD [6], J/PI10 ;REMEMBER WE ARE AT LEVEL 6
U 3527, 3531,4751,1206,4374,4007,0700,0000,0000,0006
; 7433 PIP7: [BRX]_O XWD [7], J/PI10 ;REMEMBER WE ARE AT LEVEL 7
U 3530, 3531,4751,1206,4374,4007,0700,0000,0000,0007
; 7434
; 7435 PI10: [AR]_[PI].AND.# CLR LH, ;TURN OFF PI SYSTEM
U 3531, 3532,4251,1403,4374,4007,0700,0000,0007,7577
; 7436 #/077577 ; TILL WE ARE DONE
U 3532, 3533,7443,0300,4174,4437,0700,0000,0000,0000
; 7437 .NOT.[AR], LOAD PI ; ..
U 3533, 2376,4223,0000,4364,4277,0700,0200,0000,0010
; 7438 ABORT MEM CYCLE ;NO MORE TRAPS
; 7439 =0 [AR]_VMA IO READ, ;SETUP TO READ WRU BITS
; 7440 WRU CYCLE/1, ; ..
; 7441 CALL [STRIO] ;START THE CYCLE
; 7442 MEM READ, ;WAIT FOR DATA
; 7443 [AR]_IO DATA, 3T, ;PUT DATA IN AR
; 7444 SKIP ADR.EQ.0 ;SEE IF ANYONE THERE
U 2377, 2400,3771,0003,4364,4007,0331,0200,0000,0002
; 7445 =0 [ARX]_O, J/VECINT ;YES--VECTORED INTERRUPT
U 2400, 3542,4221,0004,4174,4007,0700,0000,0000,0000
; 7446 [AR]_[BRX]*2 ;N*2
U 2401, 3534,3445,0603,4174,4007,0700,0000,0000,0000
; 7447 [AR]_[AR]+#, #/40, 3T, ;2*N+40
; 7448 HOLD LEFT ; ..
; 7449 [AR]_[AR]+[EBR], ;ABSOLUTE ADDRESS OF
; 7450 J/PI40 ; INTERRUPT INSTRUCTION
; 7451
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 206
IO -- INTERNAL DEVICES -- SUBROUTINES

```
U 3536, 3537,3443,0300,4174,4007,0700,0200,0024,1016 ; 7452 ;HERE WITH ABSOLUTE ADDRESS OF INTERRUPT INSTRUCTION IN [AR]
; 7453 PI40: VMA_[AR], VMA PHYSICAL READ ;FETCH THE INSTRUCTION
; 7454 PI50: MEM READ, [AR]_MEM, LOAD VMA, ;FETCH INSTRUCTION
; 7455 3T, FORCE EXEC ;E IS EXEC MODE
; 7456 [AR].XOR.#, #/254340, 3T, SKIP ADL.EQ.0
U 3537, 3540,3771,0003,4365,5007,0701,0200,0020,0012 ; 7457 =0 [AR].XOR.#, #/264000, SKIP ADL.EQ.0, 3T, J/PIJSR
U 3540, 2402,6553,0300,4374,4007,0321,0000,0025,4340 ; 7458 [BR]_FLAGS ;SAVE FLAGS
U 2402, 2404,6553,0300,4374,4007,0321,0000,0026,4000 ; 7459 AD/ZERO, LOAD FLAGS,
U 2403, 3541,4521,1205,4074,4007,0700,0000,0000,0000 ; 7460 J/PIXPCW ;ENTER EXEC MODE AND ASSUME
; 7461 ; WE HAVE AN XPCW
; 7462 ;IF WE HALT HERE ON A VECTORED INTERRUPT, WE HAVE
; 7463 ; TO/ WHAT WE READ FROM BUS AS VECTOR
; 7464 ; ARX/ EPT+100+DEVICE
; 7465 ; BR/ ADDRESS OF ILLEGAL INSTRUCTION
; 7466 ; BRX/ VECTOR (MASKED AND SHIFTED)
; 7467 =0
U 2404, 0104,4751,1217,4374,4007,0700,0000,0000,0101 ; 7468 PIJSR: HALT [ILLII] ;NOT A JSR OR XPCW
U 2405, 0470,4443,0000,4174,4007,0700,0200,0023,0002 ; 7469 START WRITE, FORCE EXEC ;PREPARE TO STORE OLD PC
; 7470 =0*0 [BR]_PC WITH FLAGS, ;OLD PC
; 7471 CALL [STOBR] ;STORE OLD PC
; 7472 =1*0 [AR]_#, #/0, HOLD RIGHT, ;PREPARE TO CLEAR FL
GS
U 0474, 3566,3771,0003,4374,0007,0700,0010,0000,0000 ; 7473 CALL [INCAR] ;BUMP POINTER
; 7474 =1*1 [PC]_[AR], LOAD FLAGS, ;NEW PC
; 7475 J/PISET ;CLEAR PI CYCLE & START
; 7476 ; INTERRUPT PROGRAM
; 7477 =
; 7478
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 207
ID -- INTERNAL DEVICES -- SUBROUTINES

U 3542, 2406,3445,0303,4174,4007,0530,0000,0000,0000
U 2406, 3542,0111,1504,4174,4007,0700,0000,0000,0000
U 2407, 2410,4571,1203,4374,4007,0700,0000,0024,1240
U 2410, 3570,3111,0403,4174,4007,0700,0010,0000,0000
U 2411, 2412,3771,0016,4364,4007,0700,0200,0000,0002
U 2412, 3565,0551,1005,4374,4007,0701,0010,0000,0100
U 2413, 3543,0111,0504,4174,4007,0700,0200,0024,1016
U 3543, 2414,3771,0005,4365,5007,0331,0200,0000,0002
U 2414, 3544,4557,1606,4374,4007,0701,0000,0000,0774
U 2415, 0104,4751,1217,4374,4007,0700,0000,0000,0102
U 3544, 3545,3447,0606,4174,4007,0700,0000,0000,0000

U 3545, 3537,0111,0605,4174,4007,0700,0200,0024,0012

```
; 7479 ;HERE TO PROCESS A VECTORED INTERRUPT. AT THIS POINT:
; 7480 ; AR/ WRU BITS (BIT 18 FOR DEVICE 0)
; 7481 ; ARX/ 0
; 7482 VECINT: [AR]_[AR]*2, ;SHIFT LEFT (UNSHIFTED ON DP)
; 7483 SKIP DP18 ;ANYONE THERE?
; 7484 =0 [ARX]_[ARX]+[XWD1], ;NO--BUMP BOTH HALVES
; 7485 J/VECINT ;KEEP LOOKING
; 7486 [AR]_VMA IO READ, ;SETUP FOR VECTOR CYCLE
; 7487 VECTOR CYCLE/1 ;...
; 7488 =0 [AR]_[AR].OR.[ARX], ;PUT IN UNIT NUMBER
; 7489 CALL [STRIO] ;START CYCLE
; 7490 MEM READ, ;WAIT FOR VECTOR (SEE DPM5)
; 7491 [TO] IO DATA ;GET VECTOR
; 7492 =0 [BR]_[EBR]+#, 3T, #/100, ;EPT+100
; 7493 CALL [CLARXL] ;CLEAR ARX LEFT
; 7494 [ARX]_[ARX]+[BR], ;EPT+100+DEVICE
; 7495 VMA PHYSICAL READ ;FETCH WORD
; 7496 MEM READ, [BR]_MEM, 3T, ;GET POINTER
; 7497 SKIP ADR.EQ.0 ;SEE IF NON-ZERO
; 7498 =0 [BRX]_[([TO].AND.#)*.5, 3T, ;OK--MAKE VECTOR MOD 400
; 7499 #/774, J/VECIN1 ; AND SHIFT OVER
; 7500 HALT [ILLINT]
; 7501 VECIN1: [BRX]_[BRX]*.5 ;SHIFT 1 MORE PLACE
; 7502 [BR]_[BR]+[BRX], ;ADDRESS OF WORD TO USE
; 7503 LOAD VMA, FORCE EXEC, ;FORCE EXEC VIRTUAL ADDRESS
; 7504 START READ, J/PI50 ;GO GET INSTRUCTION
; 7505
```


; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 208
PRIORITY INTERRUPTS -- DISMISS SUBROUTINE

U 2416, 3546,3771,0005,4374,4007,0700,0000,0004,0000
U 2417, 0004,4443,0000,4174,4004,1700,0000,0000,0000

U 3546, 2420,4113,0514,4174,4007,0330,0000,0000,0000
U 2420, 0004,5111,0514,4170,4004,1700,0000,0000,0000
U 2421, 3546,3447,0505,4174,4007,0700,0000,0000,0000

```
; 7506 .TOC "PRIORITY INTERRUPTS -- DISMISS SUBROUTINE"  
; 7507  
; 7508 ;SUBROUTINE TO DISMISS THE HIGHEST PI IN PROGRESS  
; 7509 ;RETURNS 4 ALWAYS  
; 7510  
; 7511 ;DISMISS:  
; 7512 ; TR [PI], #/077400 ;ANY PI IN PROGRESS?  
; 7513 =0  
; 7514 JEN1: [BR]#, PI.IP1/1, J/DSMS1 ;YES--START LOOP  
; 7515 RETURN [4] ;NO--JUST RETURN  
; 7516  
; 7517 DSMS1: [PI].AND.[BR], SKIP ADR.EQ.0  
; 7518 =0 [PI]_[PI].AND.NOT.[BR], HOLD LEFT, RETURN [4]  
; 7519 [BR]_[BR]*.5, J/DSMS1  
; 7520
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 209
EXTERNAL IO INSTRUCTIONS

```
; 7521 .TOC "EXTERNAL IO INSTRUCTIONS"
; 7522
; 7523 .DCODE
D 0710, 1210, 1614, 0100 ; 7524 710: IOT, WORD-TNE, J/TIOX
D 0711, 1214, 1614, 0100 ; 7525 711: IOT, WORD-TNN, J/TIOX
D 0720, 1200, 1614, 0100 ; 7526 720: IOT, TNE, J/TIOX
D 0721, 1204, 1614, 0100 ; 7527 721: IOT, TNN, J/TIOX
; 7528 .UCODE
; 7529
; 7530 1614:
U 1614, 2422, 4443, 0000, 4174, 4007, 0700, 0010, 0000, 0000 ; 7531 TIOX: CALL [IORD]
U 1617, 0014, 4551, 0305, 0274, 4003, 7700, 0000, 0000, 0000 ; 7532 1617: [BR]_[AR].AND.AC, TEST DISP
; 7533
; 7534 .DCODE
D 0712, 1210, 1460, 0100 ; 7535 712: IOT, B/10, J/RDIO
D 0713, 1210, 1461, 0100 ; 7536 713: IOT, B/10, J/WRIO
D 0722, 1200, 1460, 0100 ; 7537 722: IOT, B/O, J/RDIO
D 0723, 1200, 1461, 0100 ; 7538 723: IOT, B/O, J/WRIO
; 7539 .UCODE
; 7540
; 7541 1460:
U 1460, 2422, 4443, 0000, 4174, 4007, 0700, 0010, 0000, 0000 ; 7542 RDIO: CALL [IORD]
U 1463, 1400, 3440, 0303, 0174, 4007, 0700, 0400, 0000, 0000 ; 7543 1463: AC_[AR], J/DONE
; 7544
; 7545 1461:
U 1461, 2432, 3771, 0005, 0276, 6007, 0700, 0000, 0000, 0000 ; 7546 WRIO: [BR]_AC, J/IOWR
; 7547
; 7548 .DCODE
D 0714, 1210, 1644, 0100 ; 7549 714: IOT, B/10, J/BIXUB
D 0715, 1214, 1644, 0100 ; 7550 715: IOT, B/14, J/BIXUB
D 0724, 1200, 1644, 0100 ; 7551 724: IOT, B/O, J/BIXUB
D 0725, 1204, 1644, 0100 ; 7552 725: IOT, B/4, J/BIXUB
; 7553 .UCODE
; 7554
; 7555 1644:
U 1644, 2422, 3441, 0306, 4174, 4007, 0700, 0010, 0000, 0000 ; 7556 BIXUB: [BRX]_[AR], ;SAVE EFFECTIVE ADDRESS
; 7557 CALL [IORD] ;GO GET THE DATA
U 1647, 1013, 3441, 0305, 4174, 4003, 7700, 0000, 0000, 0000 ; 7558 1647: [BR]_[AR], ;COPY DATA ITEM
; 7559 B DISP ;SEE IF SET OR CLEAR
; 7560 =1011 [BR]_[BR].OR.AC, ;SET BITS
; 7561 J/BIXUB1 ;GO DO WRITE
; 7562 [BR]_[BR].AND.NOT.AC, ;CLEAR BITS
; 7563 J/BIXUB1 ;GO DO WRITE
; 7564
; 7565 BIXUB1: [AR]_[BRX], ;RESTORE ADDRESS
; 7566 J/IOWR
; 7567
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 210
EXTERNAL IO INSTRUCTIONS

```
; 7568 ;SUBROUTINE TO READ FROM AN IO DEVICE
; 7569 ;CALL WITH:
; 7570 ; SECTION 0 EFFECTIVE ADDRESS IN AR
; 7571 ; INSTRUCTION IN HR
; 7572 ;RETURN 3 WITH WORD OR BYTE IN AR
; 7573 ;
; 7574 =0
; 7575 IORD: CLR IO BUSY, ;CLEAR BUSY
; 7576 CALL [IOEA] ;COMPUTE IO EA
; 7577 B DISP
; 7578 =10111 [BR]_VMA IO READ, ;BYTE MODE
; 7579 IO BYTE/1, ;SET BYTE FLAG
; 7580 J/IORD1 ;GO DO C/A CYCLE
; 7581 =11111 [BR]_VMA IO READ ;WORD MODE
; 7582 =
; 7583 =0
; 7584 IORD1: VMA [AR].OR.[BR] WITH FLAGS,
; 7585 CALL [IOWAIT] ;WAIT FOR THINGS COMPLETE
; 7586 MEM READ, ;MAKE SURE REALLY READY
; 7587 [BR]_IO DATA, ;PUT DATA IN BR
; 7588 B DISP ;SEE IF BYTE MODE
; 7589 =0111 TR [AR], #/1, J/IORD2 ;BYTE MODE SEE IF ODD
; 7590 [AR]_[BR], RETURN [3] ;ALL DONE
; 7591
; 7592 ;HERE ON WORD MODE
; 7593 =0
; 7594 IORD2: [BR]_[BR]*.5, SC_5, ;LEFT BYTE
; 7595 J/IORD3 ;GO SHIFT IT
; 7596 [AR]_[BR].AND.#, ;MASK IT
; 7597 #/377, RETURN [3] ;ALL DONE
; 7598
; 7599 =0
; 7600 IORD3: [BR]_[BR]*.5, ;SHIFT OVER
; 7601 STEP SC, J/IORD3 ;..
; 7602 [AR]_[BR].AND.#, ;MASK IT
; 7603 #/377, RETURN [3] ;ALL DONE
; 7604
```

U 2422, 2442,4443,0000,4174,4137,0700,0010,0000,0000
U 2423, 0067,4443,0000,4174,4003,7700,0000,0000,0000

U 0067, 2424,4571,1205,4374,4007,0700,0000,0024,1220
U 0077, 2424,4571,1205,4374,4007,0700,0000,0024,1200

U 2424, 3560,3113,0305,4174,4007,0701,0210,0000,0036

U 2425, 1027,3771,0005,4364,4003,7700,0200,0000,0002
U 1027, 2426,4553,0300,4374,4007,0331,0000,0000,0001
U 1037, 0003,3441,0503,4174,4004,1700,0000,0000,0000

U 2426, 2430,3447,0505,4174,4007,0700,2000,0071,0005

U 2427, 0003,4551,0503,4374,4004,1700,0000,0000,0377

U 2430, 2430,3447,0505,4174,4007,0630,2000,0060,0000

U 2431, 0003,4551,0503,4374,4004,1700,0000,0000,0377

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 211
EXTERNAL IO INSTRUCTIONS

```
; 7605 ;ROUTINE TO WRITE TO AN IO DEVICE
; 7606 ;CALL WITH:
; 7607 ; SECTION 0 EFFECTIVE ADDRESS IN AR
; 7608 ; INSTRUCTION IN HR
; 7609 ; WORD OR BYTE IN BR
; 7610 ;RETURNS BACK TO USER
; 7611 ;
; 7612 =0
; 7613 IOWR: CLR IO BUSY, ;CLEAR BUSY
; 7614 CALL [IOEA] ;COMPUTE IO EA
; 7615 B DISP
; 7616 =10111 TR [AR], #/1, J/IOWR2 ;BYTE MODE
; 7617 =11111 [ARX]_VMA IO WRITE ;SETUP FLAGS
; 7618 =
; 7619 IOWR1: VMA [AR].OR.[ARX] WITH FLAGS
; 7620 =0 MEM WRITE, MEM_[BR], ;SEND DATA
; 7621 CALL [IOWAIT] ;WAIT FOR DATA
; 7622 DONE ;RETURN
; 7623
; 7624 ;HERE FOR BYTE MODE
; 7625 =0
; 7626 IOWR2: [BR]_[BR]*2, SC_5, ;ODD--MOVE LEFT
; 7627 J/IOWR3 ;..
; 7628 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7629 IO BYTE/1, J/IOWR1 ;..
; 7630
; 7631 =0
; 7632 IOWR3: [BR]_[BR]*2, STEP SC, ;SHIFT LEFT
; 7633 J/IOWR3 ;KEEP SHIFTING
; 7634 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7635 IO BYTE/1, J/IOWR1 ;..
; 7636
```

U 2432, 2442,4443,0000,4174,4137,0700,0010,0000,0000
U 2433, 0227,4443,0000,4174,4003,7700,0000,0000,0000
U 0227, 2436,4553,0300,4374,4007,0331,0000,0000,0001
U 0237, 3550,4571,1204,4374,4007,0700,0000,0021,1200

U 3550, 2434,3113,0304,4174,4007,0701,0200,0000,0036

U 2434, 3560,3333,0005,4175,5007,0701,0210,0000,0002
U 2435, 0110,3443,0100,4174,4156,4700,0200,0014,0012

U 2436, 2440,3445,0505,4174,4007,0700,2000,0071,0005

U 2437, 3550,4571,1204,4374,4007,0700,0000,0021,1220

U 2440, 2440,3445,0505,4174,4007,0630,2000,0060,0000

U 2441, 3550,4571,1204,4374,4007,0700,0000,0021,1220

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 212
EXTERNAL IO INSTRUCTIONS

```
; 7637 ;HERE TO COMPUTE IO EFFECTIVE ADDRESS
; 7638 ;CALL WITH:
; 7639 ; SECTION 0 EFFECTIVE ADDRESS IN AR
; 7640 ; INSTRUCTION IN HR
; 7641 ;RETURN 1 WITH EA IN AR
; 7642 ;
; 7643 =0
; 7644 IOEA: VMA [PC]-1, ;GET INSTRUCTION
; 7645 START READ, ;..
; 7646 CALL [LOADAR] ;PUT WORD IN AR
; 7647 [BRX].NOT.[AR] ;SEE IF IN RANGE 700-777
; 7648 TL [BRX], #/700000 ;..
; 7649 =0
; 7650 IOEA1: TL [HR], #/20, J/IOEA2 ;INDIRECT?
; 7651 WORK[YSAVE]_[AR] CLR LH, ;DIRECT IO INSTRUCTION
; 7652 J/IOEA1 ;SAVE Y FOR EA CALCULATION
; 7653 =0
; 7654 IOEA2: [AR] WORK[YSAVE], ;@--GET SAVED Y
; 7655 J/IOEA1 ;GET Y AND GO
; 7656 EA MODE DISP ;WAS THERE INDEXING?
; 7657 =1101 [ARX] XR, SKIP ADL.LE.O, ;SEE IF LOCAL OR GLOBAL INDEXING
; 7658 2T, J/IOEAX ;..
; 7659 [AR] WORK[YSAVE], ;JUST PLAIN IO
; 7660 CLR IO LATCH, RETURN [1]
; 7661
; 7662 IOEA1: READ [HR], DBUS/DP, ;LOAD XR FLOPS IN CASE
; 7663 LOAD INST EA ; THERE IS INDEXING
; 7664 TL [HR], #/17 ;WAS THERE ALSO INDEXING
; 7665 =0 [AR]_[AR]+XR, 3T, HOLD LEFT ;YES--ADD IN INDEX VALUE
; 7666 VMA [AR], START READ ;FETCH DATA WORD
; 7667 MEM READ, [AR] MEM, ;GO GET DATA WORD
; 7668 CLR IO LATCH, RETURN [1]
; 7669
; 7670 =0
; 7671 IOEAX: [AR]_[ARX]+WORK[YSAVE], ;GLOBAL INDEXING
; 7672 CLR IO LATCH, RETURN [1]
; 7673 [AR]_[ARX]+WORK[YSAVE] ;LOCAL INDEXING
; 7674 [AR]_O, HOLD RIGHT,
; 7675 CLR IO LATCH, RETURN [1]
; 7676

U 2442, 3562,1113,0701,4170,4007,0700,4210,0004,0012
U 2443, 3551,7441,0306,4174,4007,0700,0000,0000,0000
U 3551, 2444,4553,0600,4374,4007,0321,0000,0070,0000

U 2444, 2446,4553,0200,4374,4007,0321,0000,0000,0020

U 2445, 2444,4713,1203,7174,4007,0700,0400,0000,0422

U 2446, 3552,3771,0003,7274,4007,0701,0000,0000,0422
U 2447, 1055,4443,0000,2174,4006,6700,0000,0000,0000

U 1055, 2452,3771,0004,2274,4007,0120,0000,0000,0000

U 1057, 0001,3771,0003,7274,4124,1701,0000,0000,0422

U 3552, 3553,3333,0002,4174,4217,0700,0000,0000,0000
U 3553, 2450,4553,0200,4374,4007,0321,0000,0000,0017
U 2450, 2451,0551,0303,2270,4007,0701,0000,0000,0000
U 2451, 3554,3443,0300,4174,4007,0700,0200,0004,0012

U 3554, 0001,3771,0003,4365,5124,1700,0200,0000,0002

U 2452, 0001,0551,0403,7274,4124,1701,0000,0000,0422
U 2453, 3555,0551,0403,7274,4007,0701,0000,0000,0422

U 3555, 0001,4221,0003,4174,0124,1700,0000,0000,0000
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 213
EXTERNAL IO INSTRUCTIONS

```

; 7677 ;WAIT FOR IO TO COMPLETE
; 7678 ;RETURNS 1 OR PAGE FAILS
; 7679 ;
; 7680 IOWAIT: SC S#, S#/200, ;DELAY
; 7681 [TO]_VMA, ;GET VMA
; 7682 SKIP/-IO BUSY ;SEE IF BUSY YET
; 7683 =00
; 7684 IOW1: CLR IO LATCH, ;WENT BUSY
; 7685 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7686 J/IOW2 ;WAIT FOR IT TO CLEAR
; 7687 SC_SC-1, SCAD DISP, 5T, ;SEE IF DONE YET
; 7688 SKIP/-IO BUSY, ;...
; 7689 J/IOW1 ;BACK TO LOOP
; 7690 CLR IO LATCH, ;WENT BUSY AND TIMEOUT
; 7691 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7692 J/IOW2 ;...
; 7693 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7694 J/IOW5 ;GO TRAP
; 7695
; 7696 IOW2: SC S#, S#/777, ;GO TIME IO
; 7697 SKIP/-IO BUSY ;...
; 7698 =0
; 7699 IOW3: CLR IO LATCH, ;TRY TO CLEAR LATCH
; 7700 STEP SC, J/IOW4 ;STILL BUSY
; 7701 RETURN [1] ;IDLE
; 7702
; 7703 =0
; 7704 IOW4: CLR IO LATCH, 5T, ;TRY TO CLEAR LATCH
; 7705 SKIP/-IO BUSY, ;SEE IF STILL BUSY
; 7706 J/IOW3 ;...
; 7707 IOW5: [BRX]_[200000] XWD 0, J/HARD
; 7708
```

U 3560, 1074,3771,0016,4354,4007,0650,2000,0071,0200

U 1074, 3561,3333,0016,7174,4127,0700,0400,0000,0210

U 1075, 1074,4443,0000,4174,4006,7653,2000,0060,0000

U 1076, 3561,3333,0016,7174,4127,0700,0400,0000,0210

U 1077, 2457,3333,0016,7174,4007,0700,0400,0000,0210

U 3561, 2454,4443,0000,4174,4007,0650,2000,0071,0777

U 2454, 2456,4443,0000,4174,4127,0630,2000,0060,0000

U 2455, 0001,4443,0000,4174,4004,1700,0000,0000,0000

U 2456, 2454,4443,0000,4174,4127,0653,0000,0000,0000

U 2457, 3664,4571,1206,4374,4007,0700,0000,0020,0000

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 214
SMALL SUBROUTINES

```
U 3562, 0001,3771,0003,4365,5004,1700,0200,0000,0002
U 3563, 0001,3771,0004,4365,5004,1700,0200,0000,0002
U 3564, 0001,3772,0000,4365,5004,1700,0200,0000,0002
U 3565, 0001,4221,0004,4174,0004,1700,0000,0000,0000
U 3566, 0001,0111,0703,4174,4004,1700,0000,0000,0000
U 3567, 0001,3445,0505,4174,4004,1700,0000,0000,0000
U 3570, 0001,3443,0300,4174,4004,1701,0200,0000,0036
U 3571, 0004,3333,0005,4175,5004,1701,0200,0000,0002
U 3572, 0001,3333,0001,4175,5004,1701,0200,0000,0002
U 3573, 0001,3440,0404,0174,4004,1700,0400,0000,0000

; 7709 .TOC "SMALL SUBROUTINES"
; 7710
; 7711 ;HERE ARE A COLLECTION ON 1-LINE SUBROUTINES
; 7712 LOADAR: MEM READ, [AR]_MEM, ;FROM MEMORY TO AR
; 7713 RETURN [1] ;RETURN TO CALLER
; 7714
; 7715 LOADARX: MEM READ, [ARX]_MEM, RETURN [1]
; 7716
; 7717 LOADQ: MEM READ, Q_MEM, RETURN [1]
; 7718
; 7719 CLARXL: [ARX]_O, HOLD RIGHT, RETURN [1]
; 7720
; 7721 INCAR: [AR]_[AR]+1, RETURN [1]
; 7722
; 7723 SBRL: [BR]_[BR]*2, RETURN [1]
; 7724
; 7725 STRTIO: VMA_[AR] WITH FLAGS, RETURN [1]
; 7726
; 7727 STOBR: MEM WRITE, MEM_[BR], RETURN [4]
; 7728
; 7729 STOPC: MEM WRITE, MEM_[PC], RETURN [1]
; 7730
; 7731 AC_ARX: AC_[ARX], RETURN [1]
; 7732
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 215
UNDEFINED IO INSTRUCTIONS

```

; 7733 .TOC "UNDEFINED IO INSTRUCTIONS"
; 7734
; 7735 .DCODE
D 0703, 0003, 1650, 2100 ; 7736 703: I, B/3, J/IOT700
D 0706, 0006, 1650, 2100 ; 7737 706: I, B/6, J/IOT700
D 0707, 0007, 1650, 2100 ; 7738 I, B/7, J/IOT700
; 7739
D 0716, 0006, 1651, 2100 ; 7740 716: I, B/6, J/IOT710
D 0717, 0007, 1651, 2100 ; 7741 I, B/7, J/IOT710
; 7742
D 0726, 0006, 1652, 2100 ; 7743 726: I, B/6, J/IOT720
D 0727, 0007, 1652, 2100 ; 7744 I, B/7, J/IOT720
; 7745
D 0730, 0000, 1653, 2100 ; 7746 730: I, B/0, J/IOT730
D 0731, 0001, 1653, 2100 ; 7747 I, B/1, J/IOT730
D 0732, 0002, 1653, 2100 ; 7748 I, B/2, J/IOT730
D 0733, 0003, 1653, 2100 ; 7749 I, B/3, J/IOT730
D 0734, 0004, 1653, 2100 ; 7750 I, B/4, J/IOT730
D 0735, 0005, 1653, 2100 ; 7751 I, B/5, J/IOT730
D 0736, 0006, 1653, 2100 ; 7752 I, B/6, J/IOT730
D 0737, 0007, 1653, 2100 ; 7753 I, B/7, J/IOT730
; 7754
D 0740, 0000, 1654, 2100 ; 7755 740: I, B/0, J/IOT740
D 0741, 0001, 1654, 2100 ; 7756 I, B/1, J/IOT740
D 0742, 0002, 1654, 2100 ; 7757 I, B/2, J/IOT740
D 0743, 0003, 1654, 2100 ; 7758 I, B/3, J/IOT740
D 0744, 0004, 1654, 2100 ; 7759 I, B/4, J/IOT740
D 0745, 0005, 1654, 2100 ; 7760 I, B/5, J/IOT740
D 0746, 0006, 1654, 2100 ; 7761 I, B/6, J/IOT740
D 0747, 0007, 1654, 2100 ; 7762 I, B/7, J/IOT740
; 7763
D 0750, 0000, 1655, 2100 ; 7764 750: I, B/0, J/IOT750
D 0751, 0001, 1655, 2100 ; 7765 I, B/1, J/IOT750
D 0752, 0002, 1655, 2100 ; 7766 I, B/2, J/IOT750
D 0753, 0003, 1655, 2100 ; 7767 I, B/3, J/IOT750
D 0754, 0004, 1655, 2100 ; 7768 I, B/4, J/IOT750
D 0755, 0005, 1655, 2100 ; 7769 I, B/5, J/IOT750
D 0756, 0006, 1655, 2100 ; 7770 I, B/6, J/IOT750
D 0757, 0007, 1655, 2100 ; 7771 I, B/7, J/IOT750
; 7772
D 0760, 0000, 1656, 2100 ; 7773 760: I, B/0, J/IOT760
D 0761, 0001, 1656, 2100 ; 7774 I, B/1, J/IOT760
D 0762, 0002, 1656, 2100 ; 7775 I, B/2, J/IOT760
D 0763, 0003, 1656, 2100 ; 7776 I, B/3, J/IOT760
D 0764, 0004, 1656, 2100 ; 7777 I, B/4, J/IOT760
D 0765, 0005, 1656, 2100 ; 7778 I, B/5, J/IOT760
D 0766, 0006, 1656, 2100 ; 7779 I, B/6, J/IOT760
D 0767, 0007, 1656, 2100 ; 7780 I, B/7, J/IOT760
; 7781
```


; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 216
UNDEFINED IO INSTRUCTIONS

D 0770, 0000,1657,2100
D 0771, 0001,1657,2100
D 0772, 0002,1657,2100
D 0773, 0003,1657,2100
D 0774, 0004,1657,2100
D 0775, 0005,1657,2100
D 0776, 0006,1657,2100
D 0777, 0007,1657,2100

; 7782 770: I, B/0, J/IOT770
; 7783 I, B/1, J/IOT770
; 7784 I, B/2, J/IOT770
; 7785 I, B/3, J/IOT770
; 7786 I, B/4, J/IOT770
; 7787 I, B/5, J/IOT770
; 7788 I, B/6, J/IOT770
; 7789 I, B/7, J/IOT770
; 7790 .UCODE
; 7791

U 1650, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7792 1650:
; 7793 IOT700: UUO
; 7794 1651:
; 7795 IOT710:
; 7796 .IFNOT/UBABLT
; 7797 UUO
; 7798 .IF/UBABLT
; 7799 J/BLTX

U 1651, 0670,4443,0000,4174,4007,0700,0000,0000,0000

; 7800 .ENDIF/UBABLT

; GO TO COMMON CODE FOR UBABLT INSTRS

U 1652, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7801 1652:
; 7802 IOT720: UUO

U 1653, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7803 1653:
; 7804 IOT730: UUO

U 1654, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7805 1654:
; 7806 IOT740: UUO

U 1655, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7807 1655:
; 7808 IOT750: UUO

U 1656, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7809 1656:
; 7810 IOT760: UUO

U 1657, 2621,4551,0202,4374,0007,0700,0000,0077,7740

; 7811 1657:
; 7812 IOT770: UUO

; 7813

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 217
UMOVE AND UMOVEM

D 0704, 1200,1754,0100
D 0705, 1200,1755,0100

U 1754, 3574,3443,0300,4174,4207,0700,0200,0004,0012

U 3574, 1515,3771,0003,4365,5007,0700,0200,0000,0002

U 1755, 3575,3443,0300,4174,4207,0700,0200,0003,0012

U 3575, 1516,3771,0003,0276,6007,0700,0000,0000,0000

```
; 7814 .TOC "UMOVE AND UMOVEM"
; 7815
; 7816 .DCODE
; 7817 704: IOT, J/UMOVE
; 7818 IOT, J/UMOVEM
; 7819 .UCODE
; 7820
; 7821 1754:
; 7822 UMOVE: VMA [AR], ;LOAD VMA
; 7823 START READ, ;START MEMORY
; 7824 SPEC/PREV ;FORCE PREVIOUS
; 7825 MEM READ, ;WAIT FOR MEMORY
; 7826 [AR]_MEM, ;PUT DATA IN AR
; 7827 J/STAC ;GO PUT AR IN AC
; 7828
; 7829 1755:
; 7830 UMOVEM: VMA [AR], ;LOAD VMA
; 7831 START WRITE, ;START MEMORY
; 7832 SPEC/PREV ;FORCE PREVIOUS
; 7833 [AR]_AC, ;FETCH AC
; 7834 J/STMEM ;STORE IN MEMORY
; 7835
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 218
UMOVE AND UMOVEM

U 0104, 3600,3333,0004,7174,4007,0700,0410,0000,0212
U 0114, 3577,4223,0000,4364,4277,0700,0210,0000,0010
U 0116, 3576,4221,0004,4174,4007,0700,0200,0021,1016
U 3576, 2460,3333,0017,4175,5007,0701,0200,0000,0002
U 2460, 3572,0111,0704,4170,4007,0700,0210,0023,1016
U 2461, 0005,4443,0000,4174,4107,0700,0000,0000,0074

U 0004, 2462,4443,0000,4174,4107,0640,0000,0000,0062
U 0005, 0004,4443,0000,4174,4007,0660,0000,0000,0000

U 2462, 0117,3443,0100,4174,4007,0700,0200,0014,0012
U 2463, 2464,4571,1203,4374,4007,0700,0000,0024,1200

U 2464, 3570,3551,0303,4370,4007,0700,0010,0020,0000

U 2465, 2600,3771,0002,4365,5617,0700,0200,0000,0002

```
; 7836 ;HERE WITH HALT CODE IN THE T1
; 7837 =010*
; 7838 HALTED: WORK[SV.ARX]_[ARX], ;SAVE TEMP REGISTER
; 7839 CALL [SAVVMA] ;PUT VMA IN WORK[SV.VMA]
; 7840 =110* ABORT MEM CYCLE, ;ABORT CYCLE IN PROGRESS
; 7841 CALL [WRTHSB] ;WRITE HALT STATUS BLOCK
; 7842 =111*
; 7843 PWRON: [ARX]_O, VMA PHYSICAL WRITE ;STORE HALT CODE
; 7844 =
; 7845 MEM WRITE, MEM [T1] ; IN LOCATION O
; 7846 =0 NEXT [ARX] PHYSICAL WRITE,
; 7847 CALL [STOPC]
; 7848 H1: SET HALT, J/HALTLP ;TELL CONSOLE WE HAVE HALTED
; 7849
; 7850
; 7851 4: UNHALT, ;RESET CONSOLE
; 7852 SKIP EXECUTE, J/CONT ;SEE IF CO OR EX
; 7853 5:
; 7854 HALTLP: SKIP/-CONTINUE, J/4 ;WAIT FOR CONTINUE
; 7855
; 7856 =0
; 7857 CONT: VMA [PC], ;LOAD PC INTO VMA
; 7858 FETCH, ;START READ
; 7859 J/XCTGO ;DO THE INSTRUCTION
; 7860 [AR]_VMA IO READ ;PUT FLAGS IN AR
; 7861 =0 [AR]_[AR].OR.#, ;PUT IN ADDRESS
; 7862 #/200000, HOLD LEFT, ; OF CSL REGISTER
; 7863 CALL [STRTIO]
; 7864 CONT1: MEM READ, ;WAIT FOR DATA
; 7865 [HR]_MEM, ;PUT IN HR
; 7866 LOAD INST, ;LOAD IR, ETC.
; 7867 J/XCT1 ;GO DO THE INSTRUCTION
; 7868
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 219
WRITE HALT STATUS BLOCK

```
; 7869 .TOC "WRITE HALT STATUS BLOCK"  
; 7870  
; 7871 ;THE HALT STATUS BLOCK LOOKS LIKE:  
; 7872  
; 7873 ; !-----!  
; 7874 ; !00! MAG !  
; 7875 ; !-----!  
; 7876 ; !01! PC !  
; 7877 ; !-----!  
; 7878 ; !02! HR !  
; 7879 ; !-----!  
; 7880 ; !03! AR !  
; 7881 ; !-----!  
; 7882 ; !04! ARX !  
; 7883 ; !-----!  
; 7884 ; !05! BR !  
; 7885 ; !-----!  
; 7886 ; !06! BRX !  
; 7887 ; !-----!  
; 7888 ; !07! ONE !  
; 7889 ; !-----!  
; 7890 ; !10! EBR !  
; 7891 ; !-----!  
; 7892 ; !11! UBR !  
; 7893 ; !-----!  
; 7894 ; !12! MASK !  
; 7895 ; !-----!  
; 7896 ; !13! FLG !  
; 7897 ; !-----!  
; 7898 ; !14! PI !  
; 7899 ; !-----!  
; 7900 ; !15! XWD1 !  
; 7901 ; !-----!  
; 7902 ; !16! TO !  
; 7903 ; !-----!  
; 7904 ; !17! T1 !  
; 7905 ; !-----!  
; 7906 ; ! VMA FLAGS ! VMA !  
; 7907 ; !-----!  
; 7908
```

; T10KI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 220
WRITE HALT STATUS BLOCK

U 0001, 3600,3333,0004,7174,4007,0700,0410,0000,0212
U 0011, 0224,3771,0004,7274,4007,0701,0000,0000,0227
U 0224, 3602,4223,0000,4364,4277,0700,0210,0000,0010
U 0226, 2461,4443,0000,4174,4107,0700,0000,0000,0074

U 3577, 2466,3771,0004,7274,4007,0422,0000,0000,0227

U 2466, 3602,3333,0012,4174,4437,0700,0000,0000,0000

U 2467, 0002,3771,0004,7274,4004,1701,0000,0000,0212

U 3600, 3601,3771,0004,4354,4007,0700,0000,0000,0000

U 3601, 0010,3333,0004,7174,4004,1700,0400,0000,0210

```
; 7909 ;START AT 1 TO DUMP 2901 REGISTERS INTO MAIN MEMORY
; 7910 1:   WORK[SV.ARX]_[ARX],   ;SAVE TEMP REGISTER
; 7911   CALL [SAVVMA]           ;WORK[SV.VMA]_VMA
; 7912 11:  [ARX]_WORK[HSBADR]
; 7913 =10* ABORT MEM CYCLE, CALL [DUMP]
; 7914   SET HALT, J/H1
; 7915
; 7916
; 7917 WRTHSB: [ARX]_WORK[HSBADR], ;GET ADDRESS OF HSB
; 7918   SKIP AD.LE.O, 4T         ;SEE IF VALID
; 7919 =0   READ [MASK], LOAD PI, ;TURN OFF PI SYSTEM
; 7920   J/DUMP                   ; AND GO TAKE DUMP
; 7921   [ARX]_WORK[SV.ARX],
; 7922   RETURN [2]              ;DO NOT DUMP ANYTHING
; 7923
; 7924 SAVVMA: [ARX]_VMA
; 7925   WORK[SV.VMA]_[ARX],
; 7926   RETURN [10]
; 7927
```

; T1OKI.MCR[10,1141]
; INOUT.MIC[10,1141]

11:45 11-AUG-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 221
WRITE HALT STATUS BLOCK

```
U 3602, 2341,3333,0004,4174,4007,0700,0200,0021,1016 ; 7928 ;DUMP OUT THE 2901
U 2341, 2630,3333,0000,4175,5007,0701,0210,0000,0002 ; 7929 DUMP: READ [ARX], VMA PHYSICAL WRITE
U 2343, 3603,3333,0001,4175,5007,0701,0200,0000,0002 ; 7930 =0* MEM WRITE, MEM_[MAG], CALL [NEXT]
U 3603, 2470,0111,0704,4170,4007,0700,0200,0023,1016 ; 7931 MEM WRITE, MEM_[PC]
U 2470, 2630,3333,0002,4175,5007,0701,0210,0000,0002 ; 7932 NEXT [ARX] PHYSICAL WRITE
U 2472, 2471,3333,0003,4175,5007,0701,0200,0000,0002 ; 7933 =0* MEM WRITE, MEM_[HR], CALL [NEXT]
U 2471, 2630,3333,0003,7174,4007,0700,0410,0000,0211 ; 7934 MEM WRITE, MEM_[AR]
U 2473, 2474,3771,0003,7274,4007,0701,0000,0000,0212 ; 7935 =0* WORK[SV.AR]_[AR], CALL [NEXT]
U 2474, 2630,3333,0003,4175,5007,0701,0210,0000,0002 ; 7936 [AR] WORK[SV.ARX]
U 2476, 3604,3333,0005,4175,5007,0701,0200,0000,0002 ; 7937 =0* MEM WRITE, MEM_[AR], CALL [NEXT]
U 3604, 2475,0111,0704,4170,4007,0700,0200,0023,1016 ; 7938 MEM WRITE, MEM_[BR]
U 2475, 2630,3333,0006,4175,5007,0701,0210,0000,0002 ; 7939 NEXT [ARX] PHYSICAL WRITE
U 2477, 3605,3333,0007,4175,5007,0701,0200,0000,0002 ; 7940 =0* MEM WRITE, MEM_[BRX], CALL [NEXT]
U 3605, 2500,0111,0704,4170,4007,0700,0200,0023,1016 ; 7941 MEM WRITE, MEM_[ONE]
U 2500, 2630,3333,0010,4175,5007,0701,0210,0000,0002 ; 7942 NEXT [ARX] PHYSICAL WRITE
U 2502, 3606,3333,0011,4175,5007,0701,0200,0000,0002 ; 7943 =0* MEM WRITE, MEM_[EBR], CALL [NEXT]
U 3606, 2501,0111,0704,4170,4007,0700,0200,0023,1016 ; 7944 MEM WRITE, MEM_[UBR]
U 2501, 2630,3333,0012,4175,5007,0701,0210,0000,0002 ; 7945 NEXT [ARX] PHYSICAL WRITE
U 2503, 3607,3333,0013,4175,5007,0701,0200,0000,0002 ; 7946 =0* MEM WRITE, MEM_[MASK], CALL [NEXT]
U 3607, 2504,0111,0704,4170,4007,0700,0200,0023,1016 ; 7947 MEM WRITE, MEM_[FLG]
U 2504, 2630,3333,0014,4175,5007,0701,0210,0000,0002 ; 7948 NEXT [ARX] PHYSICAL WRITE
U 2506, 3610,3333,0015,4175,5007,0701,0200,0000,0002 ; 7949 =0* MEM WRITE, MEM_[PI], CALL [NEXT]
U 3610, 2505,0111,0704,4170,4007,0700,0200,0023,1016 ; 7950 MEM WRITE, MEM_[XWD1]
U 2505, 2630,3333,0016,4175,5007,0701,0210,0000,0002 ; 7951 NEXT [ARX] PHYSICAL WRITE
U 2507, 2510,3333,0017,4175,5007,0701,0200,0000,0002 ; 7952 =0* MEM WRITE, MEM_[TO], CALL [NEXT]
U 2510, 2630,3771,0003,7274,4007,0701,0200,0000,0002 ; 7953 MEM WRITE, MEM_[T1]
U 2512, 3611,3333,0003,4175,5007,0701,0200,0000,0002 ; 7954 =0* [AR] WORK[SV.VMA], CALL [NEXT]
U 3611, 3612,3771,0003,7274,4007,0701,0000,0000,0211 ; 7955 MEM WRITE, MEM_[AR]
U 3612, 3613,3771,0004,7274,4007,0701,0000,0000,0210 ; 7956 HSBDON: [AR] WORK[SV.AR]
U 3613, 3614,3443,0400,4174,4007,0700,0200,0000,0010 ; 7957 [ARX] WORK[SV.VMA]
; 7958 VMA [ARX]
; 7959 [ARX] WORK[SV.ARX],
; 7960 RETURN [6]
; 7961
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 222
WRITE HALT STATUS BLOCK

```
; 7962      .NOBIN
; 7963      .TOC      "PAGE FAIL REFIL LOGIC"
; 7964
; 7965      ;WHEN THE CPU CAN NOT COMPLETE A MEMORY REFERENCE BECAUSE THE PAGE
; 7966      ; TABLE DOES NOT CONTAIN VALID INFORMATION FOR THE VIRTUAL PAGE INVOLVED
; 7967      ; THE HARDWARE CALLS THIS ROUTINE TO RELOAD THE HARDWARE PAGE TABLE.
; 7968      ;
; 7969      ;THIS CODE WILL EITHER DO THE RELOAD OR GENERATE A PAGE FAIL FOR THE
; 7970      ; SOFTWARE. THE INFORMATION LOADED CONSISTS OF THE PHYSICAL PAGE NUMBER,
; 7971      ; THE CACHE ENABLE BIT AND THE WRITE ENABLE BIT.
; 7972
; 7973      ;THIS LOGIC USES MANY VARIABLES. THEY ARE DESCRIBED BRIEFLY HERE:
; 7974
; 7975      ;THING          WHERE KEPT          USE
; 7976      ;OLD VMA          WORKSPACE WORD 210      SAVES VMA
; 7977      ;OLD AR          WORKSPACE WORD 211      SAVES AR
; 7978      ;OLD ARX        WORKSPACE WORD 212      SAVES ARX
; 7979      ;OLD BR        WORKSPACE WORD 213      SAVES BR
; 7980      ;OLD BRX        WORKSPACE WORD 214      SAVES BRX
; 7981      ;KL PAGING BIT  EBR BIT 1 (IN 2901)      INDICATES KL STYLE (TOPS-20) PAGING
; 7982      ;              ;              ;          INSTEAD OF KI STYLE (TOPS-10 AND DIAGNOSTIC)
; 7983      ;              ;              ;          MODE PAGING
; 7984      ;W BIT          FLG BIT 4              PAGE CAN BE WRITTEN
; 7985      ;C BIT          FLG BIT 6              DATA IN THIS PAGE MAY BE PUT
; 7986      ;              ;              ;          INTO CACHE
; 7987      ;PI CYCLE        FLG BIT 5              STORING OLD PC DURING PI
; 7988      ;MAP FLAG        FLG BIT 18             MAP INSTRUCTION IN PROGRESS
; 7989      ;CLEANUP CODE    FLG BITS 32-35         WHAT TO DO SO INSTRUCTION MAY BE
; 7990      ;              ;              ;          RESTARTED
; 7991      ;SPT BASE          WORKSPACE WORD 215     ADDRESS OF SHARED-POINTER-TABLE
; 7992      ;CST BASE          WORKSPACE WORD 216     ADDRESS OF CORE-STATUS-TABLE
; 7993      ;CST MASK          WORKSPACE WORD 217     BITS TO KEEP ON CST UPDATE
; 7994      ;CST DATA (PUR)  WORKSPACE WORD 220     BITS TO SET ON CST UPDATE
; 7995      ;PAGE TABLE ADDRESS AR                  WHERE THIS PAGE TABLE IS LOCATED
; 7996      ;PHYSICAL PAGE # (PPN) AR                RESULT OF THIS PROCESS
; 7997      ;CST ENTRY          AR                  CORE STATUS TABLE ENTRY
; 7998      ;SPT ENTRY          AR                  WORD FROM SPT
; 7999      ;PAGE TABLE ENTRY AR                  WORD FROM PT
; 8000      ;PAGE NUMBER        BR                  INDEX INTO CURENT PAGE TABLE
; 8001      ;PAGE FAIL WORD    BR                  WHAT HAPPENED (ALSO MAP RESULT)
; 8002
; 8003      .IF/INHCST
; 8004          SKIP NO CST      "AD/D,DBUS/RAM,RAMADR/#,WORK/CBR,DT/4T,SKIP/ADEQO"
; 8005      .ENDIF/INHCST
; 8006
```

```

; 8007 ;
; 8008 ;
; 8009 ;
; 8010 ;      * KL10 PAGING - WORD FORMATS
; 8011 ;
; 8012 ;Section Pointer
; 8013 ;
; 8014 ;The section pointer is found in the user or exec section table.
; 8015 ;(Part of UPT or EPT.)
; 8016 ;
; 8017 ;Section pointer provides (via the SPT) the physical address of
; 8018 ;the PAGE TABLE for the given section.
; 8019 ;

```

```

; 8020 ;      Code:  0      No-access (trap)
; 8021 ;           1      Immediate
; 8022 ;           2      Share
; 8023 ;           3      Indirect
; 8024 ;           4-7    Unused, reserved
; 8025 ;

```

```

; 8026 ;      0 1 2 3 4 5 6          18          35
; 8027 ;      +-----+-----+-----+
; 8028 ;      !CODE!PIW! !C!//////////! PAGE TABLE IDENTIFIER !
; 8029 ;      !O10 ! ! ! ! !//////////! (SPT INDEX) !
; 8030 ;      +-----+-----+-----+

```

```

; 8031 ;
; 8032 ;      NORMAL SECTION POINTER (Code = 2)
; 8033 ;
; 8034 ;      0  2 3 4 5 6          9          18          35
; 8035 ;      +-----+-----+-----+
; 8036 ;      !CODE!PIW! !C!///!SECTION !SECTION TABLE IDENTIFIER!
; 8037 ;      !O11 ! ! ! ! !///!TABLE INDEX! (SPT INDEX) !
; 8038 ;      +-----+-----+-----+

```

```

; 8039 ;
; 8040 ;      INDIRECT SECTION POINTER (Code = 3)
; 8041 ;

```



```

; 8075 ;      0 1 2 3      6      9      17 18      35
; 8076 ;      +-----+-----+-----+-----+
; 8077 ;      !CODE!SAME AS !///

```

```

; 8080 ;
; 8081 ;      INDIRECT POINTER (code field = 3)
; 8082 ;
; 8083 ;      This pointer type causes another pointer to be fetched
; 8084 ;      and interpreted. The new pointer is found in word N
; 8085 ;      (B9-17) of the page addressed by C(SPT + SPTX).
; 8086 ;
; 8087 ;
; 8088 ;

```

```

; 8089 ;      SPT ENTRY
; 8090 ;
; 8091 ;      Found in the SPT, i.e., when fetching C(SPT +SPTX)
; 8092 ;
; 8093 ;

```

```

; 8094 ;      12      35
; 8095 ;      +-----+-----+
; 8096 ;      !//////////! PHYSICAL ADDRESS OF PAGE !
; 8097 ;      !//////////! OR PAGE TABLE      !
; 8098 ;      +-----+-----+

```

```

; 8099 ;      B12-35 Give PHYSICAL ADDRESS of page.
; 8100 ;
; 8101 ;      The base address (physical core address) of the SPT
; 8102 ;      resides in one AC of the reserved AC block.
; 8103 ;

```

```
; 8104 ;PHYSICAL STORAGE ADDRESS
; 8105 ;
; 8106 ;Found in B12-35 of IMMEDIATE POINTERS and SPT ENTRIES.
; 8107 ;
; 8108 ;           12      17 18   23              35
; 8109 ;           +-----+-----+-----+
; 8110 ;           !           !MBZ ! CORE PAGE NUMBER!
; 8111 ;           !           !   !   IF B12-17 = 0 !
; 8112 ;           +-----+-----+-----+
; 8113 ;
; 8114 ;           If B12-17 = 0, then B23-35 are CORE PAGE NUMBER (i.e.,
; 8115 ;           B14-26 of physical core address) of page and B18-22
; 8116 ;           MBZ. If B12-17 > 0, then address is not core and
; 8117 ;           pager traps.
; 8118 ;
; 8119 ;
; 8120 ;
; 8121 ;CORE STATUS TABLE ENTRY
; 8122 ;
; 8123 ;Found when fetching C(CBR + CORE PAGENO)
; 8124 ;
; 8125 ;           0      5              32  34  35
; 8126 ;           +-----+-----+-----+
; 8127 ;           !  CODE !              !      !M!
; 8128 ;           +-----+-----+-----+
; 8129 ;
; 8130 ;           B0-5   are code field:
; 8131 ;
; 8132 ;           0 - unavailable, trap
; 8133 ;
; 8134 ;           1-77 - available
; 8135 ;
; 8136 ;
; 8137 ;
; 8138 ;           B32-34 reserved for future hardware specification.
; 8139 ;
; 8140 ;           B35 is "modified" bit, set on any write ref to page.
; 8141 ;
```

```
; 8142 ;QUANTITIES IN HARDWARE REGISTERS
; 8143 ;
; 8144 ;SPT      SPT Base Register
; 8145 ;
; 8146 ;          14                      35
; 8147 ;          +-----+
; 8148 ;          !   PHYSICAL CORE WORD ADDRESS   !
; 8149 ;          +-----+
; 8150 ;
; 8151 ;CBR      CST Base Register
; 8152 ;
; 8153 ;          14                      35
; 8154 ;          +-----+
; 8155 ;          !   PHYSICAL CORE WORD ADDRESS   !
; 8156 ;          +-----+
; 8157 ;
; 8158 ;CSTMSK  CST Update Mask
; 8159 ;
; 8160 ;          0                      32 35
; 8161 ;          +-----+-----+
; 8162 ;          !                   MASK           !11111!
; 8163 ;          +-----+-----+
; 8164 ;
; 8165 ;          ANDed with CST word during update
; 8166 ;
; 8167 ;(B32-35 must be all 1's to preserve existing CST information)
; 8168 ;
; 8169 ;CSTDATA  CST Update Data
; 8170 ;
; 8171 ;          0                      32 34 35
; 8172 ;          +-----+-----+-----+
; 8173 ;          !                   DATA           !000!0!
; 8174 ;          +-----+-----+-----+
; 8175 ;
; 8176 ;          IORed with CST word during update
; 8177 ;
; 8178 ;(B32-35 must be all 0's to preserve existing CST information)
; 8179 ;
; 8180 ;All unspecified bits and fields are reserved for future
; 8181 ;specification by DEC.
; 8182 ;
; 8183 ;
```

D 0257, 1215,1553,0100

U 1553, 3615,3551,0303,4374,0007,0700,0000,0016,0000
U 3615, 3616,3771,0006,4354,4007,0700,0000,0000,0000
U 3616, 3617,4551,0606,4374,0007,0700,0000,0040,0000
U 3617, 3620,3333,0006,7174,4007,0700,0400,0000,0210
U 3620, 3621,3771,0005,7274,4007,0701,0000,0000,0230
U 3621, 2514,4553,0500,4374,4007,0331,0000,0003,0000
U 2514, 3631,3771,0013,4370,4007,0700,0000,0040,0002
U 2515, 0100,3440,0303,0174,4156,4700,0400,0000,0000

U 3777, 3622,3333,0003,7174,4007,0700,0400,0000,0211
U 3622, 3623,3333,0006,7174,4007,0700,0400,0000,0214
U 3623, 3624,3771,0006,4354,4007,0700,0000,0000,0000
U 3624, 3625,3333,0006,7174,4007,0700,0400,0000,0210

U 3625, 1060,3333,0004,7174,4007,0370,0400,0000,0212

U 1060, 1060,3773,0000,4304,4003,1702,0000,0000,0000

U 1061, 3666,3333,0005,7174,4007,0700,0400,0000,0213

U 1063, 3626,3771,0006,4374,4007,0700,0000,0000,0000

U 1065, 3664,4571,1206,4374,4007,0700,0000,0037,0000

U 1067, 3664,4571,1206,4374,4007,0700,0000,0037,0000

U 1070, 3631,3333,0005,7174,4007,0700,0400,0000,0213

U 1071, 3666,3333,0005,7174,4007,0700,0400,0000,0213

U 1072, 3631,3333,0005,7174,4007,0700,0400,0000,0213

U 1073, 3631,3333,0005,7174,4007,0700,0400,0000,0213

U 3626, 3627,3333,0006,7174,4007,0700,0400,0000,0160

U 3627, 3630,3333,0006,7174,4007,0700,0400,0000,0161

```
; 8184 .BIN
; 8185
; 8186 .DCODE
; 8187 257: IOT, AC, J/MAP
; 8188 .UCODE
; 8189
; 8190 1553:
; 8191 MAP: [AR]_[AR].OR.#, ;ASSUME PHYSICAL REF
; 8192 #/160000, ;FAKE ANSWER
; 8193 HOLD RIGHT ;
; 8194 [BRX]_VMA ;PUT VMA AND FLAGS IN BRX
; 8195 [BRX]_[BRX].AND.#, ;JUST KEEP USER BIT
; 8196 #/400000, HOLD RIGHT ;
; 8197 WORK[SV.VMA]_[BRX] ;SAVE IN WORKSPACE
; 8198 [BR]_WORK[APR] ;GET APR FLAGS
; 8199 TR [BR], #/030000 ;PAGING ENABLED?
; 8200 =0 STATE [MAP], J/PFMAP ;YES--DO REAL MAP
; 8201 AC [AR], NEXT INST ;NO--RETURN VIRTUAL ADDRESS
; 8202 ;HARDWARE COMES HERE ON PAGE TABLE NOT VALID (OR INTERRUPT) WHEN
; 8203 ; STARTING A MEMORY REFERENCE. MICROWORD ADDRESS OF INSTRUCTION DOING
; 8204 ; MEM WAIT IS SAVED ON THE STACK.
; 8205 3777:
; 8206 PAGE-FAIL:
; 8207 WORK[SV.AR]_[AR]
; 8208 ITRAP: WORK[SV.BRX]_[BRX]
; 8209 [BRX]_VMA
; 8210 WORK[SV.VMA]_[BRX]
; 8211 WORK[SV.ARX]_[ARX],
; 8212 SKIP IRPT ;SEE IF INTERRUPT (SAVE DISPATCH)
; 8213 =0000
; 8214 PFD: DBM/PF DISP, DBUS/DBM, ;BRING CODE TO 2901'S
; 8215 AD/D, DEST/PASS, 4T, ;PUT ON DP 18-21
; 8216 DISP/DP LEFT, J/PFD ;DISPATCH ON IT
; 8217 =0001 ;(1) INTERRUPT
; 8218 WORK[SV.BR]_[BR], J/PFPI1
; 8219 =0011 ;(3) BAD DATA FROM MEMORY
; 8220 [BRX]_IO DATA, ;GET THE BAD DATA
; 8221 AD PARITY OK/O, ;DO NOT LOOK AT PARITY
; 8222 J/BADDATA ;SAVE IN AC BLK 7
; 8223 =0101 ;(5) NXM ERROR
; 8224 [BRX]_[370000] XWD 0, J/HARD
; 8225 =0111 ;(7) NXM & BAD DATA
; 8226 [BRX]_[370000] XWD 0, J/HARD
; 8227 =1000 ;(10) WRITE VIOLATION
; 8228 WORK[SV.BR]_[BR], J/PFMAP
; 8229 =1001 ;[123] (11) 1 ms timer and movsrj
; 8230 WORK[SV.BR]_[BR], J/PFPI1
; 8231 =1010 ;(12) PAGE NOT VALID
; 8232 WORK[SV.BR]_[BR], J/PFMAP
; 8233 =1011 ;(13) EXEC/USER MISMATCH
; 8234 WORK[SV.BR]_[BR], J/PFMAP
; 8235 =
; 8236
; 8237 BADATA:
; 8238 WORK[BADWO]_[BRX] ;SAVE BAD WORD
; 8239 WORK[BADW1]_[BRX] ;AGAIN
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 228-1
PAGE FAIL REFIL LOGIC

U 3630, 3664, 4571, 1206, 4374, 4007, 0700, 0000, 0036, 0000 ; 8240
; 8241

[BRX]_[360000] XWD O, J/HARD

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 229
PAGE FAIL REFIL LOGIC

U 3631, 3632,4223,0000,4364,4277,0700,0200,0000,0010
U 3632, 3633,3551,1313,4374,0007,0700,0000,0002,4000
U 3633, 2516,4553,0600,4374,4007,0321,0000,0002,0000
U 2516, 2517,3551,0606,4374,0007,0700,0000,0001,0000
U 2517, 3634,4551,0606,4374,0007,0700,0000,0041,1000
U 3634, 3635,6551,0606,4374,0007,0700,0000,0000,1000
U 3635, 2520,3441,0605,4174,4007,0700,2000,0071,0007
U 2520, 2520,3447,0505,4174,4007,0630,2000,0060,0000
U 2521, 3636,4251,0505,4374,4007,0700,0000,0000,0777
U 3636, 3657,3447,0504,4174,4007,0700,0000,0000,0000

```
; 8242 ;WE HAVE SAVED AR, ARX, BR AND BRX. WE MERGE IN HERE FROM MAP
; 8243 ; INSTRUCTION, SAVE THE VMA AND START THE PAGE FAIL WORD.
; 8244 PFMAP: ABORT MEM CYCLE ;CLEAR PAGE FAIL
; 8245 [FLG]_[FLG].OR.#, ;PRESET W AND C TO 1
; 8246 FLG.W/1, FLG.C/1, ;BITS INVOLVED
; 8247 HOLD RIGHT ;LEAVE RH ALONE
; 8248 TL [BRX], WRITE TEST/1 ;IS THIS A WRITE TEST?
; 8249 =0 [BRX]_[BRX].OR.#,
; 8250 #/10000,
; 8251 HOLD RIGHT ;YES--TURN INTO WRITE REF
; 8252 [BRX]_[BRX].AND.#, ;START PAGE FAIL WORD
; 8253 #/411000, ;SAVE 3 INTERESTING BITS
; 8254 HOLD RIGHT ;SAVE VIRTUAL ADDRESS
; 8255 ;USER ADDR (400000)
; 8256 ;WRITE REF (010000)
; 8257 ;PAGED REF (001000)
; 8258 [BRX]_[BRX].XOR.#, ;FIX BIT 8
; 8259 #/1000, HOLD RIGHT
; 8260 [BR]_[BRX], ;COPY VIRTUAL ADDRESS
; 8261 SC_7 ;PREPARE TO SHIFT 9 PLACES
; 8262 =0
; 8263 PF25: [BR]_[BR]*.5, ;RIGHT ADJUST PAGE #
; 8264 STEP SC, ;COUNT SHIFT STEPS
; 8265 J/PF25 ;LOOP FOR 9
; 8266 [BR]_[BR].AND.# CLR LH, ;MASK TO 9 BITS
; 8267 #/777 ; ..
;:8268 .IF/KLPAGE
;:8269 .IF/KIPAGE
;:8270 TL [EBR], ;KI MODE REFILL?
;:8271 #/40 ;FLAG BIT
;:8272 =0
;:8273 .ENDIF/KIPAGE
;:8274 READ [BRX], ;USER REF? (KL MODE)
;:8275 SKIP DPO, ; ..
;:8276 J/PF30 ;CONTINUE AT PF30
;:8277 .ENDIF/KLPAGE
;:8278 .IF/KIPAGE
;:8279 [ARX]_[BR]*.5, ;KI10 MODE REFILL
;:8280 J/KIFILL ;GO HANDLE EASY CASE
;:8281 .ENDIF/KIPAGE
;:8282
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 230
PAGE FAIL REFIL LOGIC

```
;;8283 .IF/KLPAGE
;;8284 ;HERE IN TOPS-20 MODE
;;8285 ;PICK UP CORRECT SECTION POINTER
;;8286 =0
;;8287 PF30: [AR]_WORK[PTA.E], ;EXEC MODE
;;8288 SKIP AD.EQ.O, 4T, ;SEE IF VALID
;;8289 J/PF35 ;CONTINUE BELOW
;;8290 [AR]_WORK[PTA.U], ;USER MODE
;;8291 SKIP AD.EQ.O, 4T ;SEE IF VALID
;;8292 =0 VMA_[ARX]+[BR], ;POINTER VALID
;;8293 VMA PHYSICAL READ, ;START MEMORY
;;8294 J/PF77 ;CONTINUE BELOW
;;8295 [AR]_[UBR]+#, 3T, ;USER MODE
;;8296 #/540, ;OFFSET TO UPT
;;8297 J/PF40 ;GO GET POINTER
;;8298
;;8299 =0
;;8300 PF35: VMA_[ARX]+[BR], ;POINTER VALID
;;8301 VMA PHYSICAL READ, ;START MEMORY
;;8302 J/PF77 ;CONTINUE BELOW
;;8303 [AR]_[EBR]+#, 3T, ;EXEC MODE
;;8304 #/540 ;OFFSET TO EPT
;;8305 PF40: VMA_[AR], ;LOAD THE VMA
;;8306 START READ, ;START THE MEMORY CRANKING
;;8307 VMA PHYSICAL ;ABSOLUTE ADDRESS
;;8308 MEM READ, ;WAIT FOR MEMORY
;;8309 [AR] MEM ;POINT POINTER IN AR
;;8310 ;LOOK AT SECTION POINTER AND DISPATCH ON TYPE
;;8311 =000
;;8312 PF45: SC_7, ;FETCH SECTION O POINTER
;;8313 CALL [SETPTR] ;FIGURE OUT POINTER TYPE
;;8314 SECIMM: TL [AR], ;IMMEDIATE POINTER
;;8315 #/77, ;TEST FOR 12-17 = 0
;;8316 J/PF50 ;CONTINUE AT PF50
;;8317 [AR]_[AR]+WORK[SBR], ;SHARED SECTION
;;8318 J/SECshr ;GO FETCH POINTER FROM SPT
;;8319 [AR]_[AR]+WORK[SBR], ;INDIRECT SECTION POINTER
;;8320 CALL [RDPT] ;GO FETCH SPT ENTRY
;;8321 =111 TL [AR], ;12 TO 17 = 0?
;;8322 #/77 ; ..
;;8323 =
;;8324 =0 PAGE FAIL TRAP ;NO
;;8325 [AR]_[AR]*2, ;FIRST SHIFT
;;8326 STEP SC ;SC WAS LOADED AT PF45
;;8327 =0*0
;;8328 PF60: [AR]_[AR]*2, ;CONVERT TO ADDRESS OF
;;8329 STEP SC, ;SECTION TABLE
;;8330 J/PF60
;;8331 CALL [RDPT] ;READ SECTION TABLE
;;8332 =1*1 J/PF45 ;TRY AGAIN
;;8333 =
;;8334
```


; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 231
PAGE FAIL REFIL LOGIC

```
;;8335 ;STILL .IF/KLPAGE
;;8336 ;HERE FOR SHARED SECTION. AR GETS THE ADDRESS OF PAGE TABLE
;;8337 =0**
;;8338 SECSHR: CALL [RDPT] ;READ WORD FROM SPT
;;8339 TL [AR], #/77 ;TEST FOR BITS 12-17 = 0
;;8340
;;8341 ;HERE WITH ADDRESS OF PAGE TABLE IN AR AND SKIP ON
;;8342 ; BITS 12 THRU 17 EQUAL TO ZERO
;;8343 =0
;;8344 PF50: PAGE FAIL TRAP ;BITS 12-17 .NE. 0
;;8345 [ARX][AR].AND.# CLR LH, ;PAGE NUMBER OF PAGE TABLE
;;8346 #/3777 ;11 BIT PHYSICAL PAGE #
;;8347 .IFNOT/NOCST
;;8348 =0* [AR][ARX], ;COPY ADDRESS
;;8349 CALL [UPCST] ;UPDATE CSTO
;;8350 PF70: [AR][AR].OR.WORK[PUR] ;PUT IN NEW AGE AND
;;8351 ; USE BITS
;;8352 .IFNOT/INHCST
;;8353 =0** START NO TEST WRITE, ;START MEMORY WRITE
;;8354 CALL [IBPX] ;GO STORE IN MEMORY
;;8355 .ENDIF/INHCST
;;8356 .IF/INHCST
;;8357 =0** SKIP NO CST, ;SEE IF A CST
;;8358 CALL [WRCST] ;AND GO WRITE IN MEMORY
;;8359 .ENDIF/INHCST
;;8360 SC_7 ;THIS CAN BE BUMMED
;;8361 =0
;;8362 PF75: [ARX][ARX]*2, ;CONVERT PAGE NUMBER TO
;;8363 STEP SC, ;PAGE ADDRESS
;;8364 J/PF75 ;LOOP OVER 9 STEPS
;;8365
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 232
PAGE FAIL REFIL LOGIC

```
;;8366 ;STILL .IF/KLPAGE
;;8367 ;WE NOW HAVE THE ADDRESS OF THE PAGE TABLE ENTRY. GO
;;8368 ; READ IT AND START ANALYSIS
;;8369
;;8370 ;IF WE ARE HERE FOR THE FIRST TIME FOR THE USER OR EXEC SAVE THE
;;8371 ; ADDRESS OF THE PAGE TABLE IN PTA.E OR PTA.U SO THAT WE DO NOT
;;8372 ; HAVE TO DO THE SECTION LOOKUP EVERY TIME.
;;8373 READ [BRX], SKIP DPO ;USER OR EXEC REF?
;;8374 =000 [AR]_WORK[PTA.E], ;EXEC MODE
;;8375 SKIP AD.EQ.O, 4T, ;SEE IF SET YET
;;8376 CALL [SHDREM] ;SHOULD WE REMEMBER PTR
;;8377 [AR]_WORK[PTA.U], ;USER MODE
;;8378 SKIP AD.EQ.O, 4T, ;SEE IF SET YET
;;8379 CALL [SHDREM] ;SHOULD WE REMEMBER PTR
;;8380 WORK[PTA.E]_[ARX], ;SAVE FOR EXEC
;;8381 J/PF76 ;CONTINUE BELOW
;;8382 WORK[PTA.U]_[ARX], ;SAVE FOR USER
;;8383 J/PF76 ;CONTINUE BELOW
;;8384 =111
;;8385 PF76: VMA_[ARX]+[BR], ;READ PAGE POINTER
;;8386 START READ,
;;8387 VMA PHYSICAL
;;8388 =
;;8389 =00
;;8390 PF77: MEM READ, ;START ANALYSIS OF POINTER
;;8391 [AR]_MEM,
;;8392 CALL [SETPTR]
;;8393 PTRIMM: TL [AR], ;IMMEDIATE POINTER
;;8394 #/77, ;CHECK FOR BITS 0-5
;;8395 J/PF80 ;GO TO PF80
;;8396 [AR]_[AR]+WORK[SBR], ;SHARED POINTER
;;8397 J/PTRSHR ;GO TO READ SPT
;;8398
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 233
PAGE FAIL REFIL LOGIC

```
::8399 ;STILL .IF/KLPAGE
::8400 ;INDIRECT POINTER. CHANGE PAGE # AND LOOK FOR PAGE TABLE
::8401 PTRIND: [BR]_[AR] SWAP, ;PUT IN RIGHT HALF
::8402 SKIP7-1 MS ;DID CLOCK GO OFF
::8403 =0 WORK[SV.AR1]_[AR], ;YES--UPDATE CLOCK
::8404 J/PFTICK ;
::8405 [BR]_[BR].AND.# CLR LH, ;UPDATE PAGE # AND RESTART
::8406 #/777, ;MASK FOR PAGE #
::8407 SKIP IRPT ;SEE IF THIS IS A LOOP
::8408 =0 [AR]_[AR].AND.#, ;CHANGE INDIRECT POINTER
::8409 #/277000, ; INTO SHARE POINTER
::8410 HOLD RIGHT, ;
::8411 J/PF45 ;GO BACK AND TRY AGAIN
::8412 PAGE FAIL TRAP ;POINTER LOOP
::8413
::8414 =0**
::8415 PTRSHR: CALL [RDPT] ;GO LOOK AT POINTER
::8416 TL [AR], ;BITS 12-17 .EQ. 0?
::8417 #/77
::8418
::8419 ;HERE WITH FINAL POINTER. SKIP IF 12-17 NOT EQUAL TO ZERO
::8420 =0O
::8421 PF80: PAGE FAIL TRAP ;NO--TAKE A TRAP
::8422 .IFNOT/NO CST
::8423 [ARX]_[AR].AND.# CLR LH, ;SAVE PHYSICAL PAGE #
::8424 #/3777, ;MASK TO 13 BITS
::8425 CALL [UPCST] ;UPDATE CSTO
::8426 .ENDIF/NO CST
::8427 ;HERE WE HAVE CST ENTRY IN AR, PAGE FAIL WORD IN BRX. GO LOOK
::8428 ; AT WRITABLE AND WRITTEN BITS
::8429 =11
::8430 PF90: [BRX]_[BRX].OR.#, ;TRANSLATION IS VALID
::8431 #/100000, HOLD RIGHT ;
::8432 TL [FLG], FLG.W/1 ;IS THIS PAGE WRITABLE?
::8433 =0 [BRX]_[BRX].OR.#, ;YES--INDICATE THAT IN PFW
::8434 #/020000,
::8435 J/PF100 ;NOT WRITE VIOLATION
::8436 TL [BRX], ;IS THIS A WRITE REF?
::8437 WRITE TEST/1, WRITE CYCLE/1
::8438 =0 PAGE FAIL TRAP ;WRITE VIOLATION
::8439 PF107:
::8440 .IFNOT/NO CST
::8441 [AR]_[AR].OR.WORK[PUR], ;PUT IN NEW AGE
::8442 J/PF110 ;GO TO STORE CST ENTRY
::8443 .ENDIF/NO CST
::8444 .IF/NO CST
::8445 PFDONE: TR [FLG],
::8446 #/400000,
::8447 J/PF140
::8448 .ENDIF/NO CST
::8449
::8450 =0*
::8451 PFTICK: [AR] WORK[TIME1], ;UPDATE TIMER
::8452 SPEC/CLRCLK, CALL [TOCK]
::8453 [AR] WORK[SV.AR1], ;RESTORE AR
::8454 J/PTRIND ;GO TRY AGAIN
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 233-1
PAGE FAIL REFIL LOGIC

::8455

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 234
PAGE FAIL REFIL LOGIC

```
;;8456 ;STILL .IF/KLPAGE
;;8457 ;HERE IF PAGE IS WRITABLE
;;8458 PF100: TL [BRX], WRITE CYCLE/1 ;IS THIS A WRITE REF?
;;8459 =0 [AR]_[AR].OR.#, ;YES--SET WRITTEN BIT
;;8460 #/1,
;;8461 HOLD LEFT,
;;8462 J/PF105
;;8463 TR [AR], ;NOT WRITE, ALREADY WRITTEN?
;;8464 #/1
;;8465 =0
;;8466 PF105: [BRX]_[BRX].OR.#, ;WRITTEN SET BIT
;;8467 #/040000, ;MARK PAGE AS
;;8468 HOLD RIGHT, ;WRITABLE
;;8469 J/PF107 ;STORE CST WORD
;;8470 [FLG]_[FLG].AND.NOT.#, ;NOT WRITTEN, CAUSE TRAP ON
;;8471 FLG.W/1, ;WRITE ATTEMPT
;;8472 HOLD RIGHT, ;ONLY CLEAR LH
;;8473 J/PF107
;;8474 .IFNOT/NOCST
;;8475 =0**
;;8476 PF110:
;;8477 .IFNOT/INHCST
;;8478 START NO TEST WRITE,
;;8479 CALL [IBPX] ;STORE CST ENTRY
;;8480 .ENDIF/INHCST
;;8481 .IF/INHCST
;;8482 SKIP NO CST,
;;8483 CALL [WRCST]
;;8484 .ENDIF/INHCST
;;8485
; 8486 .ENDIF/KLPAGE
; 8487
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 235
PAGE FAIL REFIL LOGIC

```
U 3637, 2522,4553,1300,4374,4007,0331,0000,0040,0000
U 2522, 2524,3441,0403,4174,4007,0700,2000,0071,0007
U 2523, 3644,3771,0003,7274,4007,0701,0000,0000,0210
U 2524, 2524,3445,0303,4174,4007,0630,2000,0060,0000
U 2525, 3640,4551,0303,4374,0007,0700,0000,0000,0003
U 3640, 3641,4221,0013,4170,4007,0700,0000,0000,0000
U 3641, 3642,3551,0606,4374,0007,0700,0000,0010,0000
U 3642, 2526,4553,1300,4374,4007,0321,0000,0000,4000
U 2526, 2527,3551,0606,4374,0007,0700,0000,0000,2000
U 2527, 3643,4551,0606,4370,4007,0700,0000,0000,0777
U 3643, 1500,3111,0603,4174,4003,7700,0200,0003,0001

; 8488
; 8489 ;HERE WHEN WE HAVE FIGURED OUT PHYSICAL ADDRESS (IN ARX) AND FLAGS
; 8490 ; (IN BRX) RELOAD PAGE TABLE.
; 8491 PFDONE: TR [FLG], ;MAP INSTRUCTION?
; 8492 #/400000
; 8493 .ENDIF/NOCST
; 8494 =0
; 8495 PF140: [AR]_[ARX], ;GET PHYSICAL PAGE #
; 8496 SC 7, ;PREPARE TO CONVERT TO
; 8497 J/PF130 ; WORD ADDRESS
; 8498 [AR]_WORK[SV.VMA], ;RESTORE VMA
; 8499 J/PF120
; 8500 =0
; 8501 PF130: [AR]_[AR]*2, ;CONVERT TO WORD #
; 8502 STEP SC,
; 8503 J/PF130
; 8504 [AR]_[AR].AND.#, ;JUST ADDRESS BITS
; 8505 #/3,
; 8506 HOLD RIGHT
; 8507 END MAP ;CLEAR MAP FLAGS
; 8508 [BRX]_[BRX].OR.#, ;TURN ON THE TRANSLATION
; 8509 #/100000, ; VALID BIT
; 8510 HOLD RIGHT ; IN LEFT HALF ONLY
; 8511 TL [FLG], FLG.C/1 ;CACHE BIT SET?
; 8512 =0 [BRX]_[BRX].OR.#, ;YES--SET IN MAP WORD
; 8513 #/002000, HOLD RIGHT ; ..
; 8514 [BRX]_[BRX].AND.#, ;PRESERVE WORD #
; 8515 #/777, HOLD LEFT ; IN PAGE FAIL WORD
; 8516 [AR]_[AR].OR.[BRX], ;COMPLETE MAP INSTRUCTION
; 8517 EXIT
; 8518
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 236
PAGE FAIL REFIL LOGIC

```
U 3644, 3645,3441,0305,4174,4007,0700,0000,0000,0000 : 8519 PF120: [BR]_[AR] ;COPY PAGE FAIL WORD
; 8520 [BR]_[AR].AND.NOT.#, ;CLEAR BITS WHICH START A CYCLE
; 8521 READ CYCLE/1, ; ..
; 8522 WRITE CYCLE/1, ; ..
; 8523 WRITE TEST/1, ; ..
U 3645, 3646,5551,0305,4374,0007,0700,0000,0007,0000 : 8524 HOLD RIGHT ;JUST DO LEFT HALF
; 8525 VMA [BR], 3T, ;RESTORE VMA
U 3646, 3647,3443,0500,4174,4007,0701,0200,0000,0030 : 8526 DP_FUNC/1 ;SET USER ACCORDING TO WHAT IT WAS
; 8527 [ARX]_[ARX].AND.# CLR LH, ;JUST KEEP PAGE #
; 8528 #/3777 ; ..
U 3647, 3650,4251,0404,4374,4007,0700,0000,0000,3777 : 8529 [BRX]_[ARX].OR.#, #/400000 ;SET VALID BITS
U 3650, 3651,3551,0406,4374,4007,0700,0000,0040,0000 : 8530 TL [FLG], FLG.W/1 ;WANT WRITE SET?
U 3651, 2530,4553,1300,4374,4007,0321,0000,0002,0000 : 8531 =0 [BRX]_[BRX].OR.#, #/040000 ;SET WRITE BIT
U 2530, 2531,3551,0606,4374,4007,0700,0000,0004,0000 : 8532 TL [FLG], FLG.C/1, ;WANT CACHE SET?
; 8533 LOAD PAGE TABLE ;LOAD PAGE TABLE ON NEXT
; 8534 ; MICRO INSTRUCTION
; 8535 =0 [BRX]_[BRX].OR.#, ;SET CACHE BIT
; 8536 #/020000, J/PF125 ;CACHE BIT
U 2532, 3652,3551,0606,4374,4007,0700,0000,0002,0000 : 8537 READ [BRX] ;LOAD PAGE TABLE
U 2533, 3652,3333,0006,4174,4007,0700,0000,0000,0000 : 8538 PF125: [ARX]_WORK[SV.ARX]
U 3652, 3653,3771,0004,7274,4007,0701,0000,0000,0212 : 8539 [BR]_WORK[SV.BR]
U 3653, 3654,3771,0005,7274,4007,0701,0000,0000,0213 : 8540 [BRX]_WORK[SV.BRX]
U 3654, 3655,3771,0006,7274,4007,0701,0000,0000,0214 : 8541 VMA [AR], ;MAKE MEM REQUEST
; 8542 ;FROM DATA PATH
; 8543 DP_FUNC/1, 3T, ;WAIT FOR PREVIOUS CYCLE TO
; 8544 WAIT/1 ; COMPLETE. (NEED THIS TO
; 8545 ; START ANOTHER CYCLE)
; 8546 [AR]_WORK[SV.AR],
U 3655, 3656,3443,0300,4174,4007,0701,0200,0000,0032 : 8547 RETURN [0]
; 8548
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 237
PAGE FAIL REFIL LOGIC

```
;;8549 .IF/KLPAGE
;;8550 .IFNOT/NOCAST
;;8551 ;SUBROUTINE TO START CST UPDATE
;;8552 ;CALL WITH:
;;8553 ; AR/ PHYSICAL PAGE NUMBER
;;8554 ;RETURN 2 WITH ENTRY IN AR, PAGE FAIL IF AGE TOO SMALL
;;8555 .IFNOT/INHCST
;;8556 =0**
;;8557 UPCST: [AR]_[AR]+WORK[CBR], ;ADDRESS OF CSTO ENTRY
;;8558 CALL [RDPT] ;READ OLD VALUE
;;8559 TL [AR], ;0 - 5 = 0?
;;8560 #/770000 ;
;;8561 =0 [AR]_[AR].AND.WORK[CSTM], ;CLEAR AGE FIELD
;;8562 RETURN [2] ;AGE IS NOT ZERO
;;8563 PAGE FAIL TRAP ;AGE TOO SMALL
;;8564 .ENDIF/INHCST
;;8565 .IF/INHCST
;;8566 UPCST: SKIP NO CST ;SEE IF A CST IS PRESENT
;;8567 =0*0 [AR]_[AR]+WORK[CBR], ;YES, ADDRESS OF CSTO ENTRY
;;8568 CALL [RDPT] ;READ OLD VALUE
;;8569 [AR]_0,RETURN [2] ;NO CST, RETURN
;;8570 TL [AR], ;CHECK AGE FIELD
;;8571 #/770000
;;8572 =
;;8573 =0 [AR]_[AR].AND.WORK[CSTM], ;CLEAR AGE FIELD
;;8574 RETURN [2] ;AGE IS NOT ZERO
;;8575 PAGE FAIL TRAP ;AGE TOO SMALL
;;8576
;;8577 =0
;;8578 WRCST: START NO TEST WRITE,
;;8579 J/IBPX
;;8580 RETURN [4]
;;8581 .ENDIF/INHCST
;;8582 .ENDIF/NOCAST
;;8583
```


; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 238
PAGE FAIL REFIL LOGIC

```
::8584 ;STILL .IF/KLPAGE
::8585 ;SUBROUTINE TO LOOK AT PAGE POINTER
::8586 ;CALL WITH POINTER IN AR
::8587 ;RETURNS 1 IF TYPE 1
::8588 ;RETURNS 2 IF TYPE 2
::8589 ;RETURNS 3 IF TYPE 3
::8590 ;GOES TO PFT IF TYPE 0 OR 4 THRU 7
::8591 SETPTR: [ARX]_[AR].OR.#, ;AND C AND W BITS
::8592 #/753777 ; OF ALL POINTERS
::8593 [FLG]_[FLG].AND.[ARX], ; ..
::8594 HOLD RIGHT ;KEEP IN LH OF FLG
::8595 READ [AR], ;TYPE 4,5,6 OR 7?
::8596 SKIP DPO ; ..
::8597 =0 TL [AR], ;HERE WE TEST FOR TYPE
::8598 #/300000, ; ZERO POINTER
::8599 J/STPTR1 ;CHECK AT STPTR1
::8600 PAGE FAIL TRAP ;BAD TYPE
::8601 =0 ;
::8602 STPTR1: TL [AR], ;NOT ZERO
::8603 #/100000, ;SEPERATE TYPE 2
::8604 J/STPTR2 ; ..
::8605 PAGE FAIL TRAP ;TYPE 0
::8606 ;
::8607 =0 ;
::8608 STPTR2: TL [AR], ;SEPERATE TYPE 1
::8609 #/200000, ; AND 3
::8610 J/STPTR3 ; ..
::8611 RETURN [2] ;TYPE 2
::8612 ;
::8613 =0 ;
::8614 STPTR3: RETURN [3] ;TYPE 3
::8615 RETURN [1] ;TYPE 1
::8616 ;
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 239
PAGE FAIL REFIL LOGIC

```
;;8617 ;STILL .IF/KLPAGE
;;8618 ;SUBROUTINE TO FETCH A PAGE POINTER OR CST ENTRY
;;8619 ;CALL WITH ADDRESS IN AR
;;8620 ;RETURN 4 WITH WORD IN AR
;;8621 ;
;;8622 RDPT: VMA [AR], ;LOAD THE VMA
;;8623 START READ, ;START MEM CYCLE
;;8624 VMA PHYSICAL, ;ABSOLUTE ADDRESS
;;8625 SKIP IRPT ;CHECK FOR INTERRUPTS
;;8626 =0 MEM READ, ;NO INTERRUPTS
;;8627 [AR] MEM, ;PUT THE DATA INTO AR
;;8628 RETURN [4] ;AND RETURN
;;8629 PAGE FAIL TRAP ;INTERRUPT
;;8630
;;8631
;;8632 ;SUBROUTINE TO SEE IF WE SHOULD REMEMBER AN EXEC SECTION PTR
;;8633 ;CALL WITH SKIP ON ADR.EQ.O
;;8634 ;RETURNS 2 IF WE SHOULD STORE AND 7 IF WE SHOULD NOT
;;8635 ;
;;8636 =0
;;8637 SHDREM: RETURN [7] ;INDIRECT PTR
;;8638 [AR].NOT.[FLG] ;FLIP BITS
;;8639 TL [AR], FLG.W/1, FLG.C/1 ;BOTH BITS SET
;;8640 =0 RETURN [7] ;NO--DON'T STORE
;;8641 RETURN [2] ;STORE
;;8642
;;8643 .ENDIF/KLPAGE
;;8644
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 240
PAGE FAIL REFIL LOGIC

```
U 3657, 2534,3333,0006,4174,4007,0520,0000,0000,0000 ; 8645 .IF/KIPAGE
; 8646 ;HERE IN KI10 MODE
; 8647 ;BR CONTAINS PAGE # AND ARX CONTAINS PAGE #/2
; 8648
; 8649 KIFILL: READ [BR], ;USER REF?
; 8650 SKIP DPO ; ..
; 8651 =0 [BR]-#, ;EXEC--LESS THAN 340?
; 8652 #/340, ; ..
; 8653 SKIP DP18, 4T, ; ..
; 8654 J/KIF10 ;FOLLOW EXEC PATH
; 8655 KIUPT: [ARX]_[ARX]+[UBR], ;POINTER TO PAGE MAP ENTRY
; 8656 LOAD VMA, ;PUT ADDRESS IN VMA
; 8657 VMA PHYSICAL, ;ABSOLUTE ADDRESS
; 8658 START READ, ;FETCH UPT WORD
; 8659 J/KIF30 ;JOIN COMMON CODE
; 8660 =0
; 8661 KIF10: [BR]-#, ;EXEC ADDRESS .GE. 340
; 8662 #/400, ; SEE IF .GT. 400
; 8663 SKIP DP18, 4T, ; ..
; 8664 J/KIEPT ;LOOK AT KIF20
; 8665 [ARX]_[ARX]+#, 3T, ;EXEC ADDRESS .LT. 340
; 8666 #/600, ;IN EBR+600
; 8667 J/KIEPT ;JOIN COMMON CODE
; 8668
; 8669 =0
; 8670 KIEPT: [ARX]_[ARX]+[EBR], ;ADD OFFSET TO
; 8671 LOAD VMA, ; EPT
; 8672 START READ, ;START FETCH
; 8673 VMA PHYSICAL, ;ABSOLUTE ADDRESS
; 8674 J/KIF30 ;GO GET POINTER
; 8675 [ARX]_[ARX]+#, ;PER PROCESS PAGE
; 8676 #/220, 3T, ; IS IN UPT + 400
; 8677 J/KIUPT ;JOIN COMMON CODE
; 8678 KIF30: MEM READ, ;WAIT FOR DATA
; 8679 [ARX]_MEM ;PLACE IT IN ARX
; 8680 TR [BR], ;SEE IF EVEN OR ODD
; 8681 #/1 ; ..
; 8682
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31.(254)

KS10 MICROCODE V124, 27-JUL-84 Page 241
PAGE FAIL REFIL LOGIC

```
U 2542, 2544, 3333, 0004, 4174, 4007, 0530, 0000, 0000, 0000
U 2543, 2542, 3770, 0404, 4344, 4007, 0700, 0000, 0000, 0000
U 2544, 2560, 4553, 1300, 4374, 4007, 0321, 0000, 0001, 0000
U 2545, 3662, 5551, 1313, 4374, 4007, 0700, 0000, 0002, 4000
U 3662, 2546, 4553, 0400, 4374, 4007, 0331, 0000, 0002, 0000
U 2546, 2547, 3551, 1313, 4374, 0007, 0700, 0000, 0000, 4000
U 2547, 2550, 4553, 0400, 4374, 4007, 0331, 0000, 0004, 0000
U 2550, 2551, 3551, 0606, 4374, 0007, 0700, 0000, 0002, 0000
U 2551, 2552, 4553, 0400, 4374, 4007, 0331, 0000, 0010, 0000
U 2552, 3663, 3551, 1313, 4374, 0007, 0700, 0000, 0002, 0000
U 2553, 2554, 4553, 0600, 4374, 4007, 0321, 0000, 0001, 0000
U 2554, 2544, 3551, 0606, 4374, 0007, 0700, 0000, 0010, 0000
U 2555, 3637, 4443, 0000, 4174, 4007, 0700, 0000, 0000, 0000
U 3663, 3637, 3551, 0606, 4374, 4007, 0700, 0000, 0004, 0000

; 8683 ;STILL .IF/KIPAGE
; 8684 =0
; 8685 KIF40: READ [ARX], ;ODD
; 8686 SKIP DP18, ;SEE IF VALID
; 8687 J/KIF50 ;JOIN COMMON CODE
; 8688 [ARX]_[ARX] SWAP, ;EVEN--FLIP AROUND
; 8689 J/KIF40 ; AND CONTINUE
; 8690
; 8691 .ENDIF/KIPAGE
; 8692 =0
; 8693 KIF50: PAGE FAIL TRAP
; 8694 ;AT THIS POINT WE HAVE THE PAGE MAP ENTRY IN RH OF AR
; 8695 [FLG]_[FLG].AND.NOT.#, ;CLEAR W AND C
; 8696 FLG.W/1, FLG.C/1 ; FLAGS
; 8697 TR [ARX], #/020000 ;CACHE ENABLED?
; 8698 =0 [FLG]_[FLG].OR.#, ;SET CACHE BITS
; 8699 FLG.C/1, HOLD RIGHT ; ..
; 8700 TR [ARX], #/040000 ;DO NOT CACHE
; 8701 ;SEE IF CACHE BIT SET
; 8702 =0 [BRX]_[BRX].OR.#, ;COPY BITS TO BRX
; 8703 #/020000,
; 8704 HOLD RIGHT
; 8705 TR [ARX], ; ..
; 8706 #/100000
; 8707 =0 [FLG]_[FLG].OR.#, ;SAVE W
; 8708 FLG.W/1, ; ..
; 8709 HOLD RIGHT, ; ..
; 8710 J/KIF90 ;ALL DONE
; 8711 TL [BRX], ;W=0, WRITE REF?
; 8712 WRITE CYCLE/1
; 8713 =0
; 8714 KIF80: [BRX]_[BRX].OR.#, ;WRITE FAILURE
; 8715 #/100000, HOLD RIGHT, ;INDICATE THAT ACCESS WAS ON
; 8716 J/KIF50 ;GO PAGE FAIL
; 8717 J/PFDONE ;ALL DONE
; 8718
; 8719 KIF90: [BRX]_[BRX].OR.#, ;PAGE IS WRITABLE
; 8720 #/40000, ;TURN ON IN BRX
; 8721 J/PFDONE ;ALL SET
; 8722
```

; T10KI.MCR[10,1141] 11:45 11-AUG-1984
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 242
PAGE FAIL REFIL LOGIC

```
U 3664, 2556,3333,0005,7174,4007,0700,0400,0000,0213 ; 8723 ;HERE ON HARD PAGE FAILURES
; 8724 HARD: WORK[SV.BR]_[BR] ;SAVE BR (CLEANUP MAY NEED IT)
; 8725 =0 [BR]_VMA, ;BUILD PAGE FAIL WORD
U 2556, 3670,3771,0005,4354,4007,0700,0010,0000,0000 ; 8726 CALL [ABORT] ;CLEAR ERROR
; 8727 [BR]_[BR].AND.#, ;SAVE THE FLAGS
; 8728 #/401237, ;
; 8729 HOLD RIGHT ;
U 2557, 3665,4551,0505,4374,0007,0700,0000,0040,1237 ; 8730 [BRX]_[BRX].OR.[BR], ;COMPLETE PAGE FAIL WORD
U 3665, 2544,3111,0506,4174,4007,0700,0000,0000,0000 ; 8731 J/KIF50 ;GO TRAP
; 8732
U 3666, 1114,4443,0000,4174,4007,0370,0000,0000,0000 ; 8733 PFPI1: SKIP IRPT ;TIMER TRAP?
; 8734 =00
; 8735 [AR]_WORK[TIME1], ;YES--GET LOW WORD
; 8736 SPEC/CLRCLK, ;CLEAR CLOCK FLAG
U 1114, 3461,3771,0003,7274,4117,0701,0010,0000,0301 ; 8737 CALL [TOCK] ;DO THE UPDATE
U 1115, 2561,4443,0000,4174,4007,0700,0000,0000,0000 ; 8738 J/PFT1 ;EXTERNAL INTERRUPT
U 1116, 3667,4223,0000,4364,4277,0700,0200,0000,0010 ; 8739 ABORT MEM CYCLE ;CLEAR IMS FLAGS
; 8740 =
; 8741 PFPI2: [AR]_WORK[SV.VMA], ;RESTORE VMA
U 3667, 3652,3771,0003,7274,4007,0701,0000,0000,0210 ; 8742 J/PF125
; 8743
; 8744
U 3670, 0001,4223,0000,4364,4274,1700,0200,0000,0010 ; 8745 ABORT: ABORT MEM CYCLE, RETURN [1]
; 8746
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 243
PAGE FAIL REFIL LOGIC

```
U 2560, 0104,4751,1217,4374,4007,0700,0000,0000,0100
U 2561, 2562,3771,0003,7274,4007,0611,0000,0000,0210
U 2562, 2564,4553,0300,4374,4007,0321,0000,0010,0000
U 2563, 3671,3771,0003,7274,4007,0701,0000,0000,0425
U 3671, 2566,3333,0003,4174,4467,0700,0000,0000,0004
U 2564, 1100,4443,0000,4174,4007,0700,0000,0000,0000
U 2565, 2566,1111,0701,4170,4007,0700,4000,0000,0000
U 2566, 1100,3333,0013,4174,4003,5701,0000,0000,0000

U 1100, 2570,4221,0013,4170,4007,0370,0000,0000,0000
U 1101, 3051,3771,0003,7274,4007,0701,0000,0000,0212
U 1102, 3674,0111,0701,4174,4007,0700,0000,0000,0000
U 1103, 3366,3771,0013,4370,4007,0700,0000,0000,0011
U 1104, 3366,3771,0013,4370,4007,0700,0000,0000,0012
U 1105, 2272,3771,0013,4370,4007,0700,0000,0000,0003
U 1106, 3375,3771,0013,4370,4007,0700,0000,0000,0012
U 1107, 3372,3771,0013,4370,4007,0700,0000,0000,0011
U 1110, 3364,3771,0013,4370,4007,0700,0000,0000,0011
U 1111, 2270,4221,0013,4170,4007,0700,0000,0000,0000
U 1112, 2272,4221,0013,4170,4007,0700,0000,0000,0000
U 1113, 2272,3771,0013,4370,4007,0700,0000,0000,0011

; 8747 ;HERE ON PAGE FAIL TRAP
; 8748 =0
; 8749 PFT: HALT [IOPF] ;IO PAGE FAILURE
; 8750 PFT1: [AR]_WORK[SV.VMA], ;SEE IF TRAP CYCLE
; 8751 SKIP/TRAP CYCLE ;IS THIS AN INSTRUCTION FETCH
; 8752 =0 TL [AR], FETCH/1, ;GO LOOK BELOW
; 8753 J/PFT1A ;RESTORE PC
; 8754 [AR]_WORK[TRAPPC] ;RESTORE PC
; 8755 READ [AR], LOAD FLAGS, J/CLDISP
; 8756 =0
; 8757 PFT1A: J/CLEANED ;YES--NO PC TO BACK UP
; 8758 FIXPC: [PC][PC]-1, HOLD LEFT ;DATA FAILURE--BACKUP PC
; 8759 =0
; 8760 CLDISP: CLEANUP DISP ;GO CLEANUP AFTER PAGE FAIL
; 8761 =0000
; 8762 CLEANUP:
; 8763 CLEANED: ;(0) NORMAL CASE
; 8764 END STATE, SKIP IRPT, ;NO MORE CLEANUP NEEDED
; 8765 J/PFT2 ;HANDLE PAGE FAIL OR INTERRUPT
; 8766 [AR]_WORK[SV.ARX], ;(1) BLT
; 8767 J/BLT-CLEANUP
; 8768 [PC][PC]+1, ;(2) MAP
; 8769 J/MAPDON
; 8770 STATE [EDIT-SRC], ;(3) SRC IN STRING MOVE
; 8771 J/STRPF
; 8772 STATE [EDIT-DST], ;(4) DST IN STRING MOVE
; 8773 J/STRPF
; 8774 STATE [SRC], ;(5) SRC+DST IN STRING MOVE
; 8775 J/BACKD
; 8776 STATE [EDIT-DST], ;(6) FILL IN MOVSRJ
; 8777 J/STRPF4
; 8778 STATE [EDIT-SRC], ;(7) DEC TO BIN
; 8779 J/PFDBIN
; 8780 STATE [EDIT-SRC], ;(10) SRC+DST IN COMP
; 8781 J/CMSDST
; 8782 END STATE, J/BACKS ;(11) EDIT SRC FAIL
; 8783 END STATE, J/BACKD ;(12) EDIT DST FAIL
; 8784 STATE [EDIT-SRC], ;(13) SRC+DST IN EDIT
; 8785 J/BACKD
; 8786 =
; 8787
```

; T10KI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 244
PAGE FAIL REFIL LOGIC

U 2570, 3672,0551,1103,4374,4007,0701,0000,0000,0500
U 2571, 0770,3551,1313,4374,0007,0700,0000,0001,0000
U 3672, 2572,3443,0300,4174,4007,0700,0200,0021,1016
U 2572, 3676,3333,0006,4175,5007,0701,0210,0000,0002
U 2573, 3673,3741,0105,4074,4007,0700,0000,0000,0000
U 3673, 3675,3333,0005,4175,5007,0701,0200,0000,0002
U 3674, 2574,4221,0013,4170,4007,0370,0000,0000,0000
U 2574, 1500,3441,0603,4174,4003,7700,0200,0003,0001
U 2575, 2571,1111,0701,4174,4007,0700,4000,0000,0000

```
; 8788 =0
; 8789 PFT2: [AR]_[UBR]+#, ;PREPARE TO STORE PFW
; 8790 #/500, 3T,
; 8791 J/PFT10
; 8792 PFT3: TAKE INTERRUPT ;PROCESS INTERRUPT
; 8793 PFT10: VMA [AR], ;WHERE TO STORE PFW
; 8794 VMA PHYSICAL WRITE
; 8795 =0 MEM WRITE, ;STORE PFW
; 8796 MEM [BR],
; 8797 CALL [NEXTAR] ;ADVANCE POINTER TO
; 8798 ;PREPARE TO STORE PC
; 8799 .IF/KLPAGE
; 8800 .IF/KIPAGE
; 8801 TL [EBR], #/400000 ;KL PAGING?
; 8802 =0
; 8803 .ENDIF/KIPAGE
; 8804 [BR]_FLAGS,J/EAPF ;YES--DO EXTENDED THING
; 8805 .ENDIF/KLPAGE
; 8806
; 8807 .IF/KIPAGE
; 8808 [BR]_PC WITH FLAGS ;GET OLD PC
; 8809 MEM WRITE, ;STORE OLD PC
; 8810 MEM [BR],
; 8811 J/EAPF1
; 8812 .ENDIF/KIPAGE
; 8813
; 8814 MAPDON: END STATE, ;CLEAR MAP BIT
; 8815 SKIP IRPT ;ANY INTERRUPT?
; 8816 =0 [AR]_[BR], ;RETURN PAGE FAIL WORD
; 8817 EXIT
; 8818 [PC]_[PC]-1, J/PFT3 ;INTERRUPTED OUT OF MAP
; 8819 ; RETRY INSTRUCTION
; 8820
```

; T1OKI.MCR[10,1141]
; PAGEF.MIC[10,1141]

11:45 11-AUG-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 245
PAGE FAIL REFIL LOGIC

U 3675, 2625,0111,0703,4174,4007,0700,0200,0024,1016

U 3676, 0001,0111,0703,4170,4004,1700,0200,0023,1016

```
; 8821  
; 8822 .IF/KLPAGE  
; 8823 =0  
; 8824 EAPF: MEM WRITE, MEM_[BR], ;STORE FLAGS  
; 8825 CALL [NEXTAR] ;STORE PC WORD  
; 8826 MEM WRITE, MEM_[PC] ; ..  
; 8827 .ENDIF/KLPAGE  
; 8828  
; 8829 EAPF1: [AR]_[AR]+1,  
; 8830 VMA PHYSICAL READ,  
; 8831 J/GOEXEC  
; 8832  
; 8833 NEXTAR: NEXT [AR] PHYSICAL WRITE, RETURN [1]  
; 8834
```

```
; Number of microwords used:  
; D words= 512  
; U words= 1954, Highest= 2047
```

END

(U) A

AR

| | | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| 636 # | | | | | | | | | | | | |
| 640 # | 2388 | 2426 | 2456 | 2457 | 2520 | 2568 | 2576 | 2715 | 2717 | 2720 | 2723 | |
| 2726 | 2729 | 2813 | 2823 | 2838 | 2855 | 2865 | 2875 | 2885 | 2916 | 2926 | 2936 | |
| 2971 | 2983 | 2990 | 2994 | 3004 | 3005 | 3007 | 3009 | 3010 | 3018 | 3041 | 3042 | |
| 3268 | 3273 | 3275 | 3287 | 3290 | 3293 | 3298 | 3299 | 3369 | 3386 | 3388 | 3453 | |
| 3456 | 3524 | 3525 | 3531 | 3551 | 3578 | 3593 | 3629 | 3640 | 3652 | 3706 | 3729 | |
| 3734 | 3769 | 3813 | 3817 | 3824 | 3826 | 3986 | 3991 | 4021 | 4056 | 4070 | 4086 | |
| 4097 | 4100 | 4103 | 4117 | 4144 | 4145 | 4150 | 4167 | 4183 | 4203 | 4315 | 4324 | |
| 4329 | 4339 | 4345 | 4364 | 4368 | 4386 | 4414 | 4418 | 4420 | 4421 | 4440 | 4472 | |
| 4475 | 4478 | 4492 | 4517 | 4534 | 4535 | 4540 | 4545 | 4553 | 4555 | 4557 | 4628 | |
| 4638 | 4640 | 4645 | 4646 | 4691 | 4696 | 4706 | 4710 | 4748 | 4752 | 4756 | 4760 | |
| 4936 | 4949 | 5034 | 5037 | 5049 | 5052 | 5058 | 5064 | 5099 | 5174 | 5177 | 5279 | |
| 5285 | 5289 | 5290 | 5320 | 5326 | 5345 | 5351 | 5352 | 5353 | 5354 | 5363 | 5368 | |
| 5388 | 5389 | 5401 | 5403 | 5412 | 5413 | 5433 | 5438 | 5439 | 5445 | 5480 | 5484 | |
| 5486 | 5489 | 5492 | 5496 | 5498 | 5500 | 5502 | 5503 | 5504 | 5505 | 5511 | 5513 | |
| 5514 | 5523 | 5528 | 5544 | 5570 | 5578 | 5583 | 5585 | 5629 | 5642 | 5661 | 5670 | |
| 5674 | 5689 | 5703 | 5705 | 5707 | 5713 | 5740 | 5747 | 5750 | 5762 | 5765 | 5773 | |
| 5775 | 5779 | 5784 | 5786 | 5802 | 5807 | 5810 | 5832 | 5835 | 5836 | 5840 | 5958 | |
| 5962 | 5966 | 5973 | 5984 | 5993 | 6008 | 6028 | 6032 | 6045 | 6048 | 6074 | 6158 | |
| 6165 | 6182 | 6204 | 6218 | 6245 | 6256 | 6257 | 6258 | 6271 | 6305 | 6335 | 6346 | |
| 6384 | 6399 | 6427 | 6432 | 6461 | 6462 | 6463 | 6464 | 6466 | 6467 | 6470 | 6475 | |
| 6524 | 6545 | 6552 | 6553 | 6579 | 6617 | 6629 | 6630 | 6637 | 6657 | 6660 | 6663 | |
| 6667 | 6668 | 6670 | 6674 | 6676 | 6730 | 6731 | 6736 | 6737 | 6744 | 6747 | 6751 | |
| 6820 | 6842 | 6848 | 6867 | 6885 | 6889 | 6892 | 6893 | 6894 | 6895 | 6917 | 6942 | |
| 6943 | 6947 | 6953 | 6954 | 6958 | 7028 | 7031 | 7052 | 7054 | 7056 | 7058 | 7061 | |
| 7063 | 7065 | 7101 | 7107 | 7115 | 7119 | 7122 | 7140 | 7146 | 7148 | 7153 | 7169 | |
| 7250 | 7331 | 7333 | 7335 | 7337 | 7339 | 7341 | 7343 | 7345 | 7347 | 7369 | 7375 | |
| 7437 | 7447 | 7453 | 7456 | 7457 | 7474 | 7482 | 7532 | 7543 | 7556 | 7558 | 7584 | |
| 7589 | 7616 | 7619 | 7647 | 7665 | 7666 | 7725 | 7822 | 7830 | 7861 | 8191 | 8201 | |

ARX

| | | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| 8501 | 8504 | 8519 | 8520 | 8541 | 8752 | 8793 | | | | | | |
| 641 # | 2488 | 2770 | 2771 | 3709 | 3711 | 3724 | 3753 | 4083 | 4095 | 4104 | 4105 | |
| 4122 | 4125 | 4127 | 4149 | 4169 | 4185 | 4205 | 4208 | 4228 | 4357 | 4385 | 4437 | |
| 4463 | 4464 | 4543 | 4635 | 4636 | 4644 | 4713 | 4903 | 4910 | 4911 | 4939 | 4942 | |
| 4963 | 5093 | 5115 | 5287 | 5325 | 5357 | 5543 | 5564 | 5567 | 5627 | 5644 | 5666 | |
| 5686 | 5706 | 5711 | 5757 | 5796 | 5803 | 6046 | 6087 | 6094 | 6128 | 6162 | 6212 | |
| 6221 | 6231 | 6240 | 6263 | 6264 | 6272 | 6304 | 6308 | 6385 | 6422 | 6430 | 6434 | |
| 6436 | 6753 | 6756 | 6802 | 6870 | 6944 | 7053 | 7057 | 7059 | 7062 | 7064 | 7093 | |
| 7095 | 7281 | 7294 | 7298 | 7488 | 7671 | 7673 | 7731 | 7958 | 8495 | 8527 | 8529 | |

BR

| | | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| 8665 | 8675 | 8688 | 8697 | 8700 | 8705 | | | | | | | |
| 642 # | 2187 | 2384 | 2419 | 2421 | 2425 | 2906 | 3037 | 3070 | 3071 | 3074 | 3075 | |
| 3076 | 3077 | 3092 | 3095 | 3098 | 3101 | 3105 | 3108 | 3110 | 3142 | 3143 | 3146 | |
| 3147 | 3428 | 3431 | 3434 | 3437 | 3440 | 3443 | 3446 | 3449 | 3774 | 3779 | 3807 | |
| 3832 | 3840 | 3841 | 3988 | 3993 | 4091 | 4223 | 4388 | 4390 | 4398 | 4402 | 4465 | |
| 4468 | 4542 | 4546 | 4547 | 4548 | 4551 | 4556 | 4587 | 4729 | 4731 | 4790 | 4791 | |
| 4792 | 4793 | 4794 | 4801 | 4810 | 4814 | 4823 | 4884 | 4885 | 4886 | 4887 | 4888 | |
| 4909 | 4952 | 4958 | 5061 | 5101 | 5103 | 5113 | 5118 | 5202 | 5286 | 5293 | 5294 | |
| 5298 | 5299 | 5355 | 5358 | 5392 | 5443 | 5526 | 5550 | 5552 | 5562 | 5576 | 5655 | |
| 5673 | 5676 | 5912 | 5924 | 5925 | 5927 | 5928 | 5955 | 5964 | 6009 | 6014 | 6039 | |
| 6042 | 6133 | 6198 | 6201 | 6202 | 6215 | 6223 | 6229 | 6352 | 6362 | 6373 | 6382 | |
| 6386 | 6404 | 6457 | 6472 | 6510 | 6521 | 6588 | 6844 | 6845 | 6886 | 6887 | 6927 | |
| 6935 | 6946 | 6950 | 7027 | 7030 | 7050 | 7084 | 7086 | 7089 | 7131 | 7139 | 7152 | |
| 7154 | 7179 | 7180 | 7187 | 7188 | 7194 | 7249 | 7260 | 7267 | 7336 | 7338 | 7340 | |
| 7346 | 7348 | 7384 | 7397 | 7494 | 7517 | 7518 | 7519 | 7560 | 7562 | 7590 | 7594 | |
| 7596 | 7600 | 7602 | 7626 | 7632 | 7723 | 8199 | 8263 | 8266 | 8279 | 8525 | 8651 | |
| 8661 | 8680 | 8727 | 8730 | | | | | | | | | |

| | | | | | | | | | | | | |
|-------|--------|------|------|------|------|------|------|------|------|------|------|------|
| BRX | 643 # | 4119 | 4147 | 4153 | 4224 | 4247 | 4267 | 4271 | 4275 | 4279 | 4285 | 4289 |
| | 4293 | 4297 | 4374 | 4427 | 4467 | 4579 | 4582 | 4593 | 5040 | 5092 | 5121 | 5129 |
| | 5136 | 5160 | 5245 | 5248 | 5318 | 5319 | 5321 | 5360 | 5365 | 5555 | 5572 | 5638 |
| | 5652 | 5653 | 5695 | 5916 | 6015 | 6085 | 6096 | 6097 | 6121 | 6170 | 6178 | 6259 |
| | 6288 | 6301 | 6310 | 6315 | 6323 | 6325 | 6334 | 6340 | 6345 | 6357 | 6396 | 6398 |
| | 6416 | 6425 | 6452 | 6454 | 6456 | 6487 | 6490 | 6506 | 6518 | 6520 | 6600 | 6602 |
| | 6605 | 6671 | 6678 | 6772 | 6776 | 6780 | 6784 | 6788 | 6792 | 6808 | 7076 | 7196 |
| | 7197 | 7198 | 7296 | 7446 | 7501 | 7502 | 7565 | 7648 | 8195 | 8248 | 8249 | 8252 |
| | 8258 | 8260 | 8508 | 8512 | 8514 | 8516 | 8531 | 8535 | 8702 | 8711 | 8714 | 8719 |
| | 8816 | | | | | | | | | | | |
| EBR | 645 # | 6974 | 7177 | 7449 | 7492 | 8670 | | | | | | |
| FLG | 648 # | 3620 | 5507 | 5608 | 5609 | 5616 | 5743 | 5746 | 5749 | 5752 | 5812 | 5819 |
| | 5820 | 6596 | 8245 | 8491 | 8511 | 8530 | 8532 | 8693 | 8695 | 8698 | 8707 | 8792 |
| HR | 639 # | 2288 | 2294 | 2300 | 2306 | 2315 | 2319 | 2328 | 2333 | 2434 | 2782 | 2787 |
| | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3563 | 3567 | 3577 | 3582 | 3585 | 3605 |
| | 3626 | 3639 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 | 3938 | 4036 |
| | 5914 | 6907 | 6984 | 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 |
| | 7014 | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7217 | 7239 |
| | 7334 | 7650 | 7664 | 7793 | 7802 | 7804 | 7806 | 7808 | 7810 | 7812 | | |
| MAG | 637 # | 2423 | 4129 | 4132 | 4151 | 4154 | 4155 | 4168 | 4174 | 4176 | 4210 | 4212 |
| | 4214 | 4392 | 4410 | 4432 | 4436 | 4439 | 4480 | 4482 | 4483 | 4610 | 4615 | 4620 |
| | 4626 | 4627 | 5222 | 5223 | 5546 | 5637 | 5692 | 5718 | 5753 | 5788 | 5792 | 5794 |
| MASK | 647 # | 2149 | 2150 | 2151 | 2154 | 2157 | 2190 | 2201 | 2237 | 2241 | 2244 | 2247 |
| | 2256 | 2259 | 2262 | 2265 | 2516 | 2739 | 2760 | 2967 | 3015 | 3579 | 3627 | 3641 |
| | 3662 | 3942 | 3982 | 3985 | 4030 | 4090 | 4226 | 4260 | 4394 | 4405 | 4433 | 4558 |
| | 4770 | 4890 | 4901 | 5207 | 5227 | 5228 | 5229 | 5243 | 5448 | 5508 | 5581 | 5631 |
| | 5721 | 5790 | 5879 | 5881 | 5883 | 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | 6130 |
| | 6132 | 6154 | 6269 | 6591 | 7248 | 7291 | 7292 | 7293 | 7427 | 7428 | 7429 | 7430 |
| | 7431 | 7432 | 7433 | 7439 | 7458 | 7468 | 7486 | 7500 | 7578 | 7581 | 7617 | 7628 |
| | 7634 | 7651 | 7707 | 7860 | 8224 | 8226 | 8240 | 8749 | | | | |
| ONE | 644 # | 2179 | 2183 | 2228 | 2230 | 2275 | 2362 | 2413 | 2416 | 2946 | 3337 | 3403 |
| | 3419 | 3486 | 3500 | 3526 | 3535 | 3538 | 3613 | 3631 | 3634 | 3978 | 4025 | 4041 |
| | 4320 | 4735 | 5032 | 5100 | 5141 | 5143 | 5148 | 5165 | 5166 | 5250 | 5251 | 5435 |
| | 5447 | 5522 | 5671 | 5829 | 5950 | 5991 | 6024 | 6030 | 6051 | 6069 | 6079 | 6098 |
| | 6099 | 6134 | 6135 | 6177 | 6241 | 6246 | 6309 | 6321 | 6328 | 6348 | 6355 | 6360 |
| | 6369 | 6379 | 6383 | 6387 | 6388 | 6424 | 6438 | 6449 | 6501 | 6525 | 6619 | 6641 |
| | 6694 | 6817 | 6818 | 6824 | 6846 | 6888 | 6930 | 7255 | 7278 | 7308 | 7360 | 7644 |
| | 7721 | 7846 | 7932 | 7939 | 7942 | 7945 | 7948 | 7951 | 8758 | 8768 | 8818 | 8829 |
| | 8833 | | | | | | | | | | | |
| PC | 638 # | 2227 | 2231 | 2249 | 2342 | 2404 | 2429 | 3325 | 3452 | 3457 | 3688 | 3805 |
| | 3811 | 3829 | 4014 | 4215 | 4217 | 4692 | 4701 | 6963 | 7351 | 7470 | 7622 | 7857 |
| | 8808 | | | | | | | | | | | |
| PI | 649 # | 3591 | 3592 | 3606 | 7009 | 7010 | 7328 | 7342 | 7344 | 7349 | 7359 | 7361 |
| | 7420 | 7421 | 7422 | 7423 | 7424 | 7425 | 7426 | 7435 | | | | |
| TO | 651 # | 4195 | 4199 | 4435 | 4446 | 4453 | 4456 | 4458 | 4459 | 5563 | 5571 | 5597 |
| | 5599 | 5600 | 5601 | 5604 | 5612 | 5615 | 5617 | 5657 | 5723 | 5731 | 5732 | 6353 |
| | 6420 | 7362 | 7498 | | | | | | | | | |
| T1 | 652 # | 4412 | 4552 | 5218 | 5220 | 5224 | 5239 | 5242 | 5587 | 5672 | 5675 | 5694 |
| | 5704 | | | | | | | | | | | |
| UBR | 646 # | 3944 | 3984 | 6966 | 7112 | 7134 | 7186 | 7193 | 8655 | 8789 | | |
| XWD1 | 650 # | 3508 | 3678 | 7484 | | | | | | | | |
| (D) A | 1314 # | | | | | | | | | | | |
| DBLAC | 1318 # | 2774 | | | | | | | | | | |
| DFP | 1324 # | 5533 | 5534 | 5622 | 5683 | | | | | | | |
| DREAD | 1317 # | 2765 | 2766 | 4078 | 4079 | 4162 | 4380 | | | | | |

| | | | | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| DSHIFT | 1320 # | 2956 | 2957 | | | | | | | | | |
| FP | 1322 # | 5259 | 5260 | 5261 | 5262 | 5264 | 5265 | 5267 | 5268 | 5269 | 5270 | 5272 |
| | 5273 | 5304 | 5305 | 5306 | 5308 | 5310 | 5311 | 5333 | 5334 | 5335 | 5337 | 5339 |
| | 5340 | 5418 | 5419 | | | | | | | | | |
| FPI | 1321 # | 5263 | 5271 | 5309 | 5338 | | | | | | | |
| IOT | 1325 # | 6995 | 6996 | 7203 | 7524 | 7525 | 7526 | 7527 | 7535 | 7536 | 7537 | 7538 |
| | 7549 | 7550 | 7551 | 7552 | 7817 | 7818 | 8187 | | | | | |
| RD-PF | 1323 # | 2546 | 2551 | 2556 | 2561 | 2599 | 2604 | 2609 | 2614 | 2619 | 2624 | 2629 |
| | 2634 | 2641 | 2646 | 2651 | 2656 | 2661 | 2666 | 2671 | 2676 | 2806 | 2816 | 2826 |
| | 2831 | 2848 | 2858 | 2868 | 2878 | 2898 | 2909 | 2919 | 2929 | 4049 | 4063 | 4111 |
| | 4136 | 4303 | 4308 | | | | | | | | | |
| READ | 1315 # | 2549 | 2554 | 2559 | 2564 | 2583 | 2601 | 2602 | 2606 | 2607 | 2612 | 2617 |
| | 2622 | 2627 | 2632 | 2637 | 2643 | 2644 | 2648 | 2649 | 2654 | 2659 | 2664 | 2669 |
| | 2674 | 2679 | 2808 | 2809 | 2818 | 2819 | 2828 | 2829 | 2833 | 2834 | 2841 | 2850 |
| | 2851 | 2860 | 2861 | 2870 | 2871 | 2880 | 2881 | 2900 | 2901 | 2911 | 2912 | 2921 |
| | 2922 | 2931 | 2932 | 3185 | 3186 | 3187 | 3188 | 3189 | 3190 | 3201 | 3202 | 3203 |
| | 3204 | 3205 | 3206 | 3207 | 3208 | 3219 | 3220 | 3221 | 3222 | 3223 | 3224 | 3225 |
| | 3226 | 3236 | 3237 | 3238 | 3239 | 3240 | 3241 | 3242 | 3243 | 3358 | 3359 | 3360 |
| | 3361 | 3362 | 3363 | 3364 | 3365 | 3375 | 3376 | 3377 | 3378 | 3379 | 3380 | 3381 |
| | 3382 | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 | 3408 | 3409 | 3410 |
| | 3411 | 3412 | 3413 | 3414 | 3415 | 3646 | 4051 | 4052 | 4065 | 4066 | 4113 | 4114 |
| | 4138 | 4139 | 4305 | 4306 | 4310 | 4311 | 4681 | 4682 | 4683 | 4684 | 4685 | 5383 |
| SHIFT | 1319 # | 2951 | 2952 | 2953 | | | | | | | | |
| WRITE | 1316 # | 2548 | 2553 | 2558 | 2563 | 2611 | 2616 | 2621 | 2626 | 2631 | 2636 | 2653 |
| | 2658 | 2663 | 2668 | 2673 | 2678 | 2775 | 2843 | 2844 | | | | |
| (U) ACALU | 1218 # | | | | | | | | | | | |
| AC+N | 1220 # | 2172 | 2173 | 2180 | 2370 | 2382 | 2488 | 2771 | 2780 | 3021 | 3042 | 3086 |
| | 3116 | 4083 | 4095 | 4174 | 4176 | 4178 | 4186 | 4188 | 4195 | 4197 | 4209 | 4210 |
| | 4211 | 4212 | 4213 | 4214 | 4326 | 4410 | 4432 | 4436 | 4439 | 4453 | 4454 | 4456 |
| | 4459 | 4478 | 4480 | 4482 | 4483 | 4609 | 4610 | 4613 | 4614 | 4615 | 4618 | 4619 |
| | 4620 | 4626 | 4627 | 5546 | 5637 | 5690 | 5692 | 5698 | 5757 | 5765 | 5803 | 5810 |
| | 5954 | 5962 | 5972 | 5975 | 6005 | 6008 | 6021 | 6028 | 6035 | 6045 | 6083 | 6093 |
| | 6113 | 6128 | 6145 | 6160 | 6162 | 6188 | 6191 | 6193 | 6200 | 6201 | 6202 | 6204 |
| | 6210 | 6212 | 6215 | 6221 | 6223 | 6251 | 6253 | 6254 | 6258 | 6265 | 6266 | 6267 |
| | 6271 | 6272 | 6279 | 6281 | 6286 | 6292 | 6308 | 6319 | 6331 | 6340 | 6346 | 6357 |
| | 6360 | 6362 | 6366 | 6379 | 6385 | 6386 | 6509 | 6540 | 6545 | 6550 | 6587 | 6589 |
| | 6693 | 6694 | 6806 | 6813 | 6820 | 6839 | 6841 | 6848 | 6868 | 6889 | 6925 | 6927 |
| | 6933 | 6935 | 6939 | 6942 | 6947 | | | | | | | |
| B | 1219 # | | | | | | | | | | | |
| (D) ACDISP | 1345 # | 3515 | 6995 | 6996 | 7203 | | | | | | | |
| (U) ACN | 1221 # | 2370 | 2382 | 2488 | 2771 | 2780 | 3021 | 3042 | 3086 | 3116 | 4083 | 4095 |
| | 4174 | 4176 | 4178 | 4186 | 4188 | 4195 | 4197 | 4209 | 4210 | 4211 | 4212 | 4213 |
| | 4214 | 4326 | 4410 | 4432 | 4436 | 4439 | 4453 | 4454 | 4456 | 4459 | 4478 | 4480 |
| | 4482 | 4483 | 4609 | 4610 | 4613 | 4614 | 4615 | 4618 | 4619 | 4620 | 4626 | 4627 |
| | 5546 | 5637 | 5690 | 5692 | 5698 | 5757 | 5765 | 5803 | 5810 | 6292 | 6308 | 6319 |
| | 6366 | 6385 | | | | | | | | | | |
| BINO | 1228 # | 2172 | 6188 | 6210 | 6215 | 6223 | 6254 | 6265 | 6266 | 6267 | 6271 | 6279 |
| BIN1 | 1229 # | 2173 | 2180 | 6160 | 6162 | 6191 | 6193 | 6200 | 6201 | 6202 | 6204 | 6212 |
| | 6221 | 6251 | 6253 | 6258 | 6272 | 6281 | | | | | | |
| DLEN | 1225 # | 5954 | 5962 | 5975 | 6005 | 6008 | 6035 | 6045 | 6083 | 6093 | 6128 | 6286 |
| | 6331 | 6340 | 6346 | 6357 | 6360 | 6362 | 6379 | 6386 | 6693 | 6694 | 6806 | 6820 |
| | 6839 | 6939 | 6942 | 6947 | | | | | | | | |
| DSTP | 1226 # | 5972 | 6145 | 6540 | 6545 | 6589 | 6813 | 6868 | 6889 | 6933 | 6935 | |
| MARK | 1227 # | 6509 | 6587 | | | | | | | | | |
| SRCLN | 1223 # | | | | | | | | | | | |

| | | | | | | | | | | | | | |
|---------|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| | 4823 | 4901 | 4910 | 5753 | 5788 | | | | | | | | |
| A.EQV.B | 594 # | | | | | | | | | | | | |
| A.EQV.Q | 593 # | 4129 | 4132 | 4154 | 4155 | 4210 | 4212 | 4214 | 4483 | | | | |
| A.OR.B | 559 # | 2906 | 4091 | 4911 | 5064 | 5916 | 6015 | 6048 | 6087 | 6245 | 6676 | 7053 | |
| | 7057 | 7064 | 7095 | 7122 | 7139 | 7140 | 7340 | 7346 | 7488 | 7584 | 7619 | 8516 | |
| | 8730 | | | | | | | | | | | | |
| A.OR.Q | 558 # | 5218 | 5239 | 5587 | | | | | | | | | |
| A.XOR.B | 586 # | 5794 | 7294 | | | | | | | | | | |
| A.XOR.Q | 585 # | | | | | | | | | | | | |
| B | 561 # | 2159 | 2167 | 2169 | 2170 | 2173 | 2175 | 2182 | 2184 | 2375 | 2395 | 2494 | |
| | 2526 | 2532 | 2537 | 2540 | 2571 | 2590 | 2735 | 2742 | 2749 | 2756 | 2786 | 3067 | |
| | 3083 | 3139 | 3284 | 3328 | 3331 | 3334 | 3340 | 3343 | 3346 | 3405 | 3559 | 3607 | |
| | 3618 | 3658 | 3694 | 3700 | 3739 | 3745 | 3816 | 3952 | 3974 | 3996 | 4002 | 4013 | |
| | 4017 | 4033 | 4040 | 4121 | 4266 | 4270 | 4292 | 4296 | 4327 | 4333 | 4347 | 4352 | |
| | 4354 | 4359 | 4426 | 4428 | 4445 | 4461 | 4470 | 4531 | 4538 | 4559 | 4689 | 4698 | |
| | 4702 | 4717 | 4720 | 4736 | 4817 | 4883 | 4891 | 4899 | 4948 | 4961 | 4981 | 5013 | |
| | 5020 | 5284 | 5288 | 5371 | 5373 | 5405 | 5409 | 5442 | 5482 | 5501 | 5569 | 5668 | |
| | 5677 | 5726 | 5744 | 5782 | 5918 | 5921 | 5934 | 5936 | 5937 | 5967 | 5979 | 6013 | |
| | 6029 | 6055 | 6056 | 6057 | 6072 | 6114 | 6117 | 6119 | 6123 | 6136 | 6147 | 6155 | |
| | 6163 | 6294 | 6296 | 6339 | 6358 | 6371 | 6375 | 6377 | 6426 | 6433 | 6459 | 6473 | |
| | 6478 | 6500 | 6504 | 6556 | 6576 | 6590 | 6642 | 6650 | 6724 | 6745 | 6767 | 6799 | |
| | 6849 | 6851 | 6873 | 6913 | 6964 | 7066 | 7068 | 7069 | 7071 | 7074 | 7078 | 7155 | |
| | 7156 | 7222 | 7226 | 7230 | 7234 | 7238 | 7252 | 7263 | 7268 | 7284 | 7287 | 7303 | |
| | 7307 | 7309 | 7317 | 7318 | 7370 | 7382 | 7388 | 7395 | 7620 | 7662 | 7685 | 7691 | |
| | 7693 | 7727 | 7729 | 7838 | 7845 | 7910 | 7919 | 7925 | 7929 | 7930 | 7931 | 7933 | |
| | 7934 | 7935 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 | 7946 | 7947 | 7949 | 7950 | |
| | 7952 | 7953 | 7955 | 8197 | 8207 | 8208 | 8210 | 8211 | 8218 | 8228 | 8230 | 8232 | |
| | 8234 | 8238 | 8239 | 8537 | 8649 | 8685 | 8724 | 8755 | 8760 | 8796 | 8810 | | |
| B-.25 | 545 # | | | | | | | | | | | | |
| B-A-.25 | 543 # | 3419 | 3526 | 3535 | 3538 | 4274 | 4278 | 4548 | 4579 | 6009 | 6030 | 6134 | |
| | 6135 | 6246 | 6335 | 6387 | 6424 | 6817 | 6946 | 7260 | 7384 | 7397 | 7644 | 8758 | |
| | 8818 | | | | | | | | | | | | |
| D | 565 # | 2148 | 2152 | 2177 | 2178 | 2180 | 2218 | 2268 | 2338 | 2352 | 2356 | 2361 | |
| | 2365 | 2369 | 2370 | 2382 | 2383 | 2388 | 2393 | 2404 | 2410 | 2568 | 2587 | 2688 | |
| | 2690 | 2715 | 2717 | 2720 | 2721 | 2723 | 2726 | 2727 | 2729 | 2744 | 2746 | 2751 | |
| | 2753 | 2780 | 2964 | 2981 | 2987 | 2999 | 3002 | 3040 | 3086 | 3089 | 3268 | 3273 | |
| | 3386 | 3431 | 3434 | 3437 | 3443 | 3446 | 3449 | 3472 | 3548 | 3555 | 3572 | 3587 | |
| | 3592 | 3612 | 3616 | 3677 | 3688 | 3716 | 3722 | 3749 | 3758 | 3769 | 3771 | 3805 | |
| | 3811 | 3826 | 3827 | 3839 | 3840 | 3844 | 4001 | 4014 | 4103 | 4118 | 4127 | 4146 | |
| | 4150 | 4188 | 4193 | 4208 | 4211 | 4213 | 4215 | 4316 | 4325 | 4326 | 4408 | 4537 | |
| | 4691 | 4696 | 4710 | 4712 | 4716 | 4731 | 4764 | 4790 | 4791 | 4792 | 4793 | 4794 | |
| | 4810 | 4884 | 4885 | 4886 | 4887 | 4888 | 4936 | 4945 | 4952 | 4966 | 4968 | 4972 | |
| | 4976 | 4978 | 4982 | 5011 | 5016 | 5039 | 5052 | 5054 | 5092 | 5101 | 5105 | 5112 | |
| | 5156 | 5174 | 5175 | 5200 | 5205 | 5282 | 5315 | 5347 | 5411 | 5489 | 5505 | 5547 | |
| | 5557 | 5649 | 5700 | 5835 | 5911 | 5930 | 5939 | 5954 | 5957 | 5971 | 5972 | 5975 | |
| | 5997 | 6000 | 6005 | 6021 | 6022 | 6032 | 6042 | 6059 | 6060 | 6061 | 6062 | 6070 | |
| | 6076 | 6084 | 6093 | 6095 | 6113 | 6115 | 6122 | 6126 | 6139 | 6141 | 6145 | 6157 | |
| | 6160 | 6169 | 6188 | 6191 | 6193 | 6200 | 6210 | 6251 | 6253 | 6254 | 6259 | 6286 | |
| | 6291 | 6292 | 6293 | 6305 | 6314 | 6319 | 6331 | 6337 | 6347 | 6350 | 6351 | 6366 | |
| | 6378 | 6380 | 6381 | 6399 | 6402 | 6451 | 6456 | 6461 | 6462 | 6502 | 6509 | 6524 | |
| | 6540 | 6550 | 6559 | 6563 | 6578 | 6587 | 6589 | 6593 | 6614 | 6621 | 6623 | 6654 | |
| | 6670 | 6691 | 6693 | 6731 | 6756 | 6806 | 6811 | 6813 | 6820 | 6822 | 6839 | 6841 | |
| | 6842 | 6845 | 6868 | 6885 | 6887 | 6897 | 6912 | 6925 | 6929 | 6933 | 6939 | 6958 | |
| | 6963 | 6982 | 7002 | 7003 | 7035 | 7037 | 7049 | 7060 | 7083 | 7084 | 7092 | 7104 | |
| | 7151 | 7208 | 7210 | 7212 | 7214 | 7216 | 7221 | 7225 | 7229 | 7233 | 7237 | 7245 | |

| | | | | | | | | | | | | |
|---------|-------|------|------|------|------|------|------|------|------|------|------|------|
| | 7250 | 7254 | 7259 | 7265 | 7266 | 7277 | 7297 | 7301 | 7316 | 7322 | 7336 | 7349 |
| | 7359 | 7378 | 7380 | 7391 | 7393 | 7417 | 7443 | 7454 | 7470 | 7472 | 7491 | 7496 |
| | 7514 | 7546 | 7587 | 7654 | 7657 | 7659 | 7667 | 7681 | 7712 | 7715 | 7717 | 7826 |
| | 7833 | 7865 | 7912 | 7917 | 7921 | 7924 | 7936 | 7954 | 7956 | 7957 | 7959 | 8194 |
| | 8198 | 8200 | 8209 | 8215 | 8220 | 8498 | 8538 | 8539 | 8540 | 8546 | 8679 | 8688 |
| | 8725 | 8735 | 8741 | 8750 | 8754 | 8766 | 8770 | 8772 | 8774 | 8776 | 8778 | 8780 |
| | 8784 | 8808 | | | | | | | | | | |
| D+A | 539 # | 2288 | 2300 | 2315 | 2328 | 3486 | 3508 | 3563 | 3678 | 3724 | 3753 | 3817 |
| | 3832 | 4056 | 4083 | 4086 | 4748 | 4756 | 5034 | 5103 | 5115 | 5924 | 5927 | 5991 |
| | 6024 | 6079 | 6177 | 6202 | 6212 | 6215 | 6355 | 6373 | 6416 | 6425 | 6432 | 6434 |
| | 6553 | 6629 | 6736 | 6744 | 6818 | 6824 | 6892 | 6894 | 7298 | 7447 | 7492 | 7665 |
| | 7671 | 7673 | 8665 | 8675 | 8789 | | | | | | | |
| D+Q | 540 # | 4984 | 4992 | 5023 | 5042 | | | | | | | |
| D-.25 | 557 # | | | | | | | | | | | |
| D-A-.25 | 555 # | 3369 | 3500 | 4070 | 4095 | 4097 | 4100 | 5049 | 6360 | 6379 | | |
| D-Q-.25 | 556 # | | | | | | | | | | | |
| D.AND.A | 574 # | 2421 | 2423 | 2434 | 2456 | 2770 | 2813 | 2936 | 2967 | 3015 | 3275 | 3298 |
| | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 | 3585 | 3591 | 3605 | 3606 |
| | 3626 | 3627 | 3639 | 3641 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 |
| | 3938 | 3985 | 3986 | 3991 | 4036 | 4104 | 4174 | 4176 | 4357 | 4410 | 4432 | 4439 |
| | 4635 | 4770 | 4942 | 4949 | 5061 | 5224 | 5289 | 5293 | 5318 | 5351 | 5388 | 5392 |
| | 5412 | 5507 | 5546 | 5637 | 5672 | 5675 | 5692 | 5743 | 5746 | 5749 | 5752 | 5812 |
| | 5912 | 5914 | 5955 | 5958 | 6094 | 6096 | 6132 | 6158 | 6165 | 6178 | 6198 | 6240 |
| | 6256 | 6263 | 6288 | 6304 | 6323 | 6382 | 6396 | 6398 | 6430 | 6436 | 6452 | 6464 |
| | 6467 | 6487 | 6490 | 6506 | 6521 | 6637 | 6671 | 6674 | 6737 | 6751 | 6907 | 6917 |
| | 6984 | 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 | 7014 | 7015 |
| | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7052 | 7054 | 7056 | 7058 |
| | 7061 | 7063 | 7065 | 7086 | 7089 | 7107 | 7112 | 7115 | 7119 | 7134 | 7152 | 7153 |
| | 7180 | 7188 | 7197 | 7217 | 7239 | 7281 | 7331 | 7333 | 7334 | 7335 | 7337 | 7339 |
| | 7341 | 7343 | 7345 | 7347 | 7458 | 7498 | 7532 | 7589 | 7596 | 7602 | 7616 | 7648 |
| | 7650 | 7664 | 7793 | 7802 | 7804 | 7806 | 7808 | 7810 | 7812 | 8195 | 8199 | 8248 |
| | 8252 | 8491 | 8504 | 8511 | 8514 | 8530 | 8532 | 8680 | 8693 | 8697 | 8700 | 8705 |
| | 8711 | 8727 | 8752 | | | | | | | | | |
| D.AND.Q | 575 # | 3112 | 4510 | 5210 | 5216 | 5230 | 5236 | 5323 | 5632 | | | |
| D.EQV.A | 598 # | 2885 | | | | | | | | | | |
| D.EQV.Q | 599 # | | | | | | | | | | | |
| D.OR.A | 563 # | 2150 | 2419 | 2457 | 2865 | 3293 | 3988 | 3993 | 4105 | 4458 | 4543 | 5290 |
| | 5294 | 5319 | 5352 | 5413 | 5552 | 5608 | 5609 | 5616 | 5653 | 5705 | 5819 | 6170 |
| | 6257 | 6301 | 6310 | 6315 | 6325 | 6596 | 6600 | 6602 | 6663 | 6776 | 6780 | 6784 |
| | 6792 | 6808 | 6944 | 6953 | 7154 | 7267 | 7342 | 7420 | 7421 | 7422 | 7423 | 7424 |
| | 7425 | 7426 | 7560 | 7861 | 8191 | 8245 | 8249 | 8508 | 8512 | 8529 | 8531 | 8535 |
| | 8698 | 8702 | 8707 | 8714 | 8719 | 8792 | | | | | | |
| D.OR.Q | 564 # | 3114 | 4395 | | | | | | | | | |
| D.XOR.A | 590 # | 2187 | 2855 | 3290 | 4412 | 7456 | 7457 | 8258 | | | | |
| D.XOR.Q | 591 # | | | | | | | | | | | |
| Q | 560 # | 3116 | 4123 | 4131 | 4178 | 4186 | 4186 | 4197 | 4209 | 4351 | 4406 | 4423 |
| | 4429 | 4434 | 4438 | 4444 | 4452 | 4971 | 5133 | 5159 | 5247 | 5370 | 5375 | 5446 |
| | 5488 | 5512 | 5606 | 5613 | 5633 | 5645 | 5656 | 5715 | 5717 | 5761 | 5806 | 6103 |
| | 6142 | 6544 | | | | | | | | | | |
| Q-.25 | 544 # | | | | | | | | | | | |
| Q-A-.25 | 542 # | 4427 | 5032 | | | | | | | | | |
| Q-D-.25 | 548 # | 4975 | | | | | | | | | | |
| ZERO | 571 # | 2160 | 2160 | 2162 | 2165 | 2172 | 2174 | 2176 | 2193 | 2215 | 2399 | 2803 |
| | 2980 | 3021 | 3041 | 3270 | 3608 | 4222 | 4248 | 4318 | 4454 | 4532 | 4799 | 4812 |
| | 4818 | 4819 | 4822 | 4894 | 4895 | 4898 | 4954 | 5097 | 5139 | 5163 | 5249 | 5296 |

| | | | | | | | | | | | | |
|------------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 5367 | 5378 | 5390 | 5394 | 5396 | 5397 | 5410 | 5423 | 5515 | 5688 | 5716 | 5755 |
| | 5771 | 5799 | 5980 | 5982 | 6016 | 6035 | 6083 | 6088 | 6100 | 6111 | 6140 | 6261 |
| | 6279 | 6281 | 6299 | 6333 | 6389 | 6393 | 6397 | 6400 | 6455 | 6579 | 6617 | 6657 |
| | 6727 | 6802 | 7027 | 7030 | 7093 | 7127 | 7137 | 7181 | 7257 | 7332 | 7377 | 7402 |
| | 7403 | 7435 | 7438 | 7445 | 7459 | 7674 | 7719 | 7840 | 7843 | 7913 | 8244 | 8266 |
| | 8507 | 8527 | 8739 | 8745 | 8764 | 8782 | 8783 | 8814 | | | | |
| -A-.25 | 554 # | 2576 | 2946 | 4320 | 4398 | 4475 | 4517 | 4535 | 4542 | 4636 | 4640 | 4644 |
| | 4646 | 5279 | 5354 | 5358 | 5389 | 5498 | 5514 | 5543 | 5544 | 5695 | 5706 | 5707 |
| | 5775 | 5796 | 5802 | 6264 | 6388 | 7360 | | | | | | |
| -B-.25 | 553 # | 5696 | | | | | | | | | | |
| -D-.25 | 549 # | 4449 | 4609 | 4614 | 4619 | 4624 | 5995 | 6267 | 6938 | 6952 | | |
| -Q-.25 | 552 # | 4397 | 4448 | 4474 | 4515 | 4516 | 4533 | 5376 | 5481 | 5483 | 5485 | 5487 |
| | 5770 | | | | | | | | | | | |
| .NOT.A | 597 # | 2838 | 2875 | 2916 | 2926 | 3287 | 4090 | 4329 | 4534 | 4638 | 4645 | 5448 |
| | 5496 | 5511 | 5513 | 5694 | 5704 | 5773 | 5964 | 5966 | 6518 | 7362 | 7437 | 7647 |
| .NOT.A.AND.B | 578 # | 4125 | 5581 | 7059 | 7062 | 7338 | 7348 | 7518 | | | | |
| .NOT.A.AND.Q | 577 # | 5718 | | | | | | | | | | |
| .NOT.B | 596 # | 5801 | 6173 | | | | | | | | | |
| .NOT.D | 600 # | 2895 | 2905 | 4613 | 4618 | 4623 | 6012 | 6040 | 6044 | 6180 | 6184 | 6266 |
| | 6956 | | | | | | | | | | | |
| .NOT.D.AND.A | 582 # | 2823 | 3620 | 4021 | 5550 | 5652 | 5703 | 5820 | 6404 | 6772 | 6788 | 7050 |
| | 7076 | 7344 | 7562 | 8520 | 8695 | | | | | | | |
| .NOT.D.AND.Q | 583 # | | | | | | | | | | | |
| .NOT.Q | 595 # | 4908 | 5769 | 5983 | | | | | | | | |
| O+A | 538 # | 5836 | | | | | | | | | | |
| O+B | 537 # | | | | | | | | | | | |
| O+D | 541 # | | | | | | | | | | | |
| O+Q | 536 # | 5582 | 5729 | | | | | | | | | |
| (U) AD PARITY OK | 710 # | 2180 | 2356 | 2369 | 2370 | 2587 | 2688 | 2690 | 2721 | 2727 | 2780 | 2964 |
| | 2967 | 2981 | 3015 | 3472 | 3716 | 3749 | 3771 | 3827 | 3839 | 4146 | 4188 | 4325 |
| | 4716 | 5112 | 5200 | 5282 | 5315 | 5347 | 5411 | 5557 | 5954 | 5957 | 5972 | 5975 |
| | 6005 | 6021 | 6093 | 6095 | 6113 | 6145 | 6157 | 6160 | 6188 | 6191 | 6193 | 6200 |
| | 6210 | 6251 | 6253 | 6254 | 6286 | 6291 | 6292 | 6293 | 6314 | 6319 | 6331 | 6366 |
| | 6451 | 6509 | 6550 | 6587 | 6589 | 6693 | 6806 | 6813 | 6839 | 6841 | 6868 | 6925 |
| | 6933 | 6939 | 7546 | 7833 | 8221 | | | | | | | |
| (U) ADFLGS | 1114 # | 2577 | 3403 | 3419 | 3486 | 3500 | 4057 | 4071 | 4088 | 4098 | 4101 | 4228 |
| | 4639 | 4641 | | | | | | | | | | |
| (U) AREAD | 1148 # | 2320 | | | | | | | | | | |
| (U) B | 656 # | | | | | | | | | | | |
| AR | 660 # | 2157 | 2159 | 2174 | 2237 | 2241 | 2244 | 2256 | 2259 | 2262 | 2319 | 2352 |
| | 2356 | 2361 | 2369 | 2375 | 2388 | 2393 | 2395 | 2404 | 2413 | 2413 | 2416 | 2416 |
| | 2425 | 2426 | 2456 | 2457 | 2494 | 2520 | 2526 | 2532 | 2537 | 2540 | 2568 | 2571 |
| | 2576 | 2688 | 2690 | 2715 | 2717 | 2720 | 2721 | 2723 | 2726 | 2727 | 2729 | 2735 |
| | 2737 | 2739 | 2742 | 2744 | 2746 | 2749 | 2751 | 2753 | 2756 | 2758 | 2760 | 2803 |
| | 2813 | 2823 | 2838 | 2855 | 2865 | 2875 | 2885 | 2895 | 2906 | 2906 | 2916 | 2926 |
| | 2936 | 2946 | 2964 | 2967 | 2971 | 2981 | 2983 | 2987 | 2990 | 2994 | 2999 | 3002 |
| | 3004 | 3005 | 3007 | 3009 | 3010 | 3040 | 3041 | 3042 | 3067 | 3083 | 3139 | 3268 |
| | 3273 | 3287 | 3290 | 3293 | 3298 | 3299 | 3328 | 3331 | 3334 | 3340 | 3343 | 3346 |
| | 3369 | 3386 | 3388 | 3403 | 3403 | 3405 | 3419 | 3613 | 3613 | 3631 | 3634 | 3634 |
| | 3641 | 3769 | 3774 | 3779 | 3827 | 3985 | 4001 | 4002 | 4014 | 4017 | 4021 | 4030 |
| | 4033 | 4041 | 4056 | 4070 | 4086 | 4091 | 4091 | 4097 | 4100 | 4103 | 4123 | 4129 |
| | 4131 | 4132 | 4149 | 4150 | 4167 | 4215 | 4318 | 4320 | 4325 | 4327 | 4352 | 4374 |
| | 4408 | 4414 | 4418 | 4461 | 4465 | 4465 | 4468 | 4468 | 4472 | 4475 | 4478 | 4492 |
| | 4517 | 4531 | 4534 | 4535 | 4540 | 4545 | 4548 | 4551 | 4553 | 4555 | 4556 | 4556 |
| | 4557 | 4587 | 4623 | 4624 | 4628 | 4638 | 4640 | 4645 | 4646 | 4689 | 4698 | 4702 |

Cross Reference Listing

| | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|
| 4706 | 4717 | 4720 | 4729 | 4731 | 4735 | 4735 | 4736 | 4764 | 4802 | 4817 | 4819 |
| 4822 | 4823 | 4883 | 4884 | 4885 | 4886 | 4887 | 4888 | 4891 | 4899 | 4908 | 4909 |
| 4909 | 4911 | 4911 | 4966 | 4968 | 4978 | 4981 | 4982 | 5011 | 5020 | 5034 | 5044 |
| 5049 | 5052 | 5054 | 5058 | 5136 | 5160 | 5174 | 5175 | 5177 | 5248 | 5279 | 5286 |
| 5288 | 5289 | 5290 | 5299 | 5299 | 5325 | 5326 | 5347 | 5351 | 5352 | 5363 | 5368 |
| 5370 | 5371 | 5373 | 5375 | 5376 | 5401 | 5403 | 5405 | 5409 | 5411 | 5412 | 5413 |
| 5433 | 5438 | 5439 | 5442 | 5443 | 5443 | 5445 | 5447 | 5447 | 5480 | 5482 | 5484 |
| 5486 | 5489 | 5492 | 5496 | 5498 | 5500 | 5501 | 5502 | 5503 | 5504 | 5505 | 5508 |
| 5508 | 5511 | 5513 | 5514 | 5523 | 5526 | 5526 | 5528 | 5544 | 5571 | 5576 | 5576 |
| 5578 | 5583 | 5585 | 5629 | 5666 | 5670 | 5674 | 5677 | 5700 | 5703 | 5705 | 5707 |
| 5731 | 5732 | 5740 | 5744 | 5747 | 5750 | 5762 | 5765 | 5773 | 5775 | 5779 | 5782 |
| 5784 | 5786 | 5801 | 5807 | 5810 | 5832 | 5835 | 5836 | 5840 | 5879 | 5881 | 5883 |
| 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | 5921 | 5950 | 5950 | 5957 | 5964 | 5966 |
| 5967 | 5982 | 5983 | 5991 | 5995 | 6008 | 6009 | 6009 | 6012 | 6014 | 6014 | 6015 |
| 6015 | 6021 | 6028 | 6029 | 6039 | 6040 | 6042 | 6044 | 6045 | 6046 | 6046 | 6051 |
| 6051 | 6059 | 6098 | 6098 | 6099 | 6100 | 6113 | 6119 | 6126 | 6130 | 6130 | 6133 |
| 6133 | 6136 | 6140 | 6145 | 6147 | 6154 | 6155 | 6157 | 6165 | 6173 | 6177 | 6180 |
| 6184 | 6253 | 6256 | 6257 | 6258 | 6266 | 6267 | 6271 | 6291 | 6293 | 6296 | 6305 |
| 6314 | 6331 | 6345 | 6346 | 6347 | 6348 | 6348 | 6353 | 6371 | 6373 | 6375 | 6380 |
| 6384 | 6397 | 6399 | 6400 | 6402 | 6424 | 6427 | 6432 | 6438 | 6438 | 6449 | 6461 |
| 6462 | 6463 | 6464 | 6466 | 6467 | 6500 | 6520 | 6524 | 6525 | 6525 | 6545 | 6550 |
| 6552 | 6553 | 6556 | 6578 | 6579 | 6591 | 6593 | 6614 | 6617 | 6619 | 6619 | 6621 |
| 6629 | 6637 | 6641 | 6641 | 6642 | 6654 | 6657 | 6660 | 6663 | 6667 | 6670 | 6674 |
| 6691 | 6693 | 6694 | 6724 | 6730 | 6731 | 6736 | 6744 | 6745 | 6747 | 6753 | 6802 |
| 6806 | 6811 | 6813 | 6817 | 6818 | 6820 | 6822 | 6824 | 6839 | 6841 | 6844 | 6845 |
| 6846 | 6846 | 6848 | 6849 | 6851 | 6868 | 6873 | 6886 | 6887 | 6888 | 6888 | 6889 |
| 6897 | 6912 | 6913 | 6917 | 6925 | 6929 | 6930 | 6933 | 6938 | 6942 | 6946 | 6946 |
| 6947 | 6950 | 6950 | 6952 | 6953 | 6954 | 6956 | 6966 | 6966 | 6974 | 6974 | 7104 |
| 7107 | 7146 | 7148 | 7221 | 7222 | 7225 | 7226 | 7229 | 7233 | 7233 | 7234 | 7237 |
| 7238 | 7245 | 7249 | 7249 | 7250 | 7252 | 7254 | 7255 | 7255 | 7257 | 7259 | 7260 |
| 7260 | 7263 | 7265 | 7277 | 7287 | 7301 | 7316 | 7317 | 7318 | 7377 | 7378 | 7382 |
| 7384 | 7384 | 7388 | 7391 | 7395 | 7397 | 7397 | 7435 | 7439 | 7443 | 7446 | 7447 |
| 7449 | 7449 | 7454 | 7472 | 7482 | 7486 | 7488 | 7488 | 7543 | 7565 | 7590 | 7596 |
| 7602 | 7651 | 7654 | 7659 | 7665 | 7667 | 7671 | 7673 | 7674 | 7712 | 7721 | 7721 |
| 7826 | 7833 | 7860 | 7861 | 7934 | 7935 | 7936 | 7937 | 7954 | 7955 | 7956 | 8191 |
| 8201 | 8207 | 8495 | 8498 | 8501 | 8504 | 8516 | 8516 | 8546 | 8735 | 8741 | 8750 |
| 8754 | 8755 | 8766 | 8789 | 8816 | 8829 | 8829 | 8833 | | | | |
| 661 # | 2180 | 2184 | 2365 | 2370 | 2399 | 2423 | 2488 | 2770 | 2771 | 2780 | 2786 |
| 3678 | 3709 | 3711 | 3716 | 3724 | 3749 | 3753 | 3826 | 3829 | 3942 | 3944 | 3944 |
| 3952 | 3978 | 4025 | 4083 | 4095 | 4104 | 4105 | 4121 | 4122 | 4127 | 4151 | 4154 |
| 4155 | 4168 | 4168 | 4176 | 4186 | 4199 | 4199 | 4205 | 4208 | 4222 | 4228 | 4248 |
| 4267 | 4271 | 4275 | 4279 | 4285 | 4289 | 4293 | 4297 | 4339 | 4351 | 4357 | 4423 |
| 4429 | 4438 | 4532 | 4537 | 4543 | 4559 | 4579 | 4579 | 4582 | 4582 | 4593 | 4635 |
| 4636 | 4644 | 4712 | 4716 | 4903 | 4910 | 4936 | 4939 | 4942 | 4945 | 4948 | 5092 |
| 5141 | 5141 | 5166 | 5251 | 5285 | 5353 | 5354 | 5365 | 5543 | 5645 | 5657 | 5657 |
| 5692 | 5715 | 5790 | 5790 | 5792 | 5792 | 5794 | 5794 | 5796 | 5829 | 5973 | 5975 |
| 5993 | 6024 | 6030 | 6056 | 6061 | 6069 | 6069 | 6072 | 6074 | 6079 | 6093 | 6097 |
| 6117 | 6123 | 6128 | 6134 | 6160 | 6162 | 6191 | 6212 | 6218 | 6218 | 6221 | 6231 |
| 6231 | 6240 | 6251 | 6263 | 6264 | 6269 | 6272 | 6292 | 6294 | 6304 | 6308 | 6309 |
| 6309 | 6319 | 6337 | 6366 | 6377 | 6381 | 6383 | 6383 | 6385 | 6422 | 6430 | 6434 |
| 6436 | 6576 | 6756 | 6799 | 6867 | 6870 | 6943 | 6944 | 6963 | 6964 | 7052 | 7054 |
| 7092 | 7093 | 7281 | 7284 | 7292 | 7296 | 7298 | 7309 | 7445 | 7484 | 7484 | 7494 |
| 7494 | 7617 | 7619 | 7628 | 7634 | 7657 | 7715 | 7719 | 7731 | 7838 | 7843 | 7846 |
| 7910 | 7912 | 7917 | 7921 | 7924 | 7925 | 7929 | 7932 | 7939 | 7942 | 7945 | 7948 |
| 7951 | 7957 | 7959 | 8211 | 8279 | 8527 | 8538 | 8655 | 8655 | 8665 | 8670 | 8670 |

ARX

Cross Reference Listing

| | | | | | | | | | | | | | |
|-------|-------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | | 5228 | 5229 | 5243 | 5631 | 5721 | 5757 | 5803 | 7930 | | | | |
| | MASK | 667 # | 2148 | 2149 | 2150 | 3607 | 4125 | 5222 | 5223 | 5984 | 7919 | 7946 | |
| | ONE | 664 # | 2154 | 2173 | 2175 | 2782 | 6352 | 7941 | | | | | |
| | PC | 658 # | 2215 | 2228 | 2230 | 2275 | 2288 | 2294 | 3018 | 3337 | 3453 | 3456 | 3524 |
| | | 3525 | 3526 | 3535 | 3538 | 3551 | 3578 | 3593 | 3616 | 3706 | 3758 | 3817 | 3832 |
| | | 6246 | 6501 | 6501 | 7474 | 7644 | 7729 | 7931 | 8758 | 8768 | 8768 | 8818 | |
| | PI | 669 # | 2193 | 7332 | 7338 | 7338 | 7340 | 7340 | 7342 | 7344 | 7346 | 7346 | 7348 |
| | | 7348 | 7420 | 7421 | 7422 | 7423 | 7424 | 7425 | 7426 | 7517 | 7518 | 7518 | 7949 |
| | TO | 671 # | 3592 | 4145 | 4153 | 4185 | 4188 | 4195 | 4434 | 4440 | 4444 | 4446 | 4448 |
| | | 4453 | 4458 | 4459 | 5557 | 5562 | 5569 | 5570 | 5597 | 5599 | 5600 | 5601 | 5604 |
| | | 5612 | 5615 | 5617 | 5644 | 5717 | 5723 | 6350 | 6416 | 6425 | 6426 | 7349 | 7359 |
| | | 7360 | 7361 | 7361 | 7491 | 7681 | 7685 | 7691 | 7693 | 7952 | | | |
| | T1 | 672 # | 2190 | 2201 | 2218 | 2247 | 2265 | 2516 | 3579 | 4394 | 4405 | 4412 | 4426 |
| | | 4428 | 4445 | 4470 | 4552 | 5211 | 5219 | 5224 | 5231 | 5242 | 5581 | 5581 | 5606 |
| | | 5613 | 5656 | 5668 | 5688 | 5694 | 5704 | 5726 | 6420 | 6433 | 7468 | 7500 | 7845 |
| | | 7953 | 8749 | | | | | | | | | | |
| | UBR | 666 # | 2160 | 7112 | 7122 | 7122 | 7134 | 7137 | 7139 | 7139 | 7140 | 7140 | 7944 |
| | XWD1 | 670 # | 2152 | 7950 | | | | | | | | | |
| (D) B | | 1327 # | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 | 3185 | 3186 | 3187 | 3188 | 3189 |
| | | 3190 | 3192 | 3193 | 3194 | 3195 | 3196 | 3197 | 3198 | 3199 | 3201 | 3202 | 3203 |
| | | 3204 | 3205 | 3206 | 3207 | 3208 | 3210 | 3211 | 3212 | 3213 | 3214 | 3215 | 3216 |
| | | 3217 | 3219 | 3220 | 3221 | 3222 | 3223 | 3224 | 3225 | 3226 | 3227 | 3228 | 3229 |
| | | 3230 | 3231 | 3232 | 3233 | 3234 | 3236 | 3237 | 3238 | 3239 | 3240 | 3241 | 3242 |
| | | 3243 | 3349 | 3350 | 3351 | 3352 | 3353 | 3354 | 3355 | 3356 | 3358 | 3359 | 3360 |
| | | 3361 | 3362 | 3363 | 3364 | 3365 | 3375 | 3376 | 3377 | 3378 | 3379 | 3380 | 3381 |
| | | 3382 | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 | 3408 | 3409 | 3410 |
| | | 3411 | 3412 | 3413 | 3414 | 3415 | 3461 | 3462 | 3463 | 3464 | 3465 | 3466 | 3467 |
| | | 3468 | 3475 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 | 3489 | 3490 | 3491 |
| | | 3492 | 3493 | 3494 | 3495 | 3496 | 3503 | 3504 | 3668 | 3669 | 3670 | 3765 | 3851 |
| | | 3852 | 3853 | 3854 | 3855 | 3856 | 3857 | 3858 | 3908 | 3909 | 3910 | 3911 | 3912 |
| | | 3913 | 5848 | 5849 | 5850 | 5851 | 5852 | 5853 | 5854 | 5856 | 5857 | 5858 | 5859 |
| | | 5861 | 5862 | 5863 | 5864 | 5869 | 5870 | 5871 | 5872 | 5873 | 7524 | 7525 | 7526 |
| | | 7527 | 7535 | 7536 | 7537 | 7538 | 7549 | 7550 | 7551 | 7552 | 7736 | 7737 | 7738 |
| | | 7740 | 7741 | 7743 | 7744 | 7746 | 7747 | 7748 | 7749 | 7750 | 7751 | 7752 | 7753 |
| | | 7755 | 7756 | 7757 | 7758 | 7759 | 7760 | 7761 | 7762 | 7764 | 7765 | 7766 | 7767 |
| | | 7768 | 7769 | 7770 | 7771 | 7773 | 7774 | 7775 | 7776 | 7777 | 7778 | 7779 | 7780 |
| | | 7782 | 7783 | 7784 | 7785 | 7786 | 7787 | 7788 | 7789 | | | | |
| | AC | 1331 # | 2546 | 2547 | 2551 | 2552 | 2556 | 2557 | 2561 | 2562 | 2583 | 2599 | 2600 |
| | | 2604 | 2605 | 2609 | 2610 | 2614 | 2615 | 2619 | 2620 | 2624 | 2625 | 2629 | 2630 |
| | | 2634 | 2635 | 2641 | 2642 | 2646 | 2647 | 2651 | 2652 | 2656 | 2657 | 2661 | 2662 |
| | | 2666 | 2667 | 2671 | 2672 | 2676 | 2677 | 2766 | 2796 | 2797 | 2806 | 2807 | 2816 |
| | | 2817 | 2826 | 2827 | 2831 | 2832 | 2848 | 2849 | 2858 | 2859 | 2868 | 2869 | 2878 |
| | | 2879 | 2888 | 2889 | 2898 | 2899 | 2909 | 2910 | 2919 | 2920 | 2929 | 2930 | 2939 |
| | | 2940 | 4049 | 4050 | 4063 | 4064 | 4111 | 4112 | 4681 | 8187 | | | |
| | BOTH | 1333 # | 2799 | 2809 | 2819 | 2829 | 2834 | 2851 | 2861 | 2871 | 2881 | 2891 | 2901 |
| | | 2912 | 2922 | 2932 | 2942 | 4052 | 4066 | 4114 | | | | | |
| | DBLAC | 1329 # | 2765 | 4078 | 4079 | 4136 | 4137 | 4162 | 4303 | 4304 | 4308 | 4309 | 4380 |
| | | 5622 | 5683 | | | | | | | | | | |
| | DBLB | 1330 # | 4139 | 4306 | 4311 | | | | | | | | |
| | MEM | 1332 # | 2548 | 2553 | 2558 | 2563 | 2601 | 2606 | 2611 | 2616 | 2621 | 2626 | 2631 |
| | | 2636 | 2643 | 2648 | 2653 | 2658 | 2663 | 2668 | 2673 | 2678 | 2798 | 2808 | 2818 |
| | | 2828 | 2833 | 2843 | 2844 | 2850 | 2860 | 2870 | 2880 | 2880 | 2900 | 2911 | 2921 |
| | | 2931 | 2941 | 4051 | 4065 | 4113 | 4138 | 4305 | 4310 | 7203 | | | |
| | SELF | 1328 # | 2549 | 2554 | 2559 | 2564 | 2602 | 2607 | 2612 | 2617 | 2622 | 2627 | 2632 |
| | | 2637 | 2644 | 2649 | 2654 | 2659 | 2664 | 2669 | 2674 | 2679 | | | |

| | | | | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| (U) BWRITE | 1157 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 |
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 4057 | 4071 | 4123 |
| | 4129 | 4152 | 4154 | 5443 | 5447 | 5493 | 5515 | 8517 | 8817 | | | |
| (U) BYTE | 816 # | | | | | | | | | | | |
| BYTE1 | 817 # | 4691 | 4696 | 4710 | 4731 | 4884 | 6032 | 6042 | 6524 | 6670 | 6842 | 6845 |
| | 6885 | 6887 | 6958 | | | | | | | | | |
| BYTE2 | 818 # | 4885 | | | | | | | | | | |
| BYTE3 | 819 # | 4886 | | | | | | | | | | |
| BYTE4 | 820 # | 4887 | | | | | | | | | | |
| BYTE5 | 821 # | 3040 | 4790 | 4791 | 4792 | 4793 | 4794 | 4888 | 6654 | | | |
| (U) CALL | 968 # | 2155 | 2181 | 2193 | 2398 | 2770 | 2780 | 3550 | 3589 | 3591 | 3592 | 3606 |
| | 3630 | 3633 | 3828 | 3831 | 3945 | 3976 | 3979 | 4013 | 4015 | 4120 | 4148 | 4175 |
| | 4184 | 4194 | 4204 | 4227 | 4337 | 4358 | 4413 | 4419 | 4433 | 4437 | 4443 | 4489 |
| | 4513 | 4691 | 4696 | 4700 | 4705 | 4710 | 4722 | 4970 | 4980 | 5014 | 5019 | 5041 |
| | 5096 | 5112 | 5114 | 5201 | 5213 | 5215 | 5233 | 5235 | 5322 | 5364 | 5367 | 5372 |
| | 5374 | 5482 | 5484 | 5486 | 5501 | 5502 | 5503 | 5562 | 5570 | 5598 | 5638 | 5643 |
| | 5655 | 5662 | 5669 | 5688 | 5712 | 5716 | 5720 | 5745 | 5748 | 5751 | 5783 | 5785 |
| | 5787 | 5793 | 5913 | 5953 | 5956 | 5960 | 5992 | 5998 | 6023 | 6047 | 6058 | 6073 |
| | 6077 | 6084 | 6094 | 6096 | 6102 | 6116 | 6125 | 6146 | 6161 | 6164 | 6177 | 6192 |
| | 6203 | 6211 | 6214 | 6217 | 6220 | 6228 | 6233 | 6260 | 6295 | 6304 | 6320 | 6332 |
| | 6359 | 6370 | 6378 | 6395 | 6421 | 6450 | 6453 | 6458 | 6533 | 6541 | 6551 | 6554 |
| | 6564 | 6577 | 6592 | 6599 | 6620 | 6624 | 6692 | 6726 | 6750 | 6850 | 6869 | 6874 |
| | 6926 | 6934 | 6945 | 7187 | 7256 | 7280 | 7283 | 7302 | 7349 | 7379 | 7392 | 7441 |
| | 7471 | 7473 | 7489 | 7493 | 7531 | 7542 | 7557 | 7576 | 7585 | 7614 | 7621 | 7646 |
| | 7839 | 7841 | 7847 | 7863 | 7911 | 7913 | 7930 | 7933 | 7935 | 7937 | 7940 | 7943 |
| | 7946 | 7949 | 7952 | 7954 | 8726 | 8737 | 8797 | | | | | |
| (U) CHKL | 721 # | 2268 | 2338 | 2352 | 2361 | 2365 | 2382 | 2393 | 2404 | 2410 | 2423 | 2494 |
| | 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | 3086 | 3405 | 3548 | 3572 | 3587 | 3612 |
| | 3616 | 3677 | 3694 | 3700 | 3722 | 3739 | 3745 | 3758 | 3816 | 3844 | 3974 | 4001 |
| | 4013 | 4017 | 4040 | 4118 | 4211 | 4213 | 4316 | 4326 | 4410 | 4432 | 4439 | 4736 |
| | 4764 | 4770 | 4891 | 5156 | 5205 | 5911 | 5930 | 6350 | 6540 | 6590 | 6593 | 6621 |
| | 6897 | 6912 | 6982 | 7104 | 7221 | 7225 | 7229 | 7233 | 7237 | 7277 | 7307 | 7309 |
| | 7316 | 7370 | 7454 | 7496 | 7620 | 7667 | 7712 | 7715 | 7717 | 7727 | 7729 | 7826 |
| | 7845 | 7865 | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 |
| | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7955 | 8679 | 8796 | 8810 | | |
| (U) CHKR | 728 # | 2268 | 2338 | 2352 | 2361 | 2365 | 2382 | 2393 | 2404 | 2410 | 2423 | 2494 |
| | 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | 3086 | 3405 | 3548 | 3572 | 3587 | 3612 |
| | 3616 | 3677 | 3694 | 3700 | 3722 | 3739 | 3745 | 3758 | 3816 | 3844 | 3974 | 4001 |
| | 4013 | 4017 | 4040 | 4118 | 4211 | 4213 | 4316 | 4326 | 4410 | 4432 | 4439 | 4736 |
| | 4764 | 4770 | 4891 | 5156 | 5205 | 5911 | 5930 | 6350 | 6540 | 6590 | 6593 | 6621 |
| | 6897 | 6912 | 6982 | 7104 | 7221 | 7225 | 7229 | 7233 | 7237 | 7277 | 7307 | 7309 |
| | 7316 | 7370 | 7454 | 7496 | 7620 | 7667 | 7712 | 7715 | 7717 | 7727 | 7729 | 7826 |
| | 7845 | 7865 | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 |
| | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7955 | 8679 | 8796 | 8810 | | |
| (U) CLKL | 717 # | 2228 | 2230 | 2275 | 2288 | 2289 | 2295 | 2300 | 2315 | 2316 | 2328 | 2339 |
| | 2362 | 2690 | 2721 | 2729 | 2751 | 2753 | 3337 | 3453 | 3456 | 3524 | 3525 | 3526 |
| | 3535 | 3538 | 3551 | 3563 | 3593 | 3617 | 3631 | 3759 | 3818 | 3830 | 3834 | 3978 |
| | 3990 | 3995 | 4025 | 4041 | 4092 | 4735 | 4765 | 4974 | 5033 | 5056 | 5105 | 5117 |
| | 5139 | 5143 | 5148 | 5163 | 5166 | 5176 | 5228 | 5229 | 5249 | 5251 | 5324 | 5587 |
| | 5924 | 5927 | 5930 | 5971 | 5997 | 6000 | 6016 | 6022 | 6051 | 6070 | 6076 | 6084 |
| | 6088 | 6099 | 6111 | 6115 | 6122 | 6141 | 6169 | 6246 | 6261 | 6299 | 6334 | 6351 |
| | 6378 | 6389 | 6393 | 6449 | 6455 | 6502 | 6559 | 6563 | 6598 | 6623 | 6664 | 6669 |
| | 6727 | 6846 | 6888 | 6897 | 7051 | 7077 | 7091 | 7138 | 7152 | 7154 | 7198 | 7278 |
| | 7283 | 7308 | 7342 | 7344 | 7346 | 7348 | 7360 | 7420 | 7421 | 7422 | 7423 | 7424 |
| | 7425 | 7426 | 7448 | 7518 | 7644 | 7665 | 7665 | 7846 | 7862 | 7932 | 7939 | 7942 |

| | | | | | | | | | | | | |
|---------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 7945 | 7948 | 7951 | 8200 | 8507 | 8515 | 8758 | 8764 | 8770 | 8772 | 8774 | 8776 |
| (U) CLKR | 8778 | 8780 | 8782 | 8783 | 8784 | 8814 | 8833 | | | | | |
| | 724 # | 2419 | 2421 | 2434 | 2456 | 2457 | 2688 | 2723 | 2727 | 2744 | 2746 | 2770 |
| | 3113 | 3115 | 3386 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 | 3585 |
| | 3605 | 3626 | 3639 | 3770 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 |
| | 3940 | 4022 | 4038 | 4103 | 4104 | 4105 | 4127 | 4150 | 4357 | 4396 | 4458 | 4512 |
| | 4635 | 4812 | 4947 | 4955 | 5018 | 5063 | 5098 | 5102 | 5289 | 5290 | 5293 | 5294 |
| | 5318 | 5319 | 5351 | 5352 | 5412 | 5413 | 5489 | 5505 | 5608 | 5609 | 5616 | 5632 |
| | 5754 | 5789 | 5819 | 5820 | 5835 | 5917 | 5939 | 6166 | 6171 | 6240 | 6256 | 6257 |
| | 6263 | 6302 | 6304 | 6311 | 6317 | 6326 | 6345 | 6397 | 6400 | 6405 | 6430 | 6436 |
| | 6507 | 6601 | 6603 | 6673 | 6677 | 6731 | 6773 | 6777 | 6781 | 6785 | 6789 | 6793 |
| | 6809 | 6907 | 6919 | 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 |
| | 7014 | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7045 | 7085 |
| | 7088 | 7109 | 7114 | 7124 | 7129 | 7136 | 7141 | 7182 | 7189 | 7217 | 7239 | 7250 |
| | 7334 | 7336 | 7338 | 7340 | 7472 | 7674 | 7719 | 7793 | 7802 | 7804 | 7806 | 7808 |
| | 7810 | 7812 | 8193 | 8196 | 8247 | 8251 | 8254 | 8259 | 8506 | 8510 | 8513 | 8524 |
| | 8699 | 8704 | 8709 | 8715 | 8729 | 8792 | | | | | | |
| (U) CLRFPD | 1086 # | 3689 | 3806 | 3812 | 4206 | 4706 | 4723 | 6389 | | | | |
| (D) COND FUNC | 1350 # | 2548 | 2549 | 2553 | 2554 | 2558 | 2559 | 2563 | 2564 | 2601 | 2602 | 2606 |
| | 2607 | 2611 | 2612 | 2616 | 2617 | 2621 | 2622 | 2626 | 2627 | 2631 | 2632 | 2636 |
| | 2637 | 2643 | 2644 | 2648 | 2649 | 2653 | 2654 | 2658 | 2659 | 2663 | 2664 | 2668 |
| | 2669 | 2673 | 2674 | 2678 | 2679 | 2798 | 2799 | 2808 | 2809 | 2818 | 2819 | 2828 |
| | 2829 | 2833 | 2834 | 2843 | 2844 | 2850 | 2851 | 2860 | 2861 | 2870 | 2871 | 2880 |
| | 2881 | 2890 | 2891 | 2900 | 2901 | 2911 | 2912 | 2921 | 2922 | 2931 | 2932 | 2941 |
| | 2942 | 4051 | 4052 | 4065 | 4066 | 4113 | 4114 | 4138 | 4139 | 4305 | 4306 | 4310 |
| | 4311 | 5260 | 5261 | 5264 | 5265 | 5268 | 5269 | 5272 | 5273 | 5305 | 5306 | 5310 |
| | 5311 | 5334 | 5335 | 5339 | 5340 | 7203 | | | | | | |
| (U) CRY38 | 951 # | 2576 | 2946 | 3369 | 3419 | 3500 | 3526 | 3535 | 3538 | 4070 | 4087 | 4095 |
| | 4100 | 4274 | 4278 | 4320 | 4364 | 4397 | 4420 | 4421 | 4427 | 4448 | 4474 | 4515 |
| | 4516 | 4517 | 4533 | 4535 | 4542 | 4548 | 4579 | 4609 | 4614 | 4619 | 4624 | 4636 |
| | 4640 | 4644 | 4646 | 4975 | 5032 | 5049 | 5099 | 5118 | 5136 | 5160 | 5248 | 5279 |
| | 5354 | 5357 | 5358 | 5365 | 5376 | 5389 | 5481 | 5483 | 5485 | 5487 | 5498 | 5514 |
| | 5543 | 5582 | 5695 | 5706 | 5713 | 5729 | 5770 | 5775 | 5796 | 5802 | 5836 | 5962 |
| | 5995 | 6009 | 6030 | 6097 | 6133 | 6134 | 6135 | 6182 | 6246 | 6264 | 6267 | 6335 |
| | 6352 | 6360 | 6379 | 6387 | 6388 | 6422 | 6424 | 6427 | 6475 | 6817 | 6938 | 6946 |
| | 6952 | 7260 | 7360 | 7384 | 7397 | 7644 | 8651 | 8661 | 8758 | 8818 | | |
| (U) DBM | 696 # | | | | | | | | | | | |
| APR FLAGS | 699 # | 7002 | 7003 | 7060 | 7092 | 7266 | | | | | | |
| BYTES | 700 # | | | | | | | | | | | |
| DP | 702 # | 3040 | 4691 | 4696 | 4710 | 4731 | 4790 | 4791 | 4792 | 4793 | 4794 | 4884 |
| | 4885 | 4886 | 4887 | 4888 | 6032 | 6042 | 6524 | 6654 | 6670 | 6842 | 6845 | 6885 |
| | 6887 | 6958 | | | | | | | | | | |
| DP SWAP | 703 # | 2388 | 2568 | 2715 | 2717 | 2720 | 2723 | 2726 | 2729 | 3268 | 3273 | 3592 |
| | 3769 | 3826 | 3840 | 4810 | 4936 | 4952 | 5052 | 5092 | 5101 | 5174 | 6399 | 6456 |
| | 6461 | 6462 | 6756 | 7084 | 7336 | 7349 | 7359 | 7359 | 8688 | | | |
| EXP | 701 # | 5489 | 5505 | 5835 | 7291 | 7292 | 7293 | | | | | |
| MEM | 705 # | 2160 | 2267 | 2338 | 2351 | 2360 | 2364 | 2392 | 2403 | 2409 | 2422 | 3547 |
| | 3571 | 3586 | 3608 | 3612 | 3616 | 3676 | 3721 | 3758 | 3843 | 4000 | 4763 | 4769 |
| | 5155 | 5204 | 5911 | 5930 | 6350 | 6593 | 6621 | 6897 | 6911 | 6967 | 6975 | 6982 |
| | 7103 | 7221 | 7225 | 7229 | 7233 | 7237 | 7276 | 7316 | 7438 | 7442 | 7454 | 7490 |
| | 7496 | 7586 | 7667 | 7712 | 7715 | 7717 | 7825 | 7840 | 7864 | 7913 | 8244 | 8678 |
| | 8739 | 8745 | | | | | | | | | | |
| PF DISP | 698 # | 8214 | | | | | | | | | | |
| SCAD DIAG | 697 # | | | | | | | | | | | |
| VMA | 704 # | 7681 | 7924 | 8194 | 8209 | 8725 | | | | | | |

#

| | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|
| 706 # | 2148 | 2150 | 2152 | 2154 | 2157 | 2177 | 2178 | 2187 | 2190 | 2201 | 2218 |
| 2237 | 2241 | 2244 | 2247 | 2256 | 2259 | 2262 | 2265 | 2419 | 2421 | 2434 | 2456 |
| 2457 | 2516 | 2744 | 2746 | 2751 | 2753 | 2770 | 3041 | 3112 | 3114 | 3528 | 3537 |
| 3540 | 3542 | 3543 | 3544 | 3577 | 3579 | 3582 | 3585 | 3591 | 3605 | 3606 | 3620 |
| 3626 | 3639 | 3724 | 3753 | 3817 | 3832 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 |
| 3930 | 3932 | 3938 | 3942 | 3982 | 3986 | 3988 | 3991 | 3993 | 4021 | 4030 | 4036 |
| 4104 | 4105 | 4357 | 4394 | 4395 | 4405 | 4412 | 4458 | 4510 | 4537 | 4543 | 4635 |
| 4800 | 4942 | 4945 | 4949 | 4968 | 4972 | 4978 | 4982 | 5011 | 5016 | 5049 | 5054 |
| 5061 | 5105 | 5115 | 5211 | 5216 | 5224 | 5231 | 5236 | 5289 | 5290 | 5293 | 5294 |
| 5318 | 5319 | 5323 | 5351 | 5352 | 5388 | 5392 | 5412 | 5413 | 5507 | 5550 | 5552 |
| 5608 | 5609 | 5616 | 5632 | 5652 | 5653 | 5672 | 5675 | 5703 | 5705 | 5743 | 5746 |
| 5749 | 5752 | 5812 | 5819 | 5820 | 5879 | 5881 | 5883 | 5885 | 5887 | 5889 | 5891 |
| 5893 | 5895 | 5912 | 5914 | 5939 | 5955 | 5958 | 5971 | 5997 | 6000 | 6022 | 6070 |
| 6076 | 6084 | 6094 | 6096 | 6115 | 6122 | 6141 | 6154 | 6158 | 6165 | 6169 | 6170 |
| 6178 | 6182 | 6198 | 6240 | 6256 | 6257 | 6263 | 6269 | 6288 | 6301 | 6304 | 6310 |
| 6315 | 6323 | 6325 | 6351 | 6378 | 6382 | 6396 | 6398 | 6404 | 6416 | 6425 | 6430 |
| 6436 | 6452 | 6464 | 6467 | 6475 | 6487 | 6490 | 6502 | 6506 | 6521 | 6559 | 6563 |
| 6579 | 6591 | 6596 | 6600 | 6602 | 6617 | 6623 | 6637 | 6657 | 6663 | 6671 | 6674 |
| 6751 | 6772 | 6776 | 6780 | 6784 | 6788 | 6792 | 6802 | 6808 | 6907 | 6917 | 6984 |
| 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 | 7014 | 7015 | 7018 |
| 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7027 | 7030 | 7035 | 7037 | 7050 |
| 7052 | 7054 | 7056 | 7058 | 7061 | 7063 | 7065 | 7076 | 7086 | 7089 | 7093 | 7107 |
| 7112 | 7115 | 7119 | 7134 | 7152 | 7153 | 7154 | 7180 | 7188 | 7197 | 7217 | 7239 |
| 7248 | 7267 | 7281 | 7331 | 7333 | 7334 | 7335 | 7337 | 7339 | 7341 | 7342 | 7343 |
| 7344 | 7345 | 7347 | 7378 | 7380 | 7391 | 7393 | 7420 | 7421 | 7422 | 7423 | 7424 |
| 7425 | 7426 | 7427 | 7428 | 7429 | 7430 | 7431 | 7432 | 7433 | 7435 | 7439 | 7447 |
| 7456 | 7457 | 7468 | 7472 | 7486 | 7492 | 7498 | 7500 | 7514 | 7578 | 7581 | 7589 |
| 7596 | 7602 | 7616 | 7617 | 7628 | 7634 | 7648 | 7650 | 7664 | 7707 | 7793 | 7802 |
| 7804 | 7806 | 7808 | 7810 | 7812 | 7860 | 7861 | 8191 | 8195 | 8199 | 8200 | 8224 |
| 8226 | 8240 | 8245 | 8248 | 8249 | 8252 | 8258 | 8266 | 8491 | 8504 | 8508 | 8511 |
| 8512 | 8514 | 8520 | 8527 | 8529 | 8530 | 8531 | 8532 | 8535 | 8651 | 8661 | 8665 |
| 8675 | 8680 | 8693 | 8695 | 8697 | 8698 | 8700 | 8702 | 8705 | 8707 | 8711 | 8714 |
| 8719 | 8727 | 8749 | 8752 | 8770 | 8772 | 8774 | 8776 | 8778 | 8780 | 8784 | 8789 |

8792

(U) DBUS

DBM

| | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|
| 687 # | | | | | | | | | | | |
| 693 # | 2148 | 2150 | 2152 | 2154 | 2157 | 2160 | 2177 | 2178 | 2187 | 2190 | 2201 |
| 2218 | 2237 | 2241 | 2244 | 2247 | 2256 | 2259 | 2262 | 2265 | 2267 | 2268 | 2338 |
| 2338 | 2351 | 2352 | 2360 | 2361 | 2364 | 2365 | 2388 | 2392 | 2393 | 2403 | 2404 |
| 2409 | 2410 | 2419 | 2421 | 2422 | 2423 | 2434 | 2456 | 2457 | 2516 | 2568 | 2715 |
| 2717 | 2720 | 2723 | 2726 | 2729 | 2744 | 2746 | 2751 | 2753 | 2770 | 3040 | 3041 |
| 3112 | 3114 | 3268 | 3273 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3547 | 3548 |
| 3571 | 3572 | 3577 | 3579 | 3582 | 3585 | 3586 | 3587 | 3591 | 3592 | 3605 | 3606 |
| 3608 | 3612 | 3612 | 3616 | 3616 | 3620 | 3626 | 3639 | 3676 | 3677 | 3721 | 3722 |
| 3724 | 3753 | 3758 | 3758 | 3769 | 3817 | 3826 | 3832 | 3840 | 3843 | 3844 | 3918 |
| 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 | 3938 | 3942 | 3982 | 3986 | 3988 |
| 3991 | 3993 | 4000 | 4001 | 4021 | 4030 | 4036 | 4104 | 4105 | 4357 | 4394 | 4395 |
| 4405 | 4412 | 4458 | 4510 | 4537 | 4543 | 4635 | 4691 | 4696 | 4710 | 4731 | 4763 |
| 4764 | 4769 | 4770 | 4790 | 4791 | 4792 | 4793 | 4794 | 4800 | 4810 | 4884 | 4885 |
| 4886 | 4887 | 4888 | 4936 | 4942 | 4945 | 4949 | 4952 | 4968 | 4972 | 4978 | 4982 |
| 5011 | 5016 | 5049 | 5052 | 5054 | 5061 | 5092 | 5101 | 5105 | 5115 | 5155 | 5156 |
| 5174 | 5204 | 5205 | 5210 | 5216 | 5224 | 5230 | 5236 | 5289 | 5290 | 5293 | 5294 |
| 5318 | 5319 | 5323 | 5351 | 5352 | 5388 | 5392 | 5412 | 5413 | 5489 | 5505 | 5507 |
| 5550 | 5552 | 5608 | 5609 | 5616 | 5632 | 5652 | 5653 | 5672 | 5675 | 5703 | 5705 |
| 5743 | 5746 | 5749 | 5752 | 5812 | 5819 | 5820 | 5835 | 5879 | 5881 | 5883 | 5885 |
| 5887 | 5889 | 5891 | 5893 | 5895 | 5911 | 5911 | 5912 | 5914 | 5930 | 5930 | 5939 |

Cross Reference Listing

| | | | | | | | | | | | | |
|----------|-------|------|------|------|------|------|------|------|------|------|------|------|
| | 5955 | 5958 | 5971 | 5997 | 6000 | 6022 | 6032 | 6042 | 6070 | 6076 | 6084 | 6094 |
| | 6096 | 6115 | 6122 | 6141 | 6154 | 6158 | 6165 | 6169 | 6170 | 6178 | 6182 | 6198 |
| | 6240 | 6256 | 6257 | 6263 | 6269 | 6288 | 6301 | 6304 | 6310 | 6315 | 6323 | 6325 |
| | 6350 | 6350 | 6351 | 6378 | 6382 | 6396 | 6398 | 6399 | 6404 | 6416 | 6425 | 6430 |
| | 6436 | 6452 | 6456 | 6461 | 6462 | 6464 | 6467 | 6475 | 6487 | 6490 | 6502 | 6506 |
| | 6521 | 6524 | 6559 | 6563 | 6579 | 6591 | 6593 | 6593 | 6596 | 6600 | 6602 | 6617 |
| | 6621 | 6621 | 6623 | 6637 | 6654 | 6657 | 6663 | 6670 | 6671 | 6674 | 6751 | 6756 |
| | 6772 | 6776 | 6780 | 6784 | 6788 | 6792 | 6802 | 6808 | 6842 | 6845 | 6885 | 6887 |
| | 6897 | 6897 | 6907 | 6911 | 6912 | 6917 | 6958 | 6967 | 6975 | 6982 | 6982 | 6984 |
| | 6990 | 6999 | 7000 | 7001 | 7002 | 7003 | 7005 | 7006 | 7007 | 7008 | 7013 | 7014 |
| | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7027 | 7030 | 7035 |
| | 7037 | 7050 | 7052 | 7054 | 7056 | 7058 | 7060 | 7061 | 7063 | 7065 | 7076 | 7084 |
| | 7086 | 7089 | 7092 | 7093 | 7103 | 7104 | 7107 | 7112 | 7115 | 7119 | 7134 | 7152 |
| | 7153 | 7154 | 7180 | 7188 | 7197 | 7217 | 7221 | 7221 | 7225 | 7225 | 7229 | 7229 |
| | 7233 | 7233 | 7237 | 7237 | 7239 | 7248 | 7266 | 7267 | 7276 | 7277 | 7281 | 7291 |
| | 7292 | 7293 | 7316 | 7316 | 7331 | 7333 | 7334 | 7335 | 7336 | 7337 | 7339 | 7341 |
| | 7342 | 7343 | 7344 | 7345 | 7347 | 7349 | 7359 | 7378 | 7380 | 7391 | 7393 | 7420 |
| | 7421 | 7422 | 7423 | 7424 | 7425 | 7426 | 7427 | 7428 | 7429 | 7430 | 7431 | 7432 |
| | 7433 | 7435 | 7438 | 7439 | 7442 | 7443 | 7447 | 7454 | 7454 | 7456 | 7457 | 7468 |
| | 7472 | 7486 | 7490 | 7491 | 7492 | 7496 | 7496 | 7498 | 7500 | 7514 | 7578 | 7581 |
| | 7586 | 7587 | 7589 | 7596 | 7602 | 7616 | 7617 | 7628 | 7634 | 7648 | 7650 | 7664 |
| | 7667 | 7667 | 7707 | 7712 | 7712 | 7712 | 7715 | 7715 | 7717 | 7717 | 7793 | 7802 |
| | 7804 | 7806 | 7808 | 7810 | 7812 | 7825 | 7826 | 7840 | 7860 | 7861 | 7864 | 7865 |
| | 7913 | 7924 | 8191 | 8194 | 8195 | 8199 | 8200 | 8209 | 8214 | 8220 | 8224 | 8226 |
| | 8240 | 8244 | 8245 | 8248 | 8249 | 8252 | 8258 | 8266 | 8491 | 8504 | 8508 | 8511 |
| | 8512 | 8514 | 8520 | 8527 | 8529 | 8530 | 8531 | 8532 | 8535 | 8651 | 8661 | 8665 |
| | 8675 | 8678 | 8679 | 8680 | 8688 | 8693 | 8695 | 8697 | 8698 | 8700 | 8702 | 8705 |
| | 8707 | 8711 | 8714 | 8719 | 8725 | 8727 | 8739 | 8745 | 8749 | 8752 | 8770 | 8772 |
| | 8774 | 8776 | 8778 | 8780 | 8784 | 8789 | 8792 | | | | | |
| DP | 691 # | 2172 | 2173 | 2488 | 2494 | 2520 | 2526 | 2532 | 2537 | 2540 | 2590 | 2771 |
| | 2786 | 3021 | 3042 | 3077 | 3110 | 3116 | 3299 | 3386 | 3388 | 3405 | 3428 | 3431 |
| | 3431 | 3434 | 3434 | 3437 | 3437 | 3440 | 3443 | 3443 | 3446 | 3446 | 3449 | 3449 |
| | 3653 | 3694 | 3700 | 3709 | 3711 | 3739 | 3745 | 3774 | 3779 | 3807 | 3816 | 3974 |
| | 4013 | 4017 | 4040 | 4103 | 4127 | 4150 | 4178 | 4186 | 4195 | 4197 | 4208 | 4208 |
| | 4209 | 4210 | 4212 | 4214 | 4436 | 4452 | 4453 | 4454 | 4456 | 4459 | 4478 | 4480 |
| | 4483 | 4610 | 4615 | 4620 | 4626 | 4627 | 4628 | 4706 | 4736 | 4891 | 5064 | 5113 |
| | 5133 | 5159 | 5177 | 5202 | 5247 | 5439 | 5445 | 5757 | 5765 | 5801 | 5802 | 5803 |
| | 5810 | 5962 | 6008 | 6028 | 6035 | 6045 | 6048 | 6083 | 6085 | 6087 | 6121 | 6128 |
| | 6162 | 6204 | 6221 | 6223 | 6258 | 6259 | 6259 | 6271 | 6272 | 6279 | 6281 | 6305 |
| | 6305 | 6308 | 6340 | 6346 | 6357 | 6362 | 6384 | 6385 | 6386 | 6544 | 6545 | 6590 |
| | 6605 | 6678 | 6694 | 6731 | 6820 | 6820 | 6848 | 6889 | 6927 | 6935 | 6942 | 6947 |
| | 6954 | 7250 | 7307 | 7309 | 7370 | 7543 | 7620 | 7662 | 7727 | 7729 | 7731 | 7845 |
| | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 | 7946 | 7947 |
| | 7949 | 7950 | 7952 | 7953 | 7955 | 8201 | 8760 | 8796 | 8810 | | | |
| PC FLAGS | 688 # | 3627 | 3641 | 3688 | 3805 | 3811 | 3985 | 4014 | 4215 | 6963 | 7458 | 7470 |
| | 8808 | | | | | | | | | | | |
| PI NEW | 689 # | 7417 | | | | | | | | | | |
| RAM | 692 # | 2180 | 2288 | 2300 | 2315 | 2328 | 2356 | 2369 | 2370 | 2382 | 2383 | 2587 |
| | 2688 | 2690 | 2721 | 2727 | 2780 | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2905 |
| | 2936 | 2964 | 2967 | 2981 | 2987 | 2999 | 3002 | 3015 | 3086 | 3089 | 3275 | 3290 |
| | 3293 | 3298 | 3369 | 3472 | 3486 | 3500 | 3508 | 3555 | 3563 | 3678 | 3716 | 3749 |
| | 3771 | 3827 | 3839 | 4056 | 4070 | 4083 | 4086 | 4095 | 4097 | 4100 | 4118 | 4146 |
| | 4174 | 4176 | 4188 | 4193 | 4211 | 4213 | 4316 | 4325 | 4326 | 4408 | 4410 | 4432 |
| | 4439 | 4449 | 4609 | 4613 | 4614 | 4618 | 4619 | 4623 | 4624 | 4712 | 4716 | 4748 |
| | 4756 | 4966 | 4975 | 4976 | 4988 | 4992 | 5027 | 5034 | 5039 | 5047 | 5103 | 5112 |

| | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|
| 5175 | 5200 | 5282 | 5315 | 5347 | 5411 | 5546 | 5547 | 5557 | 5637 | 5649 | 5692 |
| 5700 | 5924 | 5927 | 5954 | 5957 | 5972 | 5975 | 5991 | 5995 | 6005 | 6012 | 6021 |
| 6024 | 6040 | 6044 | 6059 | 6060 | 6061 | 6062 | 6079 | 6093 | 6095 | 6113 | 6126 |
| 6132 | 6139 | 6145 | 6157 | 6160 | 6177 | 6180 | 6184 | 6188 | 6191 | 6193 | 6200 |
| 6202 | 6210 | 6212 | 6215 | 6251 | 6253 | 6254 | 6266 | 6267 | 6286 | 6291 | 6292 |
| 6293 | 6314 | 6319 | 6331 | 6337 | 6347 | 6355 | 6360 | 6366 | 6373 | 6379 | 6380 |
| 6381 | 6402 | 6422 | 6427 | 6432 | 6434 | 6451 | 6509 | 6540 | 6550 | 6553 | 6578 |
| 6587 | 6589 | 6614 | 6629 | 6691 | 6693 | 6736 | 6737 | 6744 | 6806 | 6811 | 6813 |
| 6818 | 6822 | 6824 | 6839 | 6841 | 6868 | 6892 | 6894 | 6925 | 6929 | 6933 | 6938 |
| 6939 | 6944 | 6952 | 6953 | 6956 | 7049 | 7083 | 7151 | 7208 | 7210 | 7212 | 7214 |
| 7216 | 7245 | 7254 | 7259 | 7265 | 7297 | 7298 | 7301 | 7322 | 7532 | 7546 | 7560 |
| 7562 | 7654 | 7657 | 7659 | 7665 | 7671 | 7673 | 7833 | 7912 | 7917 | 7921 | 7936 |
| 7954 | 7956 | 7957 | 7959 | 8198 | 8498 | 8538 | 8539 | 8540 | 8546 | 8735 | 8741 |
| 8750 | 8754 | 8766 | | | | | | | | | |
| 622 # | | | | | | | | | | | |
| 623 # | 2388 | 2404 | 2488 | 2520 | 2568 | 2715 | 2717 | 2720 | 2723 | 2726 | 2729 |
| 2771 | 3042 | 3077 | 3268 | 3273 | 3299 | 3386 | 3388 | 3428 | 3431 | 3434 | 3437 |
| 3440 | 3443 | 3446 | 3449 | 3592 | 3709 | 3711 | 3769 | 3807 | 3826 | 3840 | 4103 |
| 4127 | 4150 | 4208 | 4453 | 4459 | 4628 | 4691 | 4696 | 4706 | 4710 | 4731 | 4790 |
| 4791 | 4792 | 4793 | 4794 | 4810 | 4884 | 4885 | 4886 | 4887 | 4888 | 4936 | 4952 |
| 5052 | 5092 | 5101 | 5113 | 5174 | 5177 | 5202 | 5439 | 5445 | 5489 | 5505 | 5765 |
| 5810 | 5835 | 6008 | 6028 | 6032 | 6042 | 6045 | 6085 | 6121 | 6128 | 6162 | 6221 |
| 6223 | 6258 | 6259 | 6271 | 6272 | 6305 | 6308 | 6340 | 6346 | 6357 | 6362 | 6384 |
| 6385 | 6386 | 6399 | 6456 | 6461 | 6462 | 6524 | 6545 | 6605 | 6670 | 6678 | 6731 |
| 6756 | 6820 | 6842 | 6845 | 6848 | 6885 | 6887 | 6889 | 6927 | 6935 | 6942 | 6947 |
| 6954 | 6958 | 7084 | 7250 | 7336 | 7349 | 7359 | 7543 | 7731 | 8201 | 8688 | |
| 625 # | 2148 | 2150 | 2152 | 2154 | 2157 | 2160 | 2162 | 2165 | 2174 | 2177 | 2178 |
| 2179 | 2180 | 2183 | 2190 | 2193 | 2201 | 2215 | 2218 | 2228 | 2230 | 2237 | 2241 |
| 2244 | 2247 | 2256 | 2259 | 2262 | 2265 | 2268 | 2275 | 2288 | 2294 | 2300 | 2315 |
| 2319 | 2328 | 2338 | 2352 | 2356 | 2361 | 2362 | 2365 | 2369 | 2370 | 2393 | 2399 |
| 2410 | 2413 | 2416 | 2419 | 2421 | 2434 | 2456 | 2457 | 2516 | 2576 | 2587 | 2688 |
| 2690 | 2721 | 2727 | 2737 | 2739 | 2744 | 2746 | 2751 | 2753 | 2758 | 2760 | 2770 |
| 2780 | 2803 | 2813 | 2823 | 2838 | 2855 | 2865 | 2875 | 2885 | 2895 | 2905 | 2906 |
| 2916 | 2926 | 2936 | 2946 | 2964 | 2967 | 2981 | 3015 | 3018 | 3041 | 3270 | 3275 |
| 3287 | 3290 | 3293 | 3298 | 3337 | 3369 | 3403 | 3419 | 3453 | 3456 | 3472 | 3486 |
| 3500 | 3508 | 3524 | 3525 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3548 | 3551 |
| 3563 | 3572 | 3577 | 3578 | 3579 | 3582 | 3585 | 3587 | 3593 | 3605 | 3612 | 3613 |
| 3616 | 3620 | 3626 | 3627 | 3631 | 3634 | 3639 | 3641 | 3652 | 3677 | 3678 | 3688 |
| 3706 | 3716 | 3722 | 3724 | 3749 | 3753 | 3758 | 3771 | 3805 | 3811 | 3817 | 3824 |
| 3827 | 3829 | 3832 | 3839 | 3844 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 |
| 3932 | 3938 | 3942 | 3944 | 3978 | 3982 | 3984 | 3985 | 3988 | 3993 | 4001 | 4014 |
| 4021 | 4025 | 4030 | 4036 | 4041 | 4056 | 4070 | 4083 | 4086 | 4090 | 4091 | 4095 |
| 4097 | 4100 | 4104 | 4105 | 4117 | 4123 | 4129 | 4131 | 4132 | 4145 | 4146 | 4151 |
| 4154 | 4155 | 4169 | 4185 | 4199 | 4215 | 4222 | 4315 | 4318 | 4320 | 4324 | 4325 |
| 4339 | 4345 | 4351 | 4357 | 4374 | 4394 | 4405 | 4406 | 4412 | 4423 | 4429 | 4438 |
| 4446 | 4458 | 4465 | 4468 | 4517 | 4532 | 4534 | 4535 | 4537 | 4542 | 4543 | 4556 |
| 4623 | 4624 | 4635 | 4636 | 4638 | 4640 | 4644 | 4645 | 4646 | 4716 | 4729 | 4735 |
| 4764 | 4770 | 4812 | 4823 | 4908 | 4909 | 4910 | 4911 | 4945 | 4954 | 4966 | 4968 |
| 4976 | 4978 | 4982 | 4985 | 5011 | 5024 | 5034 | 5037 | 5043 | 5054 | 5061 | 5097 |
| 5100 | 5103 | 5105 | 5112 | 5115 | 5139 | 5141 | 5143 | 5148 | 5163 | 5165 | 5166 |
| 5175 | 5200 | 5218 | 5224 | 5249 | 5250 | 5251 | 5279 | 5282 | 5285 | 5286 | 5287 |
| 5289 | 5290 | 5293 | 5294 | 5299 | 5315 | 5318 | 5319 | 5325 | 5345 | 5347 | 5351 |
| 5352 | 5353 | 5354 | 5389 | 5411 | 5412 | 5413 | 5443 | 5447 | 5448 | 5496 | 5498 |
| 5508 | 5511 | 5513 | 5514 | 5515 | 5526 | 5543 | 5544 | 5557 | 5562 | 5563 | 5570 |
| 5571 | 5581 | 5606 | 5608 | 5609 | 5613 | 5616 | 5633 | 5644 | 5656 | 5657 | 5686 |

(U) DEST A

AD

| | | | | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|------|------|------|
| 5688 | 5689 | 5694 | 5695 | 5696 | 5704 | 5715 | 5717 | 5731 | 5732 | 5755 | 5771 |
| 5773 | 5775 | 5790 | 5792 | 5794 | 5796 | 5799 | 5819 | 5820 | 5836 | 5879 | 5881 |
| 5883 | 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | 5911 | 5914 | 5916 | 5924 | 5925 |
| 5927 | 5930 | 5939 | 5950 | 5954 | 5957 | 5958 | 5964 | 5966 | 5971 | 5972 | 5973 |
| 5975 | 5983 | 5991 | 5993 | 5995 | 5997 | 6000 | 6005 | 6009 | 6012 | 6014 | 6015 |
| 6016 | 6021 | 6022 | 6024 | 6030 | 6039 | 6040 | 6044 | 6046 | 6051 | 6059 | 6060 |
| 6061 | 6062 | 6069 | 6070 | 6074 | 6076 | 6079 | 6084 | 6088 | 6093 | 6095 | 6098 |
| 6099 | 6100 | 6111 | 6113 | 6115 | 6122 | 6126 | 6130 | 6132 | 6133 | 6134 | 6135 |
| 6140 | 6141 | 6145 | 6154 | 6157 | 6158 | 6160 | 6165 | 6169 | 6170 | 6177 | 6180 |
| 6184 | 6188 | 6191 | 6193 | 6202 | 6210 | 6212 | 6215 | 6218 | 6229 | 6231 | 6240 |
| 6241 | 6245 | 6246 | 6251 | 6253 | 6254 | 6256 | 6257 | 6261 | 6263 | 6264 | 6266 |
| 6267 | 6269 | 6286 | 6288 | 6291 | 6292 | 6293 | 6299 | 6301 | 6304 | 6309 | 6310 |
| 6314 | 6315 | 6319 | 6321 | 6325 | 6328 | 6331 | 6333 | 6334 | 6335 | 6337 | 6345 |
| 6347 | 6348 | 6350 | 6351 | 6353 | 6355 | 6360 | 6366 | 6369 | 6373 | 6378 | 6379 |
| 6380 | 6381 | 6383 | 6387 | 6388 | 6389 | 6393 | 6397 | 6400 | 6402 | 6404 | 6416 |
| 6420 | 6422 | 6424 | 6425 | 6427 | 6430 | 6432 | 6434 | 6436 | 6438 | 6449 | 6451 |
| 6455 | 6464 | 6467 | 6501 | 6502 | 6506 | 6509 | 6518 | 6520 | 6525 | 6550 | 6553 |
| 6559 | 6563 | 6578 | 6579 | 6587 | 6589 | 6591 | 6593 | 6596 | 6600 | 6602 | 6614 |
| 6617 | 6619 | 6621 | 6623 | 6629 | 6637 | 6657 | 6663 | 6668 | 6671 | 6674 | 6676 |
| 6691 | 6693 | 6727 | 6736 | 6744 | 6753 | 6772 | 6776 | 6780 | 6784 | 6788 | 6792 |
| 6802 | 6806 | 6808 | 6811 | 6813 | 6817 | 6818 | 6822 | 6824 | 6839 | 6841 | 6844 |
| 6846 | 6867 | 6868 | 6886 | 6888 | 6897 | 6907 | 6912 | 6917 | 6925 | 6929 | 6933 |
| 6938 | 6939 | 6943 | 6944 | 6946 | 6950 | 6952 | 6953 | 6956 | 6963 | 6966 | 6974 |
| 6982 | 6990 | 6999 | 7000 | 7001 | 7002 | 7003 | 7005 | 7006 | 7007 | 7008 | 7009 |
| 7010 | 7013 | 7014 | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 |
| 7027 | 7030 | 7035 | 7037 | 7049 | 7050 | 7052 | 7053 | 7054 | 7057 | 7059 | 7060 |
| 7062 | 7064 | 7076 | 7083 | 7086 | 7089 | 7092 | 7093 | 7095 | 7104 | 7107 | 7112 |
| 7119 | 7122 | 7127 | 7134 | 7137 | 7139 | 7140 | 7151 | 7152 | 7154 | 7169 | 7180 |
| 7181 | 7186 | 7188 | 7193 | 7197 | 7198 | 7208 | 7210 | 7212 | 7214 | 7216 | 7217 |
| 7221 | 7225 | 7229 | 7233 | 7237 | 7239 | 7245 | 7248 | 7249 | 7254 | 7255 | 7257 |
| 7259 | 7260 | 7265 | 7266 | 7267 | 7277 | 7278 | 7281 | 7291 | 7292 | 7293 | 7296 |
| 7297 | 7298 | 7301 | 7308 | 7316 | 7322 | 7328 | 7332 | 7334 | 7335 | 7338 | 7340 |
| 7342 | 7344 | 7346 | 7348 | 7360 | 7361 | 7377 | 7378 | 7380 | 7384 | 7391 | 7393 |
| 7397 | 7420 | 7421 | 7422 | 7423 | 7424 | 7425 | 7426 | 7427 | 7428 | 7429 | 7430 |
| 7431 | 7432 | 7433 | 7435 | 7439 | 7443 | 7445 | 7447 | 7449 | 7454 | 7458 | 7468 |
| 7470 | 7472 | 7474 | 7484 | 7486 | 7488 | 7491 | 7492 | 7494 | 7496 | 7500 | 7502 |
| 7514 | 7518 | 7532 | 7546 | 7556 | 7558 | 7560 | 7562 | 7565 | 7578 | 7581 | 7587 |
| 7590 | 7596 | 7602 | 7617 | 7628 | 7634 | 7647 | 7654 | 7657 | 7659 | 7665 | 7667 |
| 7671 | 7673 | 7674 | 7681 | 7707 | 7712 | 7715 | 7719 | 7721 | 7793 | 7802 | 7804 |
| 7806 | 7808 | 7810 | 7812 | 7826 | 7833 | 7843 | 7846 | 7860 | 7861 | 7865 | 7912 |
| 7917 | 7921 | 7924 | 7932 | 7936 | 7939 | 7942 | 7945 | 7948 | 7951 | 7954 | 7956 |
| 7957 | 7959 | 8191 | 8194 | 8195 | 8198 | 8200 | 8209 | 8220 | 8224 | 8226 | 8240 |
| 8245 | 8249 | 8252 | 8258 | 8260 | 8266 | 8495 | 8498 | 8504 | 8507 | 8508 | 8512 |
| 8514 | 8516 | 8519 | 8520 | 8527 | 8529 | 8531 | 8535 | 8538 | 8539 | 8540 | 8546 |
| 8655 | 8665 | 8670 | 8675 | 8679 | 8695 | 8698 | 8702 | 8707 | 8714 | 8719 | 8725 |
| 8727 | 8730 | 8735 | 8741 | 8749 | 8750 | 8754 | 8758 | 8764 | 8766 | 8768 | 8770 |
| 8772 | 8774 | 8776 | 8778 | 8780 | 8782 | 8783 | 8784 | 8789 | 8792 | 8808 | 8814 |
| 8816 | 8818 | 8829 | 8833 | | | | | | | | |
| 632 # | 2151 | 2423 | 2425 | 2426 | 2983 | 2987 | 2999 | 3002 | 3004 | 3005 | 3009 |
| 3040 | 3092 | 3095 | 4167 | 4168 | 4174 | 4176 | 4186 | 4188 | 4193 | 4224 | 4386 |
| 4408 | 4414 | 4545 | 4546 | 4547 | 4553 | 4803 | 4814 | 4890 | 4939 | 4942 | 4949 |
| 4958 | 5360 | 5368 | 5370 | 5375 | 5376 | 5435 | 5484 | 5486 | 5502 | 5503 | 5522 |
| 5523 | 5528 | 5546 | 5547 | 5550 | 5552 | 5637 | 5638 | 5645 | 5649 | 5652 | 5653 |
| 5671 | 5692 | 5700 | 5703 | 5705 | 5829 | 5840 | 6463 | 6466 | 6470 | 6472 | 6552 |
| 6654 | 6660 | 6667 | 6730 | 7177 | 7179 | 7187 | 7194 | 7196 | 7498 | 7501 | 7519 |

AD* .5

Cross Reference Listing

| | | | | | | | | | | | | | |
|----------|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| | 7594 | 7600 | 8263 | 8279 | | | | | | | | | |
| AD*2 | 630 # | 2149 | 2971 | 2994 | 3007 | 3010 | 3037 | 4122 | 4149 | 4205 | 4448 | 4588 | |
| | 4712 | 5049 | 5058 | 5211 | 5231 | 5242 | 5355 | 5358 | 5363 | 5438 | 6200 | 6201 | |
| | 6457 | 6641 | 6747 | 7131 | 7146 | 7148 | 7446 | 7482 | 7626 | 7632 | 7723 | 8501 | |
| PASS | 627 # | 2159 | 2167 | 2169 | 2170 | 2173 | 2175 | 2176 | 2182 | 2184 | 2227 | 2231 | |
| | 2249 | 2306 | 2333 | 2342 | 2429 | 2494 | 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | |
| | 2787 | 3110 | 3325 | 3405 | 3452 | 3457 | 3531 | 3567 | 3629 | 3640 | 3662 | 3694 | |
| | 3700 | 3729 | 3734 | 3739 | 3745 | 3774 | 3779 | 3813 | 3816 | 3841 | 3974 | 4013 | |
| | 4017 | 4040 | 4125 | 4153 | 4195 | 4210 | 4212 | 4214 | 4217 | 4392 | 4436 | 4456 | |
| | 4478 | 4480 | 4483 | 4610 | 4615 | 4620 | 4626 | 4627 | 4689 | 4692 | 4701 | 4736 | |
| | 4752 | 4760 | 4891 | 4948 | 4961 | 4971 | 4981 | 5013 | 5093 | 5121 | 5129 | 5245 | |
| | 5757 | 5801 | 5802 | 5803 | 5921 | 5928 | 5934 | 5937 | 5962 | 5967 | 5984 | 6029 | |
| | 6055 | 6056 | 6057 | 6072 | 6103 | 6119 | 6142 | 6155 | 6173 | 6204 | 6294 | 6296 | |
| | 6352 | 6358 | 6371 | 6377 | 6454 | 6510 | 6576 | 6588 | 6590 | 6630 | 6694 | 6724 | |
| | 6893 | 6895 | 6913 | 6930 | 6964 | 7028 | 7031 | 7071 | 7101 | 7156 | 7222 | 7226 | |
| | 7230 | 7234 | 7238 | 7252 | 7263 | 7284 | 7287 | 7294 | 7307 | 7309 | 7317 | 7318 | |
| | 7351 | 7369 | 7370 | 7375 | 7402 | 7403 | 7453 | 7517 | 7584 | 7619 | 7620 | 7622 | |
| | 7651 | 7666 | 7685 | 7691 | 7693 | 7725 | 7727 | 7729 | 7822 | 7830 | 7838 | 7845 | |
| | 7857 | 7910 | 7925 | 7930 | 7931 | 7933 | 7934 | 7935 | 7937 | 7938 | 7940 | 7941 | |
| | 7943 | 7944 | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7958 | 7958 | 8197 | 8207 | |
| | 8208 | 8210 | 8211 | 8215 | 8218 | 8228 | 8230 | 8232 | 8234 | 8238 | 8239 | 8525 | |
| | 8541 | 8724 | 8793 | 8796 | 8810 | | | | | | | | |
| Q_AD | 626 # | 2382 | 2980 | 3086 | 3112 | 3114 | 4118 | 4144 | 4183 | 4203 | 4211 | 4213 | |
| | 4223 | 4316 | 4326 | 4385 | 4395 | 4397 | 4410 | 4432 | 4435 | 4437 | 4439 | 4450 | |
| | 4463 | 4464 | 4467 | 4474 | 4482 | 4510 | 4515 | 4516 | 4533 | 4558 | 4609 | 4613 | |
| | 4614 | 4618 | 4619 | 4818 | 4894 | 4901 | 4963 | 4972 | 4975 | 4992 | 5016 | 5032 | |
| | 5039 | 5099 | 5156 | 5205 | 5216 | 5221 | 5236 | 5239 | 5296 | 5320 | 5323 | 5367 | |
| | 5378 | 5390 | 5394 | 5396 | 5397 | 5410 | 5423 | 5481 | 5483 | 5485 | 5487 | 5555 | |
| | 5564 | 5567 | 5572 | 5582 | 5587 | 5627 | 5632 | 5642 | 5655 | 5661 | 5673 | 5676 | |
| | 5706 | 5711 | 5716 | 5718 | 5729 | 5753 | 5769 | 5770 | 5788 | 5980 | 6139 | 6540 | |
| | 7717 | | | | | | | | | | | | |
| Q_Q*.5 | 631 # | 2383 | 2384 | 3070 | 3089 | 3098 | 3142 | 4119 | 4147 | 4226 | 4247 | 4248 | |
| | 4260 | 4267 | 4275 | 4285 | 4293 | 4388 | 4390 | 4398 | 4402 | 4418 | 4433 | 4440 | |
| | 4492 | 4540 | 5040 | 5207 | 5227 | 5228 | 5229 | 5243 | 5298 | 5321 | 5326 | 5401 | |
| | 5403 | 5433 | 5576 | 5601 | 5604 | 5612 | 5615 | 5617 | 5631 | 5666 | 5707 | 5721 | |
| | 5747 | 5750 | 5784 | 5786 | 5832 | | | | | | | | |
| Q_Q*2 | 629 # | 2990 | 3071 | 3074 | 3075 | 3076 | 3101 | 3105 | 3108 | 3143 | 3146 | 3147 | |
| | 4271 | 4279 | 4289 | 4297 | 4434 | 4444 | 4472 | 4475 | 4548 | 4551 | 4552 | 4555 | |
| | 4557 | 4579 | 4582 | 4594 | 4599 | 4819 | 4822 | 4895 | 4898 | 4903 | 5222 | 5223 | |
| | 5480 | 5492 | 5500 | 5504 | 5578 | 5583 | 5585 | 5597 | 5599 | 5600 | 5629 | 5670 | |
| | 5674 | 5723 | 5740 | 5762 | 5779 | 5807 | 5982 | | | | | | |
| (U) DISP | 861 # | 4560 | | | | | | | | | | | |
| ADISP | 868 # | | | | | | | | | | | | |
| AREAD | 864 # | 2320 | | | | | | | | | | | |
| BDISP | 869 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 | |
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 3270 | 3276 | 3369 | |
| | 3388 | 3389 | 3472 | 3486 | 3500 | 3510 | 3695 | 3702 | 3740 | 4057 | 4071 | 4123 | |
| | 4129 | 4152 | 4154 | 5206 | 5435 | 5443 | 5447 | 5493 | 5505 | 5515 | 5522 | 5926 | |
| | 5968 | 6112 | 6167 | 6374 | 6728 | 6800 | 6807 | 7532 | 7559 | 7577 | 7588 | 7615 | |
| | 8517 | 8817 | | | | | | | | | | | |
| BYTE | 874 # | 4704 | 4721 | 6149 | 6852 | 6873 | | | | | | | |
| CONSOLE | 862 # | | | | | | | | | | | | |
| DP | 867 # | 4426 | 4428 | 4445 | 4470 | 6473 | 6500 | 8760 | | | | | |
| DP LEFT | 865 # | 6459 | 6754 | 7418 | 8216 | | | | | | | | |
| DROM | 863 # | 2353 | 2357 | 2366 | 2371 | 2379 | 2400 | 2406 | 2430 | 2435 | 5935 | 5938 | |

| | | | | | | | | | | | | |
|----------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| EAMODE | 875 # | 2278 | 2344 | 3553 | 4743 | 4746 | 5922 | 6890 | 6898 | 7656 | | |
| MUL | 871 # | 4267 | 4275 | 4285 | 4293 | 5579 | 5605 | 5607 | 5724 | | | |
| NICOND | 873 # | 2227 | 2228 | 2230 | 2231 | 2291 | 2297 | 2521 | 3042 | 3088 | 3116 | 3325 |
| | 3337 | 3428 | 3452 | 3453 | 3456 | 3457 | 3524 | 3525 | 3551 | 3593 | 3710 | 3786 |
| | 3793 | 4217 | 4481 | 4484 | 4692 | 5439 | 5445 | 5804 | 7351 | 7622 | 8201 | |
| NORM | 866 # | 5299 | 5328 | 5405 | 5480 | 5482 | 5492 | 5496 | 5498 | 5500 | 5501 | 5504 |
| | 5526 | 5677 | 5731 | 5732 | 5742 | 5744 | 5764 | 5774 | 5776 | 5781 | 5782 | 5809 |
| | 5838 | | | | | | | | | | | |
| PAGE FAIL | 872 # | | | | | | | | | | | |
| RETURN | 870 # | 2456 | 2457 | 3558 | 3562 | 4023 | 4026 | 4229 | 4272 | 4280 | 4290 | 4298 |
| | 4492 | 4514 | 4515 | 4517 | 4601 | 4628 | 4730 | 4732 | 4736 | 4771 | 4804 | 4823 |
| | 4891 | 5105 | 5242 | 5243 | 5525 | 5527 | 5528 | 5617 | 5812 | 5819 | 5820 | 5835 |
| | 6224 | 6234 | 6239 | 6242 | 6280 | 6281 | 6431 | 6437 | 6608 | 6631 | 6696 | 6727 |
| | 6739 | 6770 | 6786 | 6803 | 6812 | 6814 | 6840 | 6875 | 6908 | 6914 | 6919 | 6959 |
| | 7198 | 7264 | 7288 | 7363 | 7407 | 7515 | 7518 | 7590 | 7597 | 7603 | 7660 | 7668 |
| | 7672 | 7675 | 7701 | 7713 | 7715 | 7717 | 7719 | 7721 | 7723 | 7725 | 7727 | 7729 |
| | 7731 | 7922 | 7926 | 7960 | 8547 | 8745 | 8833 | | | | | |
| SCADO | 876 # | 4691 | 4696 | 4710 | 4786 | 5282 | 5425 | 5427 | 5980 | 5982 | 6033 | 6843 |
| | 6885 | 7687 | | | | | | | | | | |
| (U) DIVIDE | 962 # | 4551 | 4552 | 4596 | 4600 | | | | | | | |
| (U) DONT CACHE | 1132 # | | | | | | | | | | | |
| (U) DP FUNC | 1149 # | 7584 | 7619 | 7725 | 8526 | 8542 | | | | | | |
| (U) DT | 931 # | | | | | | | | | | | |
| 2T | 934 # | | | | | | | | | | | |
| 3T | 935 # | 2180 | 2370 | 2382 | 2383 | 2493 | 2525 | 2531 | 2536 | 2539 | 2589 | 2780 |
| | 2786 | 2987 | 2999 | 3002 | 3016 | 3086 | 3089 | 3331 | 3334 | 3343 | 3346 | 3405 |
| | 3434 | 3437 | 3446 | 3449 | 3591 | 3591 | 3606 | 3606 | 3693 | 3699 | 3738 | 3744 |
| | 3815 | 3973 | 3986 | 3991 | 4012 | 4016 | 4039 | 4083 | 4095 | 4121 | 4126 | 4188 |
| | 4193 | 4211 | 4213 | 4326 | 4328 | 4330 | 4334 | 4348 | 4387 | 4393 | 4408 | 4410 |
| | 4422 | 4432 | 4439 | 4452 | 4533 | 4609 | 4611 | 4613 | 4614 | 4616 | 4618 | 4619 |
| | 4621 | 4637 | 4644 | 4691 | 4696 | 4704 | 4710 | 4712 | 4721 | 4736 | 4891 | 4962 |
| | 4966 | 4975 | 4976 | 4991 | 5021 | 5034 | 5039 | 5103 | 5132 | 5158 | 5175 | 5247 |
| | 5299 | 5328 | 5345 | 5365 | 5371 | 5373 | 5405 | 5446 | 5480 | 5482 | 5488 | 5492 |
| | 5496 | 5498 | 5500 | 5501 | 5504 | 5507 | 5512 | 5526 | 5547 | 5649 | 5668 | 5672 |
| | 5675 | 5677 | 5689 | 5700 | 5719 | 5731 | 5732 | 5742 | 5743 | 5744 | 5746 | 5749 |
| | 5752 | 5761 | 5764 | 5770 | 5774 | 5776 | 5781 | 5782 | 5806 | 5809 | 5812 | 5830 |
| | 5838 | 5912 | 5954 | 5955 | 5972 | 5975 | 5991 | 5995 | 6005 | 6012 | 6021 | 6024 |
| | 6040 | 6044 | 6059 | 6060 | 6061 | 6062 | 6079 | 6093 | 6094 | 6096 | 6113 | 6126 |
| | 6132 | 6136 | 6139 | 6145 | 6149 | 6160 | 6177 | 6178 | 6180 | 6184 | 6188 | 6189 |
| | 6191 | 6193 | 6198 | 6200 | 6202 | 6210 | 6212 | 6215 | 6251 | 6253 | 6254 | 6265 |
| | 6266 | 6267 | 6268 | 6286 | 6292 | 6319 | 6323 | 6331 | 6336 | 6337 | 6347 | 6355 |
| | 6360 | 6366 | 6373 | 6379 | 6380 | 6381 | 6382 | 6396 | 6398 | 6402 | 6452 | 6487 |
| | 6490 | 6509 | 6540 | 6544 | 6550 | 6553 | 6557 | 6578 | 6587 | 6589 | 6590 | 6594 |
| | 6614 | 6615 | 6629 | 6691 | 6693 | 6736 | 6737 | 6738 | 6744 | 6751 | 6806 | 6811 |
| | 6813 | 6818 | 6822 | 6824 | 6839 | 6841 | 6852 | 6868 | 6873 | 6925 | 6929 | 6933 |
| | 6938 | 6939 | 6944 | 6952 | 6953 | 6956 | 7002 | 7003 | 7049 | 7056 | 7058 | 7060 |
| | 7061 | 7063 | 7065 | 7083 | 7092 | 7151 | 7208 | 7210 | 7212 | 7214 | 7216 | 7245 |
| | 7254 | 7259 | 7261 | 7265 | 7266 | 7295 | 7297 | 7298 | 7301 | 7307 | 7309 | 7322 |
| | 7331 | 7333 | 7337 | 7339 | 7341 | 7343 | 7345 | 7347 | 7370 | 7584 | 7589 | 7616 |
| | 7619 | 7620 | 7648 | 7650 | 7654 | 7657 | 7659 | 7664 | 7671 | 7673 | 7725 | 7727 |
| | 7729 | 7845 | 7912 | 7917 | 7918 | 7921 | 7930 | 7931 | 7933 | 7934 | 7936 | 7937 |
| | 7938 | 7940 | 7941 | 7943 | 7944 | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7954 |
| | 7955 | 7956 | 7957 | 7959 | 8198 | 8199 | 8248 | 8491 | 8498 | 8511 | 8530 | 8532 |
| | 8538 | 8539 | 8540 | 8546 | 8680 | 8693 | 8697 | 8700 | 8705 | 8711 | 8735 | 8741 |
| | 8750 | 8752 | 8754 | 8766 | 8795 | 8809 | | | | | | |

| | | | | | | | | | | | | | |
|---------------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 3940 | 4022 | 4038 | 4103 | 4104 | 4105 | 4127 | 4150 | 4357 | 4396 | 4458 | 4512 | |
| | 4635 | 4812 | 4947 | 4955 | 5018 | 5063 | 5098 | 5102 | 5289 | 5290 | 5293 | 5294 | |
| | 5318 | 5319 | 5351 | 5352 | 5412 | 5413 | 5489 | 5505 | 5608 | 5609 | 5616 | 5632 | |
| | 5754 | 5789 | 5819 | 5820 | 5835 | 5917 | 5939 | 6166 | 6171 | 6240 | 6256 | 6257 | |
| | 6263 | 6302 | 6304 | 6311 | 6317 | 6326 | 6345 | 6397 | 6400 | 6405 | 6430 | 6436 | |
| | 6507 | 6601 | 6603 | 6673 | 6677 | 6731 | 6773 | 6777 | 6781 | 6785 | 6789 | 6793 | |
| | 6809 | 6907 | 6919 | 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 | |
| | 7014 | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7045 | 7085 | |
| | 7088 | 7109 | 7114 | 7124 | 7129 | 7136 | 7141 | 7182 | 7189 | 7217 | 7239 | 7250 | |
| | 7334 | 7336 | 7338 | 7340 | 7472 | 7674 | 7719 | 7793 | 7802 | 7804 | 7806 | 7808 | |
| | 7810 | 7812 | 8193 | 8196 | 8247 | 8251 | 8254 | 8259 | 8506 | 8510 | 8513 | 8524 | |
| | 8699 | 8704 | 8709 | 8715 | 8729 | 8792 | | | | | | | |
| (U) HALT | 1281 | # | | | | | | | | | | | |
| BW14 | 1291 | # | 2516 | | | | | | | | | | |
| CSL | 1285 | # | 2247 | 2265 | | | | | | | | | |
| HALT | 1284 | # | 3579 | | | | | | | | | | |
| ILLII | 1288 | # | 7468 | | | | | | | | | | |
| ILLINT | 1289 | # | 7500 | | | | | | | | | | |
| IOPF | 1287 | # | 8749 | | | | | | | | | | |
| MULERR | 1293 | # | 2190 | | | | | | | | | | |
| NICOND 5 | 1292 | # | | | | | | | | | | | |
| POWER | 1283 | # | 2218 | | | | | | | | | | |
| (U) HOLD USER | 1088 | # | 2216 | 2577 | 3403 | 3419 | 3486 | 3500 | 3557 | 3561 | 3597 | 3619 | 3689 |
| | 3701 | | 3757 | 3787 | 3791 | 3806 | 3812 | 4004 | 4057 | 4071 | 4088 | 4098 | 4101 |
| | 4131 | | 4132 | 4156 | 4206 | 4218 | 4228 | 4231 | 4338 | 4373 | 4404 | 4490 | 4639 |
| | 4641 | | 4706 | 4723 | 4745 | 4997 | 5349 | 5361 | 5426 | 5714 | 6388 | 6389 | 6988 |
| (D) I | 1346 | # | 2547 | 2552 | 2557 | 2562 | 2600 | 2605 | 2610 | 2615 | 2620 | 2625 | 2630 |
| | 2635 | | 2642 | 2647 | 2652 | 2657 | 2662 | 2667 | 2672 | 2677 | 2796 | 2797 | 2798 |
| | 2799 | | 2807 | 2817 | 2827 | 2832 | 2842 | 2849 | 2859 | 2869 | 2879 | 2888 | 2889 |
| | 2890 | | 2891 | 2899 | 2910 | 2920 | 2930 | 2939 | 2940 | 2941 | 2942 | 2954 | 2955 |
| | 3174 | | 3175 | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 | 3183 | 3184 | 3192 | 3193 |
| | 3194 | | 3195 | 3196 | 3197 | 3198 | 3199 | 3210 | 3211 | 3212 | 3213 | 3214 | 3215 |
| | 3216 | | 3217 | 3227 | 3228 | 3229 | 3230 | 3231 | 3232 | 3233 | 3234 | 3349 | 3350 |
| | 3351 | | 3352 | 3353 | 3354 | 3355 | 3356 | 3461 | 3462 | 3463 | 3464 | 3465 | 3466 |
| | 3467 | | 3468 | 3475 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 | 3489 | 3490 |
| | 3491 | | 3492 | 3493 | 3494 | 3495 | 3496 | 3503 | 3504 | 3515 | 3517 | 3668 | 3669 |
| | 3670 | | 3671 | 3765 | 3798 | 3799 | 3800 | 3801 | 3851 | 3852 | 3853 | 3854 | 3855 |
| | 3856 | | 3857 | 3858 | 3862 | 3863 | 3864 | 3865 | 3866 | 3867 | 3868 | 3869 | 3870 |
| | 3871 | | 3872 | 3873 | 3874 | 3875 | 3876 | 3877 | 3878 | 3879 | 3880 | 3881 | 3882 |
| | 3883 | | 3884 | 3885 | 3886 | 3887 | 3888 | 3889 | 3890 | 3891 | 3892 | 3893 | 3897 |
| | 3898 | | 3899 | 3900 | 3904 | 3905 | 3906 | 3907 | 3908 | 3909 | 3910 | 3911 | 3912 |
| | 3913 | | 3914 | 4050 | 4064 | 4112 | 4137 | 4304 | 4309 | 5108 | 5384 | 5848 | 5849 |
| | 5850 | | 5851 | 5852 | 5853 | 5854 | 5856 | 5857 | 5858 | 5859 | 5861 | 5862 | 5863 |
| | 5864 | | 5866 | 5867 | 5868 | 5869 | 5870 | 5871 | 5872 | 5873 | 5907 | 7736 | 7737 |
| | 7738 | | 7740 | 7741 | 7743 | 7744 | 7746 | 7747 | 7748 | 7749 | 7750 | 7751 | 7752 |
| | 7753 | | 7755 | 7756 | 7757 | 7758 | 7759 | 7760 | 7761 | 7762 | 7764 | 7765 | 7766 |
| | 7767 | | 7768 | 7769 | 7770 | 7771 | 7773 | 7774 | 7775 | 7776 | 7777 | 7778 | 7779 |
| | 7780 | | 7782 | 7783 | 7784 | 7785 | 7786 | 7787 | 7788 | 7789 | | | |
| (U) I.CO3 | 1183 | # | | | | | | | | | | | |
| (U) I.CO4 | 1184 | # | | | | | | | | | | | |
| (U) I.CO5 | 1185 | # | | | | | | | | | | | |
| (U) I.CO6 | 1186 | # | | | | | | | | | | | |
| (U) I.CO7 | 1187 | # | | | | | | | | | | | |
| (U) IO BYTE | 1168 | # | 7579 | 7629 | 7635 | | | | | | | | |
| (U) IO CYCLE | 1162 | # | 7439 | 7486 | 7578 | 7581 | 7617 | 7628 | 7634 | 7860 | | | |

| | | | | | | | | | | | | |
|---------|--------|------|------|------|------|------|------|------|------|------|------|------|
| (U) J | 524 # | 2320 | 2353 | 2357 | 2366 | 2371 | 2379 | 2400 | 2406 | 2430 | 2435 | 2456 |
| | 2457 | 3558 | 3562 | 4023 | 4026 | 4229 | 4272 | 4280 | 4290 | 4298 | 4492 | 4514 |
| | 4515 | 4517 | 4561 | 4601 | 4628 | 4730 | 4732 | 4736 | 4771 | 4804 | 4823 | 4891 |
| | 5105 | 5242 | 5243 | 5525 | 5527 | 5528 | 5617 | 5812 | 5819 | 5820 | 5835 | 5935 |
| | 5938 | 6224 | 6234 | 6239 | 6242 | 6280 | 6281 | 6431 | 6437 | 6608 | 6631 | 6696 |
| | 6727 | 6739 | 6770 | 6786 | 6803 | 6812 | 6814 | 6840 | 6875 | 6908 | 6914 | 6919 |
| | 6959 | 7198 | 7264 | 7288 | 7363 | 7407 | 7515 | 7518 | 7590 | 7597 | 7603 | 7660 |
| | 7668 | 7672 | 7675 | 7701 | 7713 | 7715 | 7717 | 7719 | 7721 | 7723 | 7725 | 7727 |
| | 7729 | 7731 | 7854 | 7922 | 7926 | 7960 | 8547 | 8745 | 8833 | | | |
| | 8745 # | 3945 | 8726 | | | | | | | | | |
| ABORT | 7119 # | 7111 | | | | | | | | | | |
| ACBSET | 7731 # | 3831 | 6945 | | | | | | | | | |
| AC_ARX | 4056 # | 4049 | 4050 | 4051 | 4052 | | | | | | | |
| ADD | 6239 # | 6214 | 6220 | 6233 | | | | | | | | |
| ADDCRY | 4936 # | 4690 | | | | | | | | | | |
| ADJBP | 4939 # | 4941 | | | | | | | | | | |
| ADJBPO | 4958 # | 4960 | | | | | | | | | | |
| ADJBP1 | 4968 # | 4965 | | | | | | | | | | |
| ADJBP2 | 5011 # | 4996 | | | | | | | | | | |
| ADJBP3 | 5023 # | 5035 | | | | | | | | | | |
| ADJBP4 | 5037 # | 5031 | | | | | | | | | | |
| ADJBP5 | 5058 # | 5060 | | | | | | | | | | |
| ADJBP6 | 3769 # | 3765 | | | | | | | | | | |
| ADJSP | 3786 # | 3778 | | | | | | | | | | |
| ADJSP1 | 3791 # | 3783 | | | | | | | | | | |
| ADJSP2 | 2813 # | 2806 | 2807 | 2808 | 2809 | 2838 | | | | | | |
| AND | 2823 # | 2816 | 2817 | 2818 | 2819 | 2875 | | | | | | |
| ANDCA | 2875 # | 2868 | 2869 | 2870 | 2871 | | | | | | | |
| ANDCB | 2838 # | 2831 | 2832 | 2833 | 2834 | | | | | | | |
| ANDCM | 3508 # | 3503 | 3504 | | | | | | | | | |
| AOBJ | 3486 # | 3475 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 | | | |
| AOJ | 3403 # | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 | | | |
| AOS | 7035 # | | | | | | | | | | | |
| APRID | 7027 # | 7003 | | | | | | | | | | |
| APRSO | 7030 # | 7002 | | | | | | | | | | |
| APRSZ | 2456 # | 2398 | | | | | | | | | | |
| ARSIGN | 2980 # | 2951 | | | | | | | | | | |
| ASH | 3083 # | 2955 | | | | | | | | | | |
| ASHC | 3089 # | 3087 | | | | | | | | | | |
| ASHC1 | 3105 # | 3094 | 3107 | | | | | | | | | |
| ASHCL | 3116 # | 3077 | 3113 | | | | | | | | | |
| ASHCQ1 | 3098 # | 3100 | | | | | | | | | | |
| ASHCR | 3110 # | 3102 | | | | | | | | | | |
| ASHCX | 2990 # | 2990 | | | | | | | | | | |
| ASHL | 2987 # | 2980 | | | | | | | | | | |
| ASHLO | 2983 # | 2969 | | | | | | | | | | |
| ASHR | 2994 # | 3007 | | | | | | | | | | |
| ASHX | 3007 # | 3012 | | | | | | | | | | |
| ASHXX | 6958 # | 6926 | 6934 | | | | | | | | | |
| BACKBP | 6933 # | 8775 | 8783 | 8785 | | | | | | | | |
| BACKD | 6925 # | 8782 | | | | | | | | | | |
| BACKS | 8237 # | 8222 | | | | | | | | | | |
| BADDATA | 6402 # | 6338 | | | | | | | | | | |
| BDABT | 6404 # | 6383 | | | | | | | | | | |
| BDCFLG | 6286 # | | | | | | | | | | | |
| BDEC | | | | | | | | | | | | |

| | | | | | | | | | | | | | |
|-------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| BDECO | 6292 # | 6290 | | | | | | | | | | | |
| BDEC1 | 6299 # | 6308 | | | | | | | | | | | |
| BDEC2 | 6304 # | 6312 | | | | | | | | | | | |
| BDEC3 | 6314 # | 6300 | 6322 | | | | | | | | | | |
| BDEC4 | 6319 # | 6314 | | | | | | | | | | | |
| BDEC5 | 6331 # | 6317 | 6327 | | | | | | | | | | |
| BDECLP | 6369 # | 6388 | | | | | | | | | | | |
| BDFILL | 6353 # | 6362 | | | | | | | | | | | |
| BDSET | 6377 # | 6397 | 6400 | | | | | | | | | | |
| BDSUB | 6416 # | 6320 | 6370 | | | | | | | | | | |
| BDSUB1 | 6420 # | 6417 | | | | | | | | | | | |
| BDSUB2 | 6436 # | 6439 | | | | | | | | | | | |
| BDTBL | 6393 # | 6376 | | | | | | | | | | | |
| BITCHK | 6907 # | 5913 | 5956 | 6094 | 6096 | 6453 | | | | | | | |
| BIXUB | 7556 # | 7549 | 7550 | 7551 | 7552 | | | | | | | | |
| BIXUB1 | 7565 # | 7561 | 7563 | | | | | | | | | | |
| BLT | 5112 # | 5108 | | | | | | | | | | | |
| BLT-CLEANUP | 5173 # | 8767 | | | | | | | | | | | |
| BLTBU1 | 5227 # | 5208 | | | | | | | | | | | |
| BLTCLR | 5132 # | 5147 | | | | | | | | | | | |
| BLTGOT | 5160 # | 5135 | | | | | | | | | | | |
| BLTGO | 5158 # | 5124 | 5152 | | | | | | | | | | |
| BLTLP | 5155 # | 5170 | | | | | | | | | | | |
| BLTLP1 | 5121 # | 5157 | | | | | | | | | | | |
| BLTX | 5200 # | 7799 | | | | | | | | | | | |
| BLTXLP | 5204 # | 5253 | | | | | | | | | | | |
| BLTXV | 5239 # | 5225 | | | | | | | | | | | |
| BLTXW | 5245 # | 5240 | | | | | | | | | | | |
| BOTH | 2328 # | | | | | | | | | | | | |
| BWRITE | 2467 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 | |
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 4057 | 4071 | 4123 | |
| | 4129 | 4152 | 4154 | 8517 | 8817 | | | | | | | | |
| BYTEAS | 4743 # | 4767 | 6850 | | | | | | | | | | |
| BYTEA | 4745 # | 4700 | 4719 | | | | | | | | | | |
| BYTEAO | 4748 # | 4744 | | | | | | | | | | | |
| BYTFET | 4769 # | 4751 | 4755 | 6892 | 6893 | | | | | | | | |
| BYTIND | 4763 # | 4759 | | | | | | | | | | | |
| CAIM | 3369 # | 3350 | 3351 | 3352 | 3353 | 3354 | 3355 | 3356 | 3358 | 3359 | 3360 | 3361 | |
| | 3362 | 3363 | 3364 | 3365 | | | | | | | | | |
| CHKSN | 5812 # | 5793 | | | | | | | | | | | |
| CLARXL | 7719 # | 5096 | 7493 | | | | | | | | | | |
| CLARXO | 6430 # | 4639 | 4641 | 4645 | 4646 | 6161 | 6295 | 6421 | | | | | |
| CLDISP | 8760 # | 6927 | 6935 | 6948 | 6954 | 8755 | | | | | | | |
| CLEANED | 8763 # | 5178 | 8757 | | | | | | | | | | |
| CLEANUP | 8762 # | 8760 | | | | | | | | | | | |
| CLRB1 | 6281 # | 6279 | | | | | | | | | | | |
| CLRBIN | 6279 # | 6164 | 6260 | | | | | | | | | | |
| CLRFLG | 6917 # | 5960 | 6332 | | | | | | | | | | |
| CLRPTL | 7384 # | 7387 | | | | | | | | | | | |
| CLRPT | 7375 # | | | | | | | | | | | | |
| CLRSN | 5820 # | 5598 | 5688 | | | | | | | | | | |
| CMPDST | 6145 # | 6125 | | | | | | | | | | | |
| CMS | 6093 # | | | | | | | | | | | | |
| CMS2 | 6134 # | 6104 | | | | | | | | | | | |
| CMS3 | 6110 # | 6136 | 6140 | | | | | | | | | | |

| | | | | |
|--------|--------|------|------|-----------|
| CMS4 | 6113 # | | | |
| CMS5 | 6139 # | 6118 | | |
| CMS6 | 6125 # | 6142 | | |
| CMS7 | 6130 # | 6127 | | |
| CMS8 | 6141 # | 6139 | | |
| CMSDST | 6929 # | 8781 | | |
| COMO | 4628 # | 4623 | 4624 | |
| COMOA | 4623 # | 4627 | | |
| COM1 | 4627 # | 4618 | | |
| COM1A | 4618 # | 4626 | | |
| COM2 | 4626 # | 4613 | | |
| COM2A | 4613 # | | | |
| CONSO | 7028 # | 7010 | | |
| CONSZ | 7031 # | 7009 | | |
| CONT | 7857 # | 7852 | | |
| CONT1 | 7864 # | 4044 | | |
| CPYSGN | 4103 # | 4089 | 4099 | |
| DAC | 2488 # | 2495 | 2765 | 6402 |
| DADD | 4083 # | 4078 | | |
| DADD1 | 4086 # | 4092 | | |
| DBABT | 6245 # | 6181 | 6185 | |
| DBDN1 | 6258 # | 6256 | | |
| DBDONE | 6253 # | 6247 | 6272 | |
| DBFAST | 6198 # | 6194 | | |
| DBIN | 6154 # | | | |
| DBIN1 | 6169 # | 6172 | | |
| DBIN2 | 6173 # | 6169 | | |
| DBINLP | 6177 # | 6195 | 6205 | |
| DBLDBL | 6229 # | 6217 | 6228 | |
| DBLDIV | 4579 # | 4433 | 4443 | 5716 5720 |
| DBLMUL | 4223 # | 4194 | | |
| DBLNEG | 4635 # | 2780 | | |
| DBLNGA | 4636 # | 2770 | | |
| DBLNG1 | 4644 # | 4358 | 6304 | |
| DBNEG | 6263 # | 6252 | | |
| DBSLOW | 6210 # | 2181 | 6192 | |
| DBSLO | 6191 # | 6199 | | |
| DBXIT | 6251 # | 6179 | | |
| DDIV | 4385 # | 4380 | | |
| DDIV1 | 4402 # | 4389 | 4393 | |
| DDIV2 | 4405 # | 4403 | | |
| DDIV3A | 4410 # | 4415 | | |
| DDIV3 | 4406 # | 4400 | | |
| DDIV4 | 4418 # | 4411 | | |
| DDIV5A | 4432 # | 4423 | | |
| DDIV5B | 4456 # | 4447 | | |
| DDIV5C | 4458 # | 4453 | | |
| DDIV5 | 4426 # | 4422 | | |
| DDIV6 | 4461 # | 4454 | 4456 | |
| DDIV7 | 4463 # | | | |
| DDIV8A | 4472 # | | | |
| DDIV8 | 4470 # | 4463 | | |
| DDIV9 | 4478 # | 4473 | | |
| DDIVS | 4492 # | 4419 | 4437 | 5712 |
| DFAD | 5546 # | 5533 | | |

| | | | | | |
|--------|--------|------|------|------|-----------|
| DFADJ | 5597 # | 5562 | 5570 | | |
| DFADJ1 | 5604 # | 5605 | | | |
| DFADJ2 | 5606 # | | | | |
| DFADJ3 | 5612 # | 5608 | 5612 | | |
| DFADJ4 | 5613 # | 5609 | | | |
| DFADJ5 | 5615 # | 5607 | 5616 | | |
| DFADJ6 | 5617 # | 5615 | | | |
| DFAS1 | 5555 # | 5551 | 5553 | | |
| DFAS2 | 5562 # | 5556 | | | |
| DFAS3 | 5567 # | 5558 | | | |
| DFAS5 | 5576 # | 5564 | 5572 | | |
| DFAS6 | 5581 # | 5582 | | | |
| DFAS7 | 5583 # | 5581 | | | |
| DFDV | 5686 # | 5683 | | | |
| DFDV1 | 5692 # | 5699 | | | |
| DFDV2 | 5700 # | 5693 | | | |
| DFDV3 | 5711 # | 5703 | | | |
| DFDV4A | 5726 # | 5729 | | | |
| DFDV4B | 5731 # | 5728 | | | |
| DFDV4 | 5713 # | 5709 | | | |
| DFMP | 5627 # | 5622 | | | |
| DFMP1 | 5629 # | 5630 | | | |
| DFMP2 | 5655 # | 5652 | | | |
| DFPR1 | 2421 # | 2415 | | | |
| DFPR2 | 2422 # | 2420 | | | |
| DFSB | 5543 # | 5534 | | | |
| DIV | 4324 # | 4308 | 4309 | 4310 | 4311 |
| DIV1 | 4333 # | 4319 | 4321 | 4375 | |
| DIV2 | 4336 # | 4349 | | | |
| DIVA | 4345 # | 4331 | | | |
| DIVB | 4351 # | 4346 | | | |
| DIVC | 4364 # | 4356 | 4361 | | |
| DIVHI | 4586 # | 4580 | 4583 | 4597 | |
| DIVIDE | 4551 # | 4551 | | | |
| DIVSET | 4545 # | 4541 | | | |
| DIVSGN | 4531 # | 4513 | 5367 | | |
| DIVSUB | 4510 # | 4337 | 4970 | 4980 | 5014 |
| DMLINT | 4231 # | 4196 | | | |
| DMOVNM | 2780 # | 2775 | | | |
| DMOVN | 2770 # | 2766 | | | |
| DMOVN1 | 2782 # | 2774 | | | |
| DMTRAP | 4215 # | 4209 | | | |
| DMUL | 4167 # | 4162 | | | |
| DMUL1 | 4188 # | 4177 | | | |
| DMUL2 | 4193 # | 4187 | | | |
| DMULGO | 4222 # | 4175 | | | |
| DNEG | 5769 # | 5743 | 5746 | 5749 | 5752 |
| DNEG1 | 5771 # | 5769 | | | |
| DNEG2 | 5773 # | 5770 | | | |
| DNN1 | 5799 # | 5795 | | | |
| DNN2 | 5801 # | 5797 | | | |
| DNNORM | 5779 # | 5732 | 5774 | 5776 | 5781 5809 |
| DNNRM1 | 5806 # | 5789 | | | |
| DNORM | 5740 # | 5678 | 5731 | 5742 | 5764 |
| DNORMO | 5677 # | 5587 | | | |

| | | | | | | | | | | | | | | |
|--------|--------|------|------|------|------|------|------|------|------|------|------|------|--|--|
| DNORM1 | 5761 # | 5754 | | | | | | | | | | | | |
| DNORM2 | 5803 # | 5801 | | | | | | | | | | | | |
| DOCVT | 6291 # | 6341 | 6357 | | | | | | | | | | | |
| DOCVT1 | 6366 # | 6291 | | | | | | | | | | | | |
| DOCVT2 | 6387 # | 6367 | | | | | | | | | | | | |
| DOCVT3 | 6384 # | 6406 | | | | | | | | | | | | |
| DONE | 2227 # | 2527 | 2541 | 2841 | 2842 | 3021 | 3174 | 3175 | 3183 | 3184 | 3284 | 3328 | | |
| | 3331 | 3334 | 3349 | 3461 | 3712 | 4218 | 4723 | 5065 | 5140 | 5164 | 5249 | 7031 | | |
| | 7072 | 7126 | 7222 | 7226 | 7230 | 7234 | 7238 | 7285 | 7309 | 7319 | 7370 | 7404 | | |
| | 7543 | | | | | | | | | | | | | |
| DPB | 4712 # | 4685 | | | | | | | | | | | | |
| DPB1 | 4883 # | 4722 | 6874 | | | | | | | | | | | |
| DPB7 | 4890 # | 4884 | 4885 | 4886 | 4887 | 4888 | 4912 | | | | | | | |
| DPBSLO | 4894 # | 4883 | | | | | | | | | | | | |
| DRND1 | 5835 # | 5831 | 5839 | 5842 | | | | | | | | | | |
| DROUND | 5829 # | 5745 | 5748 | 5751 | 5783 | 5785 | 5787 | 5833 | | | | | | |
| DSMS1 | 7517 # | 7514 | 7519 | | | | | | | | | | | |
| DSTEA | 6892 # | 6898 | | | | | | | | | | | | |
| DSTIND | 6897 # | 6894 | 6895 | | | | | | | | | | | |
| DSUB | 4095 # | 4079 | | | | | | | | | | | | |
| DUMP | 7929 # | 7913 | 7920 | | | | | | | | | | | |
| DVSUB1 | 4537 # | 4534 | | | | | | | | | | | | |
| DVSUB2 | 4538 # | 4532 | | | | | | | | | | | | |
| DVSUB3 | 4540 # | 4544 | | | | | | | | | | | | |
| EACALC | 2282 # | 2345 | | | | | | | | | | | | |
| EAPF1 | 8829 # | 8811 | | | | | | | | | | | | |
| EDBYTE | 6629 # | 6592 | 6620 | | | | | | | | | | | |
| EDEXMD | 6540 # | 6511 | | | | | | | | | | | | |
| EDFILL | 6556 # | | | | | | | | | | | | | |
| EDFIL1 | 6571 # | 6558 | | | | | | | | | | | | |
| EDFLT | 6587 # | 6533 | 6577 | | | | | | | | | | | |
| EDFLT1 | 6605 # | 6601 | 6603 | | | | | | | | | | | |
| EDISP | 6470 # | 6464 | | | | | | | | | | | | |
| EDISP1 | 6472 # | 6472 | | | | | | | | | | | | |
| EDIT | 6449 # | | | | | | | | | | | | | |
| EDITLP | 6454 # | 6678 | | | | | | | | | | | | |
| EDMSG | 6614 # | 6479 | | | | | | | | | | | | |
| EDMSG1 | 6623 # | 6616 | | | | | | | | | | | | |
| EDN1A | 6660 # | 6665 | | | | | | | | | | | | |
| EDNOP | 6649 # | 6481 | 6483 | 6485 | 6499 | 6508 | 6534 | 6545 | 6566 | 6572 | 6625 | 6626 | | |
| | 6640 | | | | | | | | | | | | | |
| EDNOP1 | 6650 # | 6644 | | | | | | | | | | | | |
| EDNOP2 | 6667 # | 6661 | 6667 | | | | | | | | | | | |
| EDOPR | 6499 # | 6476 | | | | | | | | | | | | |
| EDSEL | 6550 # | 6503 | | | | | | | | | | | | |
| EDSFLT | 6576 # | 6560 | | | | | | | | | | | | |
| EDSKP | 6637 # | 6488 | 6491 | 6493 | | | | | | | | | | |
| EDSKP1 | 6641 # | 6638 | | | | | | | | | | | | |
| EDSPUT | 6563 # | 6571 | 6581 | | | | | | | | | | | |
| EDSSIG | 6533 # | 6505 | | | | | | | | | | | | |
| EDSTOP | 6518 # | 6501 | 6561 | | | | | | | | | | | |
| EDSTP1 | 6524 # | 6527 | | | | | | | | | | | | |
| ENDSKP | 6261 # | 6085 | | | | | | | | | | | | |
| EQV | 2885 # | 2878 | 2879 | 2880 | 2881 | | | | | | | | | |
| EXCH | 2587 # | 2583 | | | | | | | | | | | | |

| | | | | | | | |
|-----------|--------|------|------|------|------|------|-----------|
| EXTDSP | 5925 # | | | | | | |
| EXTEA | 5924 # | | | | | | |
| EXTEAO | 5921 # | 5919 | | | | | |
| EXTEA1 | 5922 # | 5930 | | | | | |
| EXTEND | 5911 # | 5907 | | | | | |
| EXTXT | 5934 # | 5926 | 5940 | | | | |
| EXTIND | 5930 # | 5927 | | | | | |
| FAD | 5282 # | 5259 | 5260 | 5261 | 5262 | 5263 | 5264 5265 |
| FAS1 | 5284 # | | | | | | |
| FAS2 | 5293 # | 5284 | | | | | |
| FAS3 | 5296 # | 5289 | 5290 | 5293 | 5294 | | |
| FAS4 | 5298 # | 5298 | | | | | |
| FDV | 5345 # | 5333 | 5334 | 5335 | 5337 | 5338 | 5339 5340 |
| FDV0 | 5351 # | 5348 | | | | | |
| FDV1 | 5353 # | 5351 | | | | | |
| FDV2 | 5354 # | 5352 | | | | | |
| FDV3 | 5355 # | 5353 | 5354 | | | | |
| FDV4 | 5357 # | 5358 | | | | | |
| FDV5 | 5360 # | 5357 | | | | | |
| FDV6 | 5363 # | 5360 | | | | | |
| FDV7 | 5367 # | 5368 | | | | | |
| FDV8 | 5370 # | | | | | | |
| FDV9 | 5378 # | 5370 | 5375 | 5376 | | | |
| FETIND | 2338 # | 2303 | 2308 | 2330 | | | |
| FIX | 5423 # | 5418 | 5419 | | | | |
| FIX++ | 4555 # | 4556 | | | | | |
| FIX1++ | 4557 # | 4555 | | | | | |
| FIXL | 5438 # | 5428 | 5438 | | | | |
| FIXPC | 8758 # | 4231 | 6418 | | | | |
| FIXR | 5433 # | 5434 | | | | | |
| FIXT | 5445 # | 5442 | | | | | |
| FIXX | 5442 # | 5435 | | | | | |
| FIXX1 | 5443 # | 5449 | | | | | |
| FL-BWRITE | 2514 # | 5443 | 5447 | 5493 | 5515 | | |
| FLEX | 5493 # | 5489 | | | | | |
| FLTR | 5388 # | 5383 | | | | | |
| FLTR1A | 5396 # | 5393 | | | | | |
| FLTR1 | 5392 # | 5389 | | | | | |
| FLTR2 | 5401 # | 5394 | 5396 | | | | |
| FLTR3 | 5403 # | 5404 | | | | | |
| FMP | 5315 # | 5304 | 5305 | 5306 | 5308 | 5309 | 5310 5311 |
| FMP1 | 5320 # | 5318 | 5319 | | | | |
| FP-LONG | 3930 # | 3908 | 3909 | 3910 | 3911 | 3912 | 3913 |
| FPRO | 2395 # | 2389 | | | | | |
| FPR1 | 2399 # | | | | | | |
| FSB | 5279 # | 5267 | 5268 | 5269 | 5270 | 5271 | 5272 5273 |
| FSC | 5409 # | 5384 | | | | | |
| GETPCW | 7193 # | 3976 | | | | | |
| GETSRC | 6841 # | 6116 | 6551 | | | | |
| GOEXEC | 4000 # | 8831 | | | | | |
| GRP700 | 7034 # | 6995 | | | | | |
| GRP701 | 7013 # | 6996 | | | | | |
| GRP702 | 7207 # | 7203 | | | | | |
| GSRC | 6839 # | 6073 | 6726 | | | | |
| GSRC1 | 6848 # | 6844 | | | | | |

| | | | | | | | | | |
|--------|--------|------|------|------|------|------|------|------|------|
| GTFILL | 6911 # | 5953 | 6450 | | | | | | |
| GTPCW1 | 7196 # | 7187 | 7196 | | | | | | |
| H1 | 7848 # | 7914 | | | | | | | |
| HALT | 3577 # | 3529 | | | | | | | |
| HALTED | 7838 # | 2190 | 2247 | 2265 | 2516 | 3579 | 7468 | 7500 | 8749 |
| HALTLP | 7854 # | 7848 | | | | | | | |
| HARD | 8724 # | 7707 | 8224 | 8226 | 8240 | | | | |
| HLL | 2690 # | 2599 | 2600 | 2643 | 2715 | | | | |
| HLL E | 2756 # | 2629 | 2630 | 2631 | 2632 | | | | |
| HLL O | 2760 # | 2619 | 2620 | 2621 | 2622 | | | | |
| HLL Z | 2758 # | 2609 | 2610 | 2611 | 2612 | | | | |
| HLR | 2717 # | 2646 | 2647 | | | | | | |
| HLRE | 2749 # | 2676 | 2677 | 2678 | 2679 | | | | |
| HLRM | 2726 # | 2648 | | | | | | | |
| HLRO | 2753 # | 2666 | 2667 | 2668 | 2669 | | | | |
| HLRS | 2729 # | 2649 | | | | | | | |
| HLRZ | 2751 # | 2656 | 2657 | 2658 | 2659 | | | | |
| HRL | 2715 # | 2604 | 2605 | | | | | | |
| HRLE | 2742 # | 2634 | 2635 | 2636 | 2637 | | | | |
| HRLM | 2720 # | 2606 | | | | | | | |
| HRLO | 2746 # | 2624 | 2625 | 2626 | 2627 | | | | |
| HRLS | 2723 # | 2607 | | | | | | | |
| HRLZ | 2744 # | 2614 | 2615 | 2616 | 2617 | | | | |
| HRR | 2688 # | 2601 | 2641 | 2642 | 2717 | | | | |
| HRRE | 2735 # | 2671 | 2672 | 2673 | 2674 | | | | |
| HRR O | 2739 # | 2661 | 2662 | 2663 | 2664 | | | | |
| HRRZ | 2737 # | 2651 | 2652 | 2653 | 2654 | | | | |
| HSBDON | 7956 # | | | | | | | | |
| IBP | 4688 # | 4681 | | | | | | | |
| IBPS | 4729 # | 4691 | 4696 | 4710 | | | | | |
| IBPX | 4736 # | 3828 | 4729 | | | | | | |
| IDIV | 4315 # | 4303 | 4304 | 4305 | 4306 | | | | |
| IDPB | 4710 # | 4684 | | | | | | | |
| IDST | 6885 # | 6146 | 6869 | | | | | | |
| IDSTX | 6889 # | 6886 | | | | | | | |
| ILDB | 4696 # | 4682 | | | | | | | |
| IMUL | 4117 # | 4111 | 4112 | 4113 | 4114 | | | | |
| IMUL 1 | 4123 # | 4340 | | | | | | | |
| IMUL 2 | 4125 # | 4122 | | | | | | | |
| IMUL 3 | 4131 # | 4128 | | | | | | | |
| INCAR | 7721 # | | 7473 | | | | | | |
| INCPC | 2275 # | 2270 | | | | | | | |
| INDEX | 2315 # | | | | | | | | |
| INDRCT | 2333 # | | | | | | | | |
| IOEA | 7644 # | 7576 | 7614 | | | | | | |
| IOEA 1 | 7650 # | 7652 | | | | | | | |
| IOEA 2 | 7654 # | 7650 | | | | | | | |
| IOEA I | 7662 # | 7655 | | | | | | | |
| IOEAX | 7671 # | 7658 | | | | | | | |
| IOR | 2865 # | 2858 | 2859 | 2860 | 2861 | 2926 | | | |
| IORD | 7575 # | 7531 | 7542 | 7557 | | | | | |
| IORD 1 | 7584 # | 7580 | | | | | | | |
| IORD 2 | 7594 # | 7589 | | | | | | | |
| IORD 3 | 7600 # | 7595 | 7601 | | | | | | |
| IOT700 | 7793 # | 7736 | 7737 | 7738 | | | | | |

| | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|
| IOT710 | 7795 # | 7740 | 7741 | | | | | | |
| IOT720 | 7802 # | 7743 | 7744 | | | | | | |
| IOT730 | 7804 # | 7746 | 7747 | 7748 | 7749 | 7750 | 7751 | 7752 | 7753 |
| IOT740 | 7806 # | 7755 | 7756 | 7757 | 7758 | 7759 | 7760 | 7761 | 7762 |
| IOT750 | 7808 # | 7764 | 7765 | 7766 | 7767 | 7768 | 7769 | 7770 | 7771 |
| IOT760 | 7810 # | 7773 | 7774 | 7775 | 7776 | 7777 | 7778 | 7779 | 7780 |
| IOT770 | 7812 # | 7782 | 7783 | 7784 | 7785 | 7786 | 7787 | 7788 | 7789 |
| IOW1 | 7684 # | 7689 | | | | | | | |
| IOW2 | 7696 # | 7686 | 7692 | | | | | | |
| IOW3 | 7699 # | 7706 | | | | | | | |
| IOW4 | 7704 # | 7700 | | | | | | | |
| IOW5 | 7707 # | 7694 | | | | | | | |
| IOWAIT | 7680 # | 7585 | 7621 | | | | | | |
| IOWR | 7613 # | 7546 | 7566 | | | | | | |
| IOWR1 | 7619 # | 7629 | 7635 | | | | | | |
| IOWR2 | 7626 # | 7616 | | | | | | | |
| IOWR3 | 7632 # | 7627 | 7633 | | | | | | |
| ITRAP | 8208 # | 6049 | | | | | | | |
| JEN | 3585 # | 3539 | | | | | | | |
| JEN1 | 7514 # | 3591 | 3606 | | | | | | |
| JEN2 | 3591 # | 3583 | | | | | | | |
| JFCL | 3597 # | 3517 | | | | | | | |
| JFFO | 3015 # | 2954 | | | | | | | |
| JFFO1 | 3023 # | 3020 | | | | | | | |
| JFFOL | 3037 # | 3039 | | | | | | | |
| JMPA | 3440 # | 3845 | | | | | | | |
| JRA | 3839 # | 3801 | | | | | | | |
| JRST | 3524 # | 3465 | 3515 | | | | | | |
| JRSTO | 3553 # | 3550 | 3574 | 3589 | | | | | |
| JRST1 | 3571 # | 3566 | 3570 | | | | | | |
| JRST10 | 3582 # | 3536 | | | | | | | |
| JRSTF | 3547 # | 3527 | | | | | | | |
| JSA | 3824 # | 3800 | | | | | | | |
| JSP | 3805 # | 3799 | | | | | | | |
| JSR | 3811 # | 3798 | | | | | | | |
| JSTAC | 3706 # | 3696 | 3703 | 3741 | | | | | |
| JSTAC1 | 3709 # | 3760 | | | | | | | |
| JSYS | 3924 # | 3905 | | | | | | | |
| JUMP | 3472 # | 3462 | 3463 | 3464 | 3466 | 3467 | 3468 | | |
| JUMP-TABLE | 3425 # | 3472 | 3486 | 3500 | 3510 | | | | |
| JUMP- | 3452 # | 3431 | 3434 | 3437 | 3600 | | | | |
| JUMPA | 3456 # | 3440 | 3443 | 3446 | 3449 | 3808 | 4006 | | |
| KIEPT | 8670 # | 8664 | 8667 | | | | | | |
| KIF10 | 8661 # | 8654 | | | | | | | |
| KIF30 | 8678 # | 8659 | 8674 | | | | | | |
| KIF40 | 8685 # | 8689 | | | | | | | |
| KIF50 | 8693 # | 8687 | 8716 | 8731 | | | | | |
| KIF80 | 8714 # | | | | | | | | |
| KIF90 | 8719 # | 8710 | | | | | | | |
| KIFILL | 8649 # | 8280 | | | | | | | |
| KIMUUD | 4012 # | 3955 | | | | | | | |
| KIUPT | 8655 # | 8677 | | | | | | | |
| L-BDEC | 5885 # | 5858 | 5859 | | | | | | |
| L-CMS | 5879 # | 5848 | 5849 | 5850 | 5852 | 5853 | 5854 | | |
| L-DBIN | 5883 # | 5856 | 5857 | | | | | | |

| | | | | | | | | | | | | | | |
|--------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|--|
| MVABT1 | 6008 # | 6010 | | | | | | | | | | | | |
| MVABT2 | 6012 # | 6008 | | | | | | | | | | | | |
| MVEND | 6015 # | | | | | | | | | | | | | |
| MVS | 5950 # | | | | | | | | | | | | | |
| MVS1 | 5967 # | 5965 | | | | | | | | | | | | |
| MVSK2 | 6055 # | 6029 | | | | | | | | | | | | |
| MVSK3 | 6045 # | 6041 | | | | | | | | | | | | |
| MVSKP | 6028 # | 6021 | 6039 | 6052 | 6063 | | | | | | | | | |
| MVSKP1 | 6039 # | 6034 | | | | | | | | | | | | |
| MVSKP2 | 6051 # | 6043 | | | | | | | | | | | | |
| MVSO | 5979 # | 5972 | | | | | | | | | | | | |
| MVSO1 | 5982 # | 5982 | | | | | | | | | | | | |
| NEXT | 4025 # | 4013 | 7930 | 7933 | 7935 | 7937 | 7940 | 7943 | 7946 | 7949 | 7952 | 7954 | | |
| NEXTAR | 8833 # | 8797 | | | | | | | | | | | | |
| NICOND | 2236 # | 2521 | 3042 | 3088 | 3116 | 3428 | 3710 | 3786 | 3793 | 4481 | 4484 | 5439 | | |
| | 5445 | 5804 | 8201 | | | | | | | | | | | |
| NICOND-FETCH | 2255 # | 2227 | 2228 | 2230 | 2231 | 2291 | 2297 | 3325 | 3337 | 3452 | 3453 | 3456 | | |
| | 3457 | 3524 | 3525 | 3551 | 3593 | 4217 | 4692 | 7351 | 7622 | | | | | |
| NIDISP | 3088 # | 3788 | 3792 | 4338 | 4373 | 4404 | 4490 | 4707 | 4997 | 5349 | 5361 | 5426 | | |
| | 5714 | | | | | | | | | | | | | |
| NODDIV | 4489 # | 4426 | 4428 | | | | | | | | | | | |
| NODIV | 4373 # | 4367 | 4371 | | | | | | | | | | | |
| NOMOD | 2319 # | | | | | | | | | | | | | |
| NXTWRD | 4735 # | 4731 | | | | | | | | | | | | |
| ORCA | 2905 # | 2898 | 2899 | 2900 | 2901 | | | | | | | | | |
| ORCB | 2936 # | 2929 | 2930 | 2931 | 2932 | | | | | | | | | |
| ORCM | 2926 # | 2919 | 2920 | 2921 | 2922 | | | | | | | | | |
| PAGE-FAIL | 8206 # | | | | | | | | | | | | | |
| PF120 | 8519 # | 8499 | | | | | | | | | | | | |
| PF125 | 8538 # | 8536 | 8742 | | | | | | | | | | | |
| PF130 | 8501 # | 8497 | 8503 | | | | | | | | | | | |
| PF140 | 8495 # | | | | | | | | | | | | | |
| PF25 | 8263 # | 8265 | | | | | | | | | | | | |
| PFD | 8214 # | 8216 | | | | | | | | | | | | |
| PFDBIN | 6952 # | 8779 | | | | | | | | | | | | |
| PFDONE | 8491 # | 8717 | 8721 | | | | | | | | | | | |
| PFGACO | 6954 # | | | | | | | | | | | | | |
| PFMAP | 8244 # | 8200 | 8228 | 8232 | 8234 | | | | | | | | | |
| PFPI1 | 8733 # | 8218 | 8230 | | | | | | | | | | | |
| PFPI2 | 8741 # | | | | | | | | | | | | | |
| PFT | 8749 # | 8693 | | | | | | | | | | | | |
| PFT1 | 8750 # | 8738 | | | | | | | | | | | | |
| PFT10 | 8793 # | 8791 | | | | | | | | | | | | |
| PFT1A | 8757 # | 8753 | | | | | | | | | | | | |
| PFT2 | 8789 # | 8765 | | | | | | | | | | | | |
| PFT3 | 8792 # | 8818 | | | | | | | | | | | | |
| PI | 7417 # | 7419 | 8792 | | | | | | | | | | | |
| PI10 | 7435 # | 7427 | 7428 | 7429 | 7430 | 7431 | 7432 | 7433 | | | | | | |
| PI40 | 7453 # | 7450 | | | | | | | | | | | | |
| PI50 | 7454 # | 7504 | | | | | | | | | | | | |
| PIEXIT | 7349 # | 3621 | | | | | | | | | | | | |
| PIJSR | 7468 # | 7457 | | | | | | | | | | | | |
| PIP1 | 7427 # | 7420 | | | | | | | | | | | | |
| PIP2 | 7428 # | 7421 | | | | | | | | | | | | |
| PIP3 | 7429 # | 7422 | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|--|--|--|--|--|--|--|--|
| SNNORM | 5500 # | 5496 | 5498 | 5500 | 5504 | | | | | | | | | | | | | | | | |
| SNNOT | 5511 # | 5507 | | | | | | | | | | | | | | | | | | | |
| SNNOT1 | 5514 # | 5509 | | | | | | | | | | | | | | | | | | | |
| SNNOT2 | 5515 # | 5511 | 5513 | 5514 | | | | | | | | | | | | | | | | | |
| SNORM | 5480 # | 5299 | 5328 | 5405 | 5480 | 5492 | | | | | | | | | | | | | | | |
| SNORMO | 5405 # | 5378 | 5390 | 5397 | 5412 | 5413 | | | | | | | | | | | | | | | |
| SNORM1 | 5492 # | 5488 | | | | | | | | | | | | | | | | | | | |
| SOJ | 3500 # | 3489 | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 | | | | | | | | | | | | |
| SOS | 3419 # | 3408 | 3409 | 3410 | 3411 | 3412 | 3413 | 3414 | 3415 | | | | | | | | | | | | |
| SRCMOD | 6724 # | 5992 | 6177 | 6819 | 6823 | 6825 | | | | | | | | | | | | | | | |
| SRND1 | 5525 # | 5522 | | | | | | | | | | | | | | | | | | | |
| SROUND | 5522 # | 5482 | 5484 | 5486 | 5501 | 5502 | 5503 | 5523 | | | | | | | | | | | | | |
| SSWEEP | 7406 # | 7379 | 7392 | | | | | | | | | | | | | | | | | | |
| STAC | 2520 # | 2489 | 2533 | 2538 | 2546 | 2547 | 2562 | 2591 | 2771 | 2972 | 2985 | 2994 | | | | | | | | | |
| | 5758 | 5765 | 5810 | 6016 | 6524 | 7827 | | | | | | | | | | | | | | | |
| STAC34 | 6271 # | 6266 | | | | | | | | | | | | | | | | | | | |
| START | 2249 # | 3820 | 3835 | | | | | | | | | | | | | | | | | | |
| STBOTH | 2531 # | | | | | | | | | | | | | | | | | | | | |
| STBTH1 | 2536 # | 2484 | | | | | | | | | | | | | | | | | | | |
| STDBTH | 2493 # | | | | | | | | | | | | | | | | | | | | |
| STMAC | 3693 # | 3685 | | | | | | | | | | | | | | | | | | | |
| STMEM | 2525 # | 7834 | | | | | | | | | | | | | | | | | | | |
| STOBR | 7727 # | 3630 | 3979 | 7471 | | | | | | | | | | | | | | | | | |
| STOPC | 7729 # | 3633 | 7847 | | | | | | | | | | | | | | | | | | |
| STORE | 2539 # | 2791 | 3641 | | | | | | | | | | | | | | | | | | |
| STPF1A | 6950 # | 6942 | | | | | | | | | | | | | | | | | | | |
| STRPF | 6938 # | 8771 | 8773 | | | | | | | | | | | | | | | | | | |
| STRPFO | 6939 # | 6956 | | | | | | | | | | | | | | | | | | | |
| STRPF1 | 6942 # | | | | | | | | | | | | | | | | | | | | |
| STRPF2 | 6953 # | 6950 | | | | | | | | | | | | | | | | | | | |
| STRPF3 | 6947 # | | | | | | | | | | | | | | | | | | | | |
| STRPF4 | 6956 # | 8777 | | | | | | | | | | | | | | | | | | | |
| STRIO | 7725 # | 7441 | 7489 | 7863 | | | | | | | | | | | | | | | | | |
| STSELF | 2483 # | 2549 | | | | | | | | | | | | | | | | | | | |
| STUBRS | 7131 # | 7133 | | | | | | | | | | | | | | | | | | | |
| SUB | 4070 # | 4063 | 4064 | 4065 | 4066 | | | | | | | | | | | | | | | | |
| SWEPL | 7397 # | 7400 | | | | | | | | | | | | | | | | | | | |
| SWEEP | 7391 # | 7143 | 7169 | | | | | | | | | | | | | | | | | | |
| T1LSH | 5242 # | 5213 | 5233 | | | | | | | | | | | | | | | | | | |
| TDONE | 3299 # | 3290 | 3293 | | | | | | | | | | | | | | | | | | |
| TDX | 3270 # | 3178 | 3187 | 3192 | 3196 | 3201 | 3205 | 3210 | 3214 | 3219 | 3223 | 3227 | | | | | | | | | |
| | 3231 | 3236 | 3240 | | | | | | | | | | | | | | | | | | |
| TDXX | 3275 # | 3176 | 3180 | 3185 | 3189 | 3194 | 3198 | 3203 | 3207 | 3212 | 3216 | 3221 | | | | | | | | | |
| | 3225 | 3229 | 3233 | 3238 | 3242 | | | | | | | | | | | | | | | | |
| TENLP | 2179 # | 2186 | | | | | | | | | | | | | | | | | | | |
| TEST-TABLE | 3281 # | 3270 | 3276 | 7532 | | | | | | | | | | | | | | | | | |
| TICK | 7245 # | 6058 | | | | | | | | | | | | | | | | | | | |
| TIOX | 7531 # | 7524 | 7525 | 7526 | 7527 | | | | | | | | | | | | | | | | |
| TOCK | 7248 # | 7302 | 8737 | | | | | | | | | | | | | | | | | | |
| TOCK1 | 7252 # | 7258 | | | | | | | | | | | | | | | | | | | |
| TOCK2 | 7259 # | 7253 | | | | | | | | | | | | | | | | | | | |
| TOCK3 | 7263 # | 7270 | | | | | | | | | | | | | | | | | | | |
| TRAP | 6963 # | 2240 | 2243 | 2246 | 2258 | 2261 | 2264 | | | | | | | | | | | | | | |
| TRNAR | 6745 # | 6554 | | | | | | | | | | | | | | | | | | | |
| TRNFNC | 6767 # | 6755 | 6774 | 6778 | 6782 | 6794 | | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | |
|---------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| TRNNS1 | 6817 # | 6807 | | | | | | | | | | | |
| TRNNS2 | 6820 # | 6817 | | | | | | | | | | | |
| TRNRET | 6799 # | 6768 | | | | | | | | | | | |
| TRNSIG | 6780 # | 6790 | | | | | | | | | | | |
| TRNSS | 6806 # | 6801 | | | | | | | | | | | |
| TRNSS1 | 6802 # | 6810 | | | | | | | | | | | |
| TRP1 | 6982 # | 6972 | 6980 | | | | | | | | | | |
| TSX | 3268 # | 3179 | 3188 | 3193 | 3197 | 3202 | 3206 | 3211 | 3215 | 3220 | 3224 | 3228 | |
| | 3232 | 3237 | 3241 | | | | | | | | | | |
| TSXX | 3273 # | 3177 | 3181 | 3186 | 3190 | 3195 | 3199 | 3204 | 3208 | 3213 | 3217 | 3222 | |
| | 3226 | 3230 | 3234 | 3239 | 3243 | | | | | | | | |
| TXXX | 3284 # | 3299 | | | | | | | | | | | |
| TXZX | 3298 # | 3287 | | | | | | | | | | | |
| UMOVEM | 7830 # | 7818 | | | | | | | | | | | |
| UMOVE | 7822 # | 7817 | | | | | | | | | | | |
| UUD | 3936 # | 3897 | 3904 | | | | | | | | | | |
| UUD101 | 3918 # | 3898 | | | | | | | | | | | |
| UUD102 | 3920 # | 3899 | | | | | | | | | | | |
| UUD103 | 3922 # | 3900 | | | | | | | | | | | |
| UUD106 | 3926 # | 3906 | | | | | | | | | | | |
| UUD107 | 3928 # | 3907 | | | | | | | | | | | |
| UUD247 | 3932 # | 3914 | | | | | | | | | | | |
| UUDFLG | 4021 # | 4015 | | | | | | | | | | | |
| UUDGO | 3942 # | 2434 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 | 3585 | 3605 | |
| | 3626 | 3639 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 | 6907 | 6990 | |
| | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 | 7014 | 7015 | 7018 | 7019 | |
| | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7217 | 7239 | 7334 | 7793 | 7802 | 7804 | |
| | 7806 | 7808 | 7810 | 7812 | | | | | | | | | |
| UUOPCW | 3973 # | 4018 | | | | | | | | | | | |
| VECINT | 7482 # | 7445 | 7485 | | | | | | | | | | |
| VECIN1 | 7501 # | 7499 | | | | | | | | | | | |
| WRAPR | 7049 # | | | | | | | | | | | | |
| WRAPR1 | 7068 # | 7080 | | | | | | | | | | | |
| WRAPR2 | 7074 # | 7067 | | | | | | | | | | | |
| WRCSE | 7224 # | | | | | | | | | | | | |
| WRCSTM | 7232 # | | | | | | | | | | | | |
| WREBR | 7146 # | | | | | | | | | | | | |
| WREBR1 | 7148 # | 7148 | | | | | | | | | | | |
| WRHSB | 7236 # | | | | | | | | | | | | |
| WRINT | 7315 # | | | | | | | | | | | | |
| WRIO | 7546 # | 7536 | 7538 | | | | | | | | | | |
| WRPI | 7331 # | | | | | | | | | | | | |
| WRPUR | 7228 # | | | | | | | | | | | | |
| WRSPB | 7220 # | | | | | | | | | | | | |
| WRTHSB | 7917 # | 7841 | | | | | | | | | | | |
| WRTIME | 7275 # | | | | | | | | | | | | |
| WRTIM1 | 7287 # | 7256 | 7283 | | | | | | | | | | |
| WRUBR | 7101 # | | | | | | | | | | | | |
| XCT | 3650 # | 3646 | | | | | | | | | | | |
| XCT1 | 3662 # | 3656 | 6989 | 7867 | | | | | | | | | |
| XCT1A | 3652 # | 3660 | | | | | | | | | | | |
| XCT2 | 2342 # | 3663 | | | | | | | | | | | |
| XCTGO | 2267 # | 2251 | 7859 | | | | | | | | | | |
| XJEN | 3605 # | 3533 | | | | | | | | | | | |
| XJURSTF | 3612 # | 3532 | 3635 | | | | | | | | | | |

Cross Reference Listing

| | | | | | | | | | | | | |
|----------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 3270 | 3276 | 3369 |
| | 3388 | 3389 | 3472 | 3486 | 3500 | 3510 | 3695 | 3702 | 3740 | 4057 | 4071 | 4123 |
| | 4129 | 4152 | 4154 | 5206 | 5435 | 5443 | 5447 | 5493 | 5505 | 5515 | 5522 | 5926 |
| | 5968 | 6112 | 6167 | 6374 | 6728 | 6800 | 6807 | 7532 | 7559 | 7577 | 7588 | 7615 |
| | 8517 | 8817 | | | | | | | | | | |
| BAD PARITY | 1954 # | | | | | | | | | | | |
| BASIC DIV STEP | 4527 # | 4548 | 4551 | | | | | | | | | |
| BWRITE DISP | 2071 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 |
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 4057 | 4071 | 4123 |
| | 4129 | 4152 | 4154 | 8517 | 8817 | | | | | | | |
| BYTE DISP | 2088 # | 4704 | 4721 | 6149 | 6852 | 6873 | | | | | | |
| BYTE STEP | 4676 # | 5980 | 5982 | | | | | | | | | |
| CALL IBP | 4670 # | 4691 | 4696 | 4710 | | | | | | | | |
| CALL LOAD PI | 2094 # | 3592 | 7349 | | | | | | | | | |
| CALL [] | 1984 # | 2181 | 2193 | 2398 | 2770 | 2780 | 3550 | 3589 | 3591 | 3592 | 3606 | 3630 |
| | 3633 | 3828 | 3831 | 3945 | 3976 | 3979 | 4013 | 4015 | 4120 | 4148 | 4175 | 4184 |
| | 4194 | 4204 | 4227 | 4337 | 4358 | 4413 | 4419 | 4433 | 4437 | 4443 | 4489 | 4513 |
| | 4691 | 4696 | 4700 | 4705 | 4710 | 4719 | 4722 | 4970 | 4980 | 5014 | 5041 | 5096 |
| | 5112 | 5114 | 5201 | 5213 | 5215 | 5233 | 5235 | 5322 | 5364 | 5367 | 5372 | 5374 |
| | 5482 | 5484 | 5486 | 5501 | 5502 | 5503 | 5562 | 5570 | 5598 | 5638 | 5643 | 5655 |
| | 5662 | 5669 | 5688 | 5712 | 5716 | 5720 | 5745 | 5748 | 5751 | 5783 | 5785 | 5787 |
| | 5793 | 5913 | 5953 | 5956 | 5960 | 5992 | 5998 | 6023 | 6047 | 6058 | 6073 | 6077 |
| | 6084 | 6094 | 6096 | 6102 | 6116 | 6125 | 6146 | 6161 | 6164 | 6177 | 6192 | 6203 |
| | 6211 | 6214 | 6217 | 6220 | 6228 | 6233 | 6260 | 6295 | 6304 | 6320 | 6332 | 6359 |
| | 6370 | 6378 | 6395 | 6421 | 6450 | 6453 | 6458 | 6533 | 6541 | 6551 | 6554 | 6564 |
| | 6577 | 6592 | 6599 | 6620 | 6624 | 6692 | 6726 | 6750 | 6850 | 6869 | 6874 | 6926 |
| | 6934 | 6945 | 7187 | 7256 | 7280 | 7283 | 7302 | 7349 | 7379 | 7392 | 7441 | 7471 |
| | 7473 | 7489 | 7493 | 7531 | 7542 | 7557 | 7576 | 7585 | 7614 | 7621 | 7646 | 7839 |
| | 7841 | 7847 | 7863 | 7911 | 7913 | 7930 | 7933 | 7935 | 7937 | 7940 | 7943 | 7946 |
| | 7949 | 7952 | 7954 | 8726 | 8737 | 8797 | | | | | | |
| CHANGE FLAGS | 2004 # | 2216 | 2217 | 2577 | 3403 | 3419 | 3486 | 3500 | 3556 | 3557 | 3560 | 3561 |
| | 3597 | 3618 | 3619 | 3689 | 3701 | 3757 | 3787 | 3791 | 3806 | 3812 | 4003 | 4004 |
| | 4005 | 4057 | 4071 | 4088 | 4098 | 4101 | 4131 | 4132 | 4156 | 4206 | 4218 | 4228 |
| | 4231 | 4338 | 4373 | 4404 | 4490 | 4639 | 4641 | 4706 | 4723 | 4745 | 4997 | 5349 |
| | 5361 | 5426 | 5714 | 6388 | 6389 | 6987 | 7459 | 7474 | 8755 | | | |
| CHK PARITY | 1953 # | 2268 | 2338 | 2352 | 2361 | 2365 | 2382 | 2393 | 2404 | 2410 | 2423 | 2494 |
| | 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | 3086 | 3405 | 3548 | 3572 | 3587 | 3612 |
| | 3616 | 3677 | 3694 | 3700 | 3722 | 3739 | 3745 | 3758 | 3816 | 3844 | 3974 | 4001 |
| | 4013 | 4017 | 4040 | 4118 | 4211 | 4213 | 4316 | 4326 | 4410 | 4432 | 4439 | 4736 |
| | 4764 | 4770 | 4891 | 5156 | 5205 | 5911 | 5930 | 6350 | 6540 | 6590 | 6593 | 6621 |
| | 6897 | 6912 | 6982 | 7104 | 7221 | 7225 | 7229 | 7233 | 7237 | 7277 | 7307 | 7309 |
| | 7316 | 7370 | 7454 | 7496 | 7620 | 7667 | 7712 | 7715 | 7717 | 7727 | 7729 | 7826 |
| | 7845 | 7865 | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 |
| | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7955 | 8679 | 8796 | 8810 | | |
| CHK PARITY L | 1946 # | | | | | | | | | | | |
| CHK PARITY R | 1950 # | | | | | | | | | | | |
| CLEANUP DISP | 2096 # | 8760 | | | | | | | | | | |
| CLEAR ARXO | 1702 # | 2770 | 4635 | 6263 | 6304 | 6430 | 6436 | | | | | |
| CLEAR CONTINUE | 1993 # | | | | | | | | | | | |
| CLEAR EXECUTE | 1994 # | | | | | | | | | | | |
| CLEAR RUN | 1995 # | | | | | | | | | | | |
| CLEAR []O | 1701 # | 2770 | 4357 | 4635 | 6240 | 6256 | 6263 | 6304 | 6430 | 6436 | | |
| CLR FPD | 2013 # | 3689 | 3806 | 3812 | 4206 | 4706 | 4723 | 6389 | | | | |
| CLR IO BUSY | 1999 # | 7575 | 7613 | | | | | | | | | |
| CLR IO LATCH | 2000 # | 7660 | 7668 | 7672 | 7675 | 7684 | 7690 | 7699 | 7704 | | | |

| | | | | | | | | | | | | | |
|----------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| CLRCSH | 1990 # | 7381 | 7382 | 7385 | | | | | | | | | |
| DFADJ | 5594 # | 5605 | | | | | | | | | | | |
| DISMISS | 2093 # | 3591 | 3606 | | | | | | | | | | |
| DIV | 1968 # | 4548 | 4551 | 4552 | 4599 | 5480 | 5492 | 5500 | 5504 | 5597 | 5599 | 5600 | |
| DIV DISP | 2087 # | | | | | | | | | | | | |
| DIV STEP | 4528 # | 4551 | | | | | | | | | | | |
| DONE | 2079 # | 2227 | 2231 | 3325 | 3452 | 3457 | 4217 | 4692 | 7351 | 7622 | | | |
| DPB SCAD | 4879 # | 4884 | 4885 | 4886 | 4887 | 4888 | | | | | | | |
| EA MODE DISP | 2068 # | 2278 | 2344 | 3553 | 4743 | 4746 | 5922 | 6898 | 7656 | | | | |
| END BLT | 2036 # | 5139 | 5163 | 5249 | | | | | | | | | |
| END MAP | 2037 # | 8507 | | | | | | | | | | | |
| END STATE | 2034 # | 5139 | 5163 | 5249 | 6016 | 6088 | 6111 | 6261 | 6389 | 6393 | 6455 | 6727 | |
| | 8507 | 8764 | 8782 | 8783 | 8814 | | | | | | | | |
| EXIT | 2073 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 | |
| | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 4123 | 4129 | 4152 | |
| | 4154 | 8517 | 8817 | | | | | | | | | | |
| EXP TEST | 2025 # | 5489 | 5505 | 5835 | | | | | | | | | |
| FETCH | 1805 # | 2227 | 2228 | 2230 | 2231 | 2250 | 2290 | 2296 | 2404 | 2429 | 3019 | 3325 | |
| | 3337 | 3452 | 3453 | 3456 | 3457 | 3524 | 3525 | 3551 | 3593 | 3708 | 4217 | 4692 | |
| | 4701 | 7351 | 7622 | 7858 | | | | | | | | | |
| FE_-1 | 1925 # | | | | | | | | | | | | |
| FE_-12. | 1923 # | 3023 | | | | | | | | | | | |
| FE_-2 | 1922 # | | | | | | | | | | | | |
| FE_-FE | 1909 # | 4808 | 4902 | | | | | | | | | | |
| FE_-FE+200 | 1936 # | 5353 | 5354 | | | | | | | | | | |
| FE_-FE+S# | 1912 # | 4813 | | | | | | | | | | | |
| FE_-FE-1 | 1910 # | 2965 | 3000 | | | | | | | | | | |
| FE_-S-10 | 1919 # | 4817 | 4883 | | | | | | | | | | |
| FE_-S-20 | 1918 # | | | | | | | | | | | | |
| FE_O | 1924 # | | | | | | | | | | | | |
| FE_EXP | 1931 # | 2396 | 2411 | 5288 | 5569 | | | | | | | | |
| FE_FE+1 | 1926 # | 2968 | 2982 | 3003 | 5368 | 5484 | 5486 | 5502 | 5503 | 5523 | 5528 | 5748 | |
| | 5751 | 5785 | 5787 | 5833 | 5841 | | | | | | | | |
| | 1928 # | 4815 | 4820 | 4896 | 4904 | | | | | | | | |
| FE_FE+2 | 1927 # | 5325 | | | | | | | | | | | |
| FE_FE+4 | 1930 # | 3038 | | | | | | | | | | | |
| FE_FE+P | 1934 # | 6520 | | | | | | | | | | | |
| FE_FE+S# | 1937 # | 5667 | 5678 | 6653 | | | | | | | | | |
| FE_FE+SC | 1913 # | 6652 | | | | | | | | | | | |
| FE_FE-1 | 1929 # | 5480 | 5492 | 5500 | 5504 | 5584 | 5586 | 5741 | 5763 | 5780 | 5808 | | |
| FE_FE-19 | 1911 # | | | | | | | | | | | | |
| FE_FE-200 | 1935 # | 5327 | | | | | | | | | | | |
| FE_FE.AND.S# | 1914 # | 4703 | 4900 | 6148 | 6526 | 6651 | 6853 | | | | | | |
| FE_P | 1915 # | 4699 | 4899 | 6650 | 6850 | 6890 | | | | | | | |
| FE_S | 1916 # | | | | | | | | | | | | |
| FE_S# | 1920 # | 3023 | 5212 | 5214 | 5232 | 5234 | 5390 | 5394 | 5396 | 5397 | 6519 | | |
| FE_S#-FE | 1921 # | 5727 | | | | | | | | | | | |
| FE_S+2 | 1917 # | 5979 | | | | | | | | | | | |
| FE_SC+EXP | 1932 # | 5316 | 5411 | 5650 | | | | | | | | | |
| FE_SC-EXP | 1933 # | 5347 | 5701 | | | | | | | | | | |
| FIRST DIV STEP | 4529 # | 4548 | | | | | | | | | | | |
| FIX [] SIGN | 1715 # | 3386 | 4103 | 4127 | 4150 | 6731 | 7250 | | | | | | |
| FL NO DIVIDE | 2029 # | 5349 | 5361 | 5714 | | | | | | | | | |
| FL-EXIT | 2075 # | 5443 | 5447 | 5493 | 5515 | | | | | | | | |
| FM WRITE | 1722 # | 2159 | 2167 | 2169 | 2170 | 2172 | 2173 | 2175 | 2176 | 2182 | 2184 | 2488 | |

| | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 2520 | 2771 | 3021 | 3042 | 3077 | 3110 | 3116 | 3299 | 3388 | 3428 | 3431 | 3434 |
| 3437 | 3440 | 3443 | 3446 | 3449 | 3662 | 3709 | 3711 | 3774 | 3779 | 3807 | 4178 |
| 4186 | 4195 | 4197 | 4208 | 4209 | 4210 | 4212 | 4214 | 4436 | 4452 | 4453 | 4454 |
| 4456 | 4459 | 4478 | 4480 | 4483 | 4610 | 4615 | 4620 | 4626 | 4627 | 4628 | 4689 |
| 4706 | 4948 | 4961 | 4971 | 4981 | 5013 | 5064 | 5113 | 5177 | 5202 | 5439 | 5445 |
| 5757 | 5765 | 5801 | 5802 | 5803 | 5810 | 5921 | 5934 | 5937 | 5962 | 5967 | 5984 |
| 6008 | 6028 | 6029 | 6035 | 6045 | 6048 | 6055 | 6056 | 6057 | 6072 | 6083 | 6085 |
| 6087 | 6103 | 6119 | 6121 | 6128 | 6142 | 6155 | 6162 | 6173 | 6204 | 6221 | 6223 |
| 6258 | 6259 | 6271 | 6272 | 6279 | 6281 | 6294 | 6296 | 6305 | 6308 | 6340 | 6346 |
| 6352 | 6357 | 6358 | 6362 | 6371 | 6377 | 6384 | 6385 | 6386 | 6545 | 6576 | 6605 |
| 6678 | 6694 | 6724 | 6820 | 6848 | 6889 | 6913 | 6927 | 6930 | 6935 | 6942 | 6947 |
| 6954 | 6964 | 7071 | 7156 | 7222 | 7226 | 7230 | 7234 | 7238 | 7252 | 7263 | 7284 |
| 7287 | 7317 | 7318 | 7402 | 7403 | 7543 | 7651 | 7685 | 7691 | 7693 | 7731 | 7838 |
| 7910 | 7925 | 7935 | 8197 | 8201 | 8207 | 8208 | 8210 | 8211 | 8218 | 8228 | 8230 |
| 8232 | 8234 | 8238 | 8239 | 8724 | | | | | | | |
| 1785 | # | 7455 | 7469 | 7503 | | | | | | | |
| 1940 | # | 4785 | | | | | | | | | |
| 1718 | # | 4819 | 4822 | 4895 | 4898 | 5982 | | | | | |
| 2095 | # | 2190 | 2247 | 2265 | 2516 | 3579 | 7468 | 7500 | 8749 | | |
| 1944 | # | 2228 | 2230 | 2275 | 2288 | 2289 | 2295 | 2300 | 2315 | 2316 | 2328 |
| 2362 | | 2690 | 2721 | 2729 | 2751 | 2753 | 3337 | 3453 | 3456 | 3524 | 3525 |
| 3535 | | 3538 | 3551 | 3563 | 3593 | 3617 | 3631 | 3759 | 3818 | 3830 | 3834 |
| 3990 | | 3995 | 4025 | 4041 | 4092 | 4735 | 4765 | 4974 | 5033 | 5056 | 5105 |
| 5139 | | 5143 | 5148 | 5163 | 5166 | 5176 | 5228 | 5229 | 5249 | 5251 | 5324 |
| 5924 | | 5927 | 5930 | 5971 | 5997 | 6000 | 6016 | 6022 | 6051 | 6070 | 6076 |
| 6088 | | 6099 | 6111 | 6115 | 6122 | 6141 | 6169 | 6246 | 6261 | 6299 | 6334 |
| 6378 | | 6389 | 6393 | 6449 | 6455 | 6502 | 6559 | 6563 | 6598 | 6623 | 6664 |
| 6727 | | 6846 | 6888 | 6897 | 7051 | 7077 | 7091 | 7138 | 7152 | 7154 | 7198 |
| 7283 | | 7308 | 7342 | 7344 | 7346 | 7348 | 7360 | 7420 | 7421 | 7422 | 7423 |
| 7425 | | 7426 | 7448 | 7518 | 7644 | 7665 | 7665 | 7846 | 7862 | 7932 | 7939 |
| 7945 | | 7948 | 7951 | 8200 | 8507 | 8515 | 8758 | 8764 | 8770 | 8772 | 8774 |
| 8778 | | 8780 | 8782 | 8783 | 8784 | 8814 | 8833 | | | | |
| 1948 | # | 2419 | 2421 | 2434 | 2456 | 2457 | 2688 | 2723 | 2727 | 2744 | 2746 |
| 3113 | | 3115 | 3386 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 |
| 3605 | | 3626 | 3639 | 3770 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 |
| 3940 | | 4022 | 4038 | 4103 | 4104 | 4105 | 4127 | 4150 | 4357 | 4396 | 4458 |
| 4635 | | 4812 | 4947 | 4955 | 5018 | 5063 | 5098 | 5102 | 5289 | 5290 | 5293 |
| 5318 | | 5319 | 5351 | 5352 | 5412 | 5413 | 5489 | 5505 | 5608 | 5609 | 5616 |
| 5754 | | 5789 | 5819 | 5820 | 5835 | 5917 | 5939 | 6166 | 6171 | 6240 | 6256 |
| 6263 | | 6302 | 6304 | 6311 | 6317 | 6326 | 6345 | 6397 | 6400 | 6405 | 6430 |
| 6507 | | 6601 | 6603 | 6673 | 6677 | 6731 | 6773 | 6777 | 6781 | 6785 | 6789 |
| 6809 | | 6907 | 6919 | 6990 | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 |
| 7014 | | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7045 |
| 7088 | | 7109 | 7114 | 7124 | 7129 | 7136 | 7141 | 7182 | 7189 | 7217 | 7239 |
| 7334 | | 7336 | 7338 | 7340 | 7472 | 7674 | 7719 | 7793 | 7802 | 7804 | 7806 |
| 7810 | | 7812 | 8193 | 8196 | 8247 | 8251 | 8254 | 8259 | 8506 | 8510 | 8513 |
| 8699 | | 8704 | 8709 | 8715 | 8729 | 8792 | | | | | |
| 4667 | # | 4691 | 4696 | 4710 | 6032 | 6842 | 6885 | 6958 | | | |
| 4668 | # | 4691 | 4696 | 4710 | 6032 | 6842 | 6885 | | | | |
| 4669 | # | 4691 | 4696 | 4710 | | | | | | | |
| 1956 | # | 3509 | 3679 | 3726 | 3755 | 3775 | 3780 | 5029 | 5104 | | |
| 2072 | # | 2353 | 2357 | 2366 | 2371 | 2379 | 2400 | 2406 | 2430 | 2435 | |
| 2085 | # | | | | | | | | | | |
| 2022 | # | 3597 | | | | | | | | | |
| 2078 | # | 3472 | 3486 | 3500 | 3510 | | | | | | |

FORCE EXEC
 GEN 17-FE
 GEN MSK []
 HALT []
 HOLD LEFT

HOLD RIGHT

IBP DP
 IBP SCAD
 IBP SPEC
 INH CRY18
 INST DISP
 INTERRUPT TRAP
 JFCL FLAGS
 JUMP DISP

Produced on Advanced Information Services Electronic Laser Printer, PK011E6, DTN: 223-7881

| | | | | | | | | | | | | | |
|-----------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| JUMPA | 2080 # | 3453 | 3456 | 3524 | 3525 | 3551 | 3593 | | | | | | |
| LDB SCAD | 4781 # | 4790 | 4791 | 4792 | 4793 | 4794 | | | | | | | |
| LEAVE USER | 2020 # | 2216 | 4004 | | | | | | | | | | |
| LOAD AC BLOCKS | 1988 # | 2162 | 7125 | 7142 | | | | | | | | | |
| LOAD BYTE EA | 1979 # | 4699 | 4718 | 4766 | 5918 | 5930 | 6849 | | | | | | |
| LOAD DST EA | 1982 # | 6886 | 6888 | 6897 | | | | | | | | | |
| LOAD FE | 1862 # | 2378 | 2396 | 2411 | 2965 | 2968 | 2972 | 2982 | 2984 | 3000 | 3003 | 3006 | |
| | 3011 | 3023 | 3038 | 3084 | 4699 | 4703 | 4715 | 4808 | 4813 | 4815 | 4817 | 4820 | |
| | 4883 | 4896 | 4899 | 4900 | 4902 | 4904 | 5212 | 5214 | 5232 | 5234 | 5242 | 5243 | |
| | 5288 | 5316 | 5325 | 5327 | 5347 | 5353 | 5354 | 5368 | 5390 | 5394 | 5396 | 5397 | |
| | 5411 | 5480 | 5484 | 5486 | 5492 | 5500 | 5502 | 5503 | 5504 | 5523 | 5528 | 5569 | |
| | 5584 | 5586 | 5650 | 5667 | 5678 | 5701 | 5727 | 5741 | 5748 | 5751 | 5763 | 5780 | |
| | 5785 | 5787 | 5808 | 5833 | 5841 | 5979 | 5980 | 5982 | 6148 | 6519 | 6520 | 6526 | |
| | 6650 | 6651 | 6652 | 6653 | 6850 | 6853 | 6872 | 6890 | | | | | |
| LOAD FLAGS | 2024 # | 2217 | 3556 | 3560 | 3618 | 4003 | 7459 | 7474 | 8755 | | | | |
| LOAD IND EA | 1980 # | 2340 | | | | | | | | | | | |
| LOAD INST | 1977 # | 2269 | 3654 | 6983 | 7866 | | | | | | | | |
| LOAD INST EA | 1978 # | 3549 | 3573 | 3588 | 7663 | | | | | | | | |
| LOAD IR | 1973 # | 5917 | | | | | | | | | | | |
| LOAD PAGE TABLE | 1987 # | 7376 | 8533 | | | | | | | | | | |
| LOAD PCU | 2018 # | 4005 | | | | | | | | | | | |
| LOAD PI | 1991 # | 3607 | 7362 | 7437 | 7919 | | | | | | | | |
| LOAD PXCT | 1985 # | 3659 | | | | | | | | | | | |
| LOAD SC | 1861 # | 2174 | 2184 | 2377 | 2396 | 2411 | 2427 | 2988 | 2990 | 3066 | 3067 | 3068 | |
| | 3070 | 3074 | 3084 | 3093 | 3096 | 3100 | 3107 | 3138 | 3139 | 3140 | 3142 | 3146 | |
| | 4118 | 4146 | 4183 | 4203 | 4223 | 4249 | 4261 | 4267 | 4275 | 4285 | 4293 | 4336 | |
| | 4433 | 4441 | 4548 | 4551 | 4591 | 4937 | 4940 | 4953 | 4959 | 4964 | 4977 | 4993 | |
| | 5038 | 5053 | 5059 | 5282 | 5286 | 5287 | 5296 | 5298 | 5320 | 5367 | 5401 | 5404 | |
| | 5409 | 5427 | 5428 | 5429 | 5431 | 5434 | 5438 | 5548 | 5558 | 5568 | 5602 | 5605 | |
| | 5612 | 5627 | 5630 | 5638 | 5642 | 5645 | 5655 | 5661 | 5716 | 5717 | 6293 | 6322 | |
| | 6461 | 6463 | 6466 | 6470 | 6472 | 6643 | 6649 | 6655 | 6660 | 6667 | 7128 | 7132 | |
| | 7146 | 7148 | 7177 | 7179 | 7187 | 7194 | 7196 | 7386 | 7399 | 7406 | 7594 | 7601 | |
| | 7626 | 7632 | 7680 | 7687 | 7696 | 7700 | 8261 | 8264 | 8496 | 8502 | | | |
| LOAD SRC EA | 1981 # | | | | | | | | | | | | |
| LOAD VMA | 1784 # | 2160 | 2179 | 2183 | 2227 | 2227 | 2228 | 2230 | 2231 | 2231 | 2249 | 2275 | |
| | 2290 | 2296 | 2302 | 2306 | 2329 | 2333 | 2342 | 2362 | 2404 | 2414 | 2417 | 2429 | |
| | 2783 | 2787 | 2788 | 3019 | 3325 | 3325 | 3337 | 3452 | 3452 | 3453 | 3456 | 3457 | |
| | 3457 | 3524 | 3525 | 3526 | 3531 | 3535 | 3538 | 3551 | 3564 | 3567 | 3593 | 3608 | |
| | 3614 | 3629 | 3631 | 3631 | 3640 | 3680 | 3707 | 3717 | 3729 | 3734 | 3750 | 3813 | |
| | 3841 | 3953 | 3954 | 3954 | 3978 | 3978 | 3997 | 3998 | 4025 | 4025 | 4034 | 4041 | |
| | 4042 | 4217 | 4217 | 4692 | 4692 | 4701 | 4748 | 4752 | 4756 | 4760 | 5093 | 5121 | |
| | 5129 | 5143 | 5144 | 5148 | 5149 | 5166 | 5167 | 5245 | 5251 | 5251 | 5925 | 5927 | |
| | 5928 | 5951 | 6099 | 6348 | 6375 | 6420 | 6426 | 6433 | 6449 | 6454 | 6510 | 6588 | |
| | 6630 | 6745 | 6892 | 6893 | 6894 | 6895 | 6971 | 6979 | 7101 | 7278 | 7303 | 7308 | |
| | 7308 | 7351 | 7351 | 7369 | 7375 | 7438 | 7453 | 7453 | 7453 | 7454 | 7495 | 7495 | |
| | 7503 | 7584 | 7619 | 7622 | 7622 | 7644 | 7666 | 7725 | 7822 | 7830 | 7840 | 7843 | |
| | 7843 | 7846 | 7846 | 7857 | 7913 | 7929 | 7929 | 7932 | 7932 | 7939 | 7939 | 7942 | |
| | 7942 | 7945 | 7945 | 7948 | 7948 | 7951 | 7958 | 8244 | 8244 | 8525 | 8541 | 8656 | |
| | 8657 | 8671 | 8673 | 8739 | 8745 | 8793 | 8794 | 8794 | 8830 | 8830 | 8833 | 8833 | |
| LSH | 1961 # | | | | | | | | | | | | |
| LSHC | 1964 # | 3070 | 3074 | 4580 | 4583 | | | | | | | | |
| LUUO | 2082 # | 5879 | 5881 | 5883 | 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | | | |
| MEM CYCLE | 1781 # | 2160 | 2179 | 2183 | 2227 | 2227 | 2227 | 2228 | 2228 | 2230 | 2230 | 2231 | |
| | 2231 | 2231 | 2249 | 2250 | 2267 | 2275 | 2290 | 2290 | 2296 | 2296 | 2301 | 2302 | |
| | 2306 | 2307 | 2329 | 2330 | 2333 | 2334 | 2338 | 2342 | 2351 | 2360 | 2362 | 2363 | |

| | | | | | | | | | | | | |
|-----------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 2364 | 2392 | 2403 | 2404 | 2404 | 2414 | 2415 | 2417 | 2418 | 2422 | 2429 | 2429 |
| | 2493 | 2525 | 2531 | 2536 | 2539 | 2588 | 2589 | 2783 | 2784 | 2786 | 2787 | 2788 |
| | 2790 | 3019 | 3019 | 3325 | 3325 | 3325 | 3337 | 3337 | 3404 | 3405 | 3452 | 3452 |
| | 3452 | 3453 | 3453 | 3456 | 3456 | 3457 | 3457 | 3457 | 3524 | 3524 | 3525 | 3525 |
| | 3526 | 3526 | 3531 | 3531 | 3535 | 3535 | 3538 | 3538 | 3547 | 3551 | 3551 | 3564 |
| | 3565 | 3567 | 3568 | 3571 | 3586 | 3593 | 3593 | 3608 | 3612 | 3614 | 3615 | 3616 |
| | 3629 | 3629 | 3631 | 3631 | 3632 | 3635 | 3640 | 3640 | 3676 | 3680 | 3681 | 3693 |
| | 3699 | 3707 | 3708 | 3717 | 3718 | 3721 | 3729 | 3731 | 3734 | 3736 | 3738 | 3744 |
| | 3750 | 3752 | 3758 | 3813 | 3814 | 3815 | 3825 | 3841 | 3842 | 3843 | 3953 | 3954 |
| | 3954 | 3973 | 3978 | 3978 | 3978 | 3997 | 3998 | 3999 | 4000 | 4012 | 4016 | 4025 |
| | 4025 | 4025 | 4034 | 4035 | 4039 | 4041 | 4042 | 4043 | 4217 | 4217 | 4217 | 4692 |
| | 4692 | 4692 | 4701 | 4701 | 4729 | 4735 | 4736 | 4748 | 4749 | 4752 | 4753 | 4756 |
| | 4757 | 4760 | 4761 | 4763 | 4769 | 4890 | 4891 | 5093 | 5094 | 5121 | 5122 | 5129 |
| | 5130 | 5132 | 5143 | 5144 | 5146 | 5148 | 5149 | 5151 | 5155 | 5158 | 5166 | 5167 |
| | 5169 | 5204 | 5245 | 5245 | 5247 | 5251 | 5251 | 5252 | 5911 | 5925 | 5927 | 5927 |
| | 5928 | 5928 | 5930 | 5951 | 5952 | 6099 | 6099 | 6348 | 6349 | 6350 | 6375 | 6376 |
| | 6420 | 6426 | 6433 | 6449 | 6449 | 6454 | 6454 | 6510 | 6510 | 6542 | 6544 | 6588 |
| | 6588 | 6590 | 6593 | 6621 | 6630 | 6630 | 6745 | 6746 | 6892 | 6892 | 6893 | 6893 |
| | 6894 | 6894 | 6895 | 6895 | 6897 | 6911 | 6967 | 6970 | 6971 | 6975 | 6978 | 6979 |
| | 6982 | 7101 | 7102 | 7103 | 7220 | 7221 | 7224 | 7225 | 7228 | 7229 | 7232 | 7233 |
| | 7236 | 7237 | 7275 | 7276 | 7278 | 7279 | 7303 | 7304 | 7307 | 7308 | 7308 | 7308 |
| | 7309 | 7315 | 7316 | 7351 | 7351 | 7351 | 7369 | 7369 | 7370 | 7375 | 7438 | 7442 |
| | 7453 | 7453 | 7453 | 7454 | 7454 | 7469 | 7490 | 7495 | 7495 | 7496 | 7503 | 7504 |
| | 7584 | 7586 | 7619 | 7620 | 7622 | 7622 | 7622 | 7644 | 7645 | 7666 | 7666 | 7667 |
| | 7712 | 7715 | 7717 | 7725 | 7727 | 7729 | 7822 | 7823 | 7825 | 7830 | 7831 | 7840 |
| | 7843 | 7843 | 7845 | 7846 | 7846 | 7846 | 7857 | 7858 | 7864 | 7913 | 7929 | 7929 |
| | 7930 | 7931 | 7932 | 7932 | 7932 | 7933 | 7934 | 7937 | 7938 | 7939 | 7939 | 7939 |
| | 7940 | 7941 | 7942 | 7942 | 7942 | 7943 | 7944 | 7945 | 7945 | 7945 | 7946 | 7947 |
| | 7948 | 7948 | 7948 | 7949 | 7950 | 7951 | 7951 | 7952 | 7952 | 7953 | 7955 | 7958 |
| | 8244 | 8525 | 8541 | 8656 | 8657 | 8658 | 8671 | 8672 | 8673 | 8678 | 8739 | 8745 |
| | 8793 | 8794 | 8794 | 8795 | 8809 | 8830 | 8830 | 8833 | 8833 | 8833 | | |
| MEM READ | 1809 # | 2267 | 2338 | 2351 | 2360 | 2364 | 2392 | 2403 | 2422 | 3547 | 3571 | 3586 |
| | 3612 | 3616 | 3676 | 3721 | 3758 | 3843 | 4000 | 4763 | 4769 | 5155 | 5204 | 5911 |
| | 5930 | 6350 | 6593 | 6621 | 6897 | 6911 | 6967 | 6975 | 6982 | 7103 | 7221 | 7225 |
| | 7229 | 7233 | 7237 | 7276 | 7316 | 7442 | 7454 | 7490 | 7496 | 7586 | 7667 | 7712 |
| | 7715 | 7717 | 7825 | 7864 | 8678 | | | | | | | |
| MEM WAIT | 1808 # | 2267 | 2338 | 2351 | 2360 | 2364 | 2392 | 2403 | 2422 | 2493 | 2525 | 2531 |
| | 2536 | 2539 | 2589 | 2786 | 3405 | 3547 | 3571 | 3586 | 3612 | 3616 | 3676 | 3693 |
| | 3699 | 3721 | 3738 | 3744 | 3758 | 3815 | 3843 | 3973 | 4000 | 4012 | 4016 | 4039 |
| | 4736 | 4763 | 4769 | 4891 | 5132 | 5155 | 5158 | 5204 | 5247 | 5911 | 5930 | 6350 |
| | 6544 | 6590 | 6593 | 6621 | 6897 | 6911 | 6967 | 6975 | 6982 | 7103 | 7221 | 7225 |
| | 7229 | 7233 | 7237 | 7276 | 7307 | 7309 | 7316 | 7370 | 7442 | 7454 | 7490 | 7496 |
| | 7586 | 7620 | 7667 | 7712 | 7715 | 7717 | 7727 | 7729 | 7825 | 7845 | 7864 | 7930 |
| | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 | 7946 | 7947 | 7949 |
| | 7950 | 7952 | 7953 | 7955 | 8678 | 8795 | 8809 | | | | | |
| MEM WRITE | 1810 # | 2493 | 2525 | 2531 | 2536 | 2539 | 2589 | 2786 | 3405 | 3693 | 3699 | 3738 |
| | 3744 | 3815 | 3973 | 4012 | 4016 | 4039 | 4736 | 4891 | 5132 | 5158 | 5247 | 6544 |
| | 6590 | 7307 | 7309 | 7370 | 7620 | 7727 | 7729 | 7845 | 7930 | 7931 | 7933 | 7934 |
| | 7937 | 7938 | 7940 | 7941 | 7943 | 7944 | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 |
| | 7955 | 8795 | 8809 | | | | | | | | | |
| MEM_Q | 1817 # | 5133 | 5159 | 5247 | 6544 | | | | | | | |
| MEM_[] | 1816 # | 2494 | 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | 3405 | 3694 | 3700 | 3739 |
| | 3745 | 3816 | 3974 | 4013 | 4017 | 4040 | 4736 | 4891 | 6590 | 7307 | 7309 | 7370 |
| | 7620 | 7727 | 7729 | 7845 | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 |
| | 7943 | 7944 | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7955 | 8796 | 8810 | |

| | | | | | | | | | | | | | |
|------------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| SC_8. | 1897 # | | | | | | | | | | | | |
| SC_9. | 1896 # | 4937 | 5053 | | | | | | | | | | |
| SC_EXP | 1877 # | 2396 | 2411 | 5286 | 5558 | | | | | | | | |
| SC_FE | 1880 # | 2427 | 5645 | | | | | | | | | | |
| SC_FE+S# | 1879 # | 3093 | 5427 | | | | | | | | | | |
| SC_S# | 1881 # | 2174 | 4118 | 4146 | 4183 | 4203 | 4223 | 4336 | 4433 | 4441 | 4937 | 4953 | |
| | 4964 | 4977 | 4993 | 5038 | 5053 | 5320 | 5367 | 5401 | 5627 | 5638 | 5642 | 5655 | |
| | 5661 | 5716 | 5717 | 6293 | 6461 | 6463 | 6470 | 6649 | 6660 | 7128 | 7146 | 7177 | |
| | 7187 | 7194 | 7406 | 7594 | 7626 | 7680 | 7696 | 8261 | 8496 | | | | |
| SC_S#-FE | 1878 # | 3096 | 5429 | | | | | | | | | | |
| SC_SC-1 | 1866 # | 7687 | | | | | | | | | | | |
| SC_SC-EXP | 1873 # | | | | | | | | | | | | |
| SC_SC-EXP-1 | 1874 # | 5282 | 5548 | | | | | | | | | | |
| SC_SC-FE | 1876 # | | | | | | | | | | | | |
| SC_SC-FE-1 | 1875 # | 5287 | 5568 | | | | | | | | | | |
| SC_SHIFT | 1867 # | 3084 | 5409 | | | | | | | | | | |
| SC_SHIFT-1 | 1868 # | 2377 | | | | | | | | | | | |
| SC_SHIFT-2 | 1869 # | | | | | | | | | | | | |
| SET APR ENABLES | 1997 # | 2165 | 7070 | 7155 | | | | | | | | | |
| SET AROV | 2006 # | 4131 | 4132 | 4156 | 4218 | 5426 | | | | | | | |
| SET FL NO DIVIDE | 2009 # | 5349 | 5361 | 5714 | | | | | | | | | |
| SET FOV | 2007 # | | | | | | | | | | | | |
| SET FPD | 2012 # | 4231 | 4745 | 6388 | | | | | | | | | |
| SET HALT | 1992 # | 7848 | 7914 | | | | | | | | | | |
| SET NO DIVIDE | 2008 # | 4338 | 4373 | 4404 | 4490 | 4997 | 5349 | 5361 | 5714 | | | | |
| SET PDL OV | 2015 # | 3701 | 3757 | 3787 | 3791 | | | | | | | | |
| SET P TO 36-S | 4672 # | 4731 | 6042 | 6845 | 6887 | | | | | | | | |
| SET TRAP1 | 2016 # | | | | | | | | | | | | |
| SHIFT | 1864 # | 2972 | 2984 | 3006 | 3011 | 5242 | 5243 | | | | | | |
| SKIP AC REF | 2060 # | | | | | | | | | | | | |
| SKIP AD.EQ.O | 2044 # | 3016 | 3331 | 3343 | 3434 | 3446 | 4121 | 4126 | 4328 | 4330 | 4334 | 4348 | |
| | 4393 | 4422 | 4452 | 4533 | 4611 | 4616 | 4621 | 4637 | 4644 | 4962 | 4991 | 5345 | |
| | 5371 | 5373 | 5446 | 5488 | 5512 | 5668 | 5719 | 5761 | 5806 | 6136 | 6189 | 6265 | |
| | 6268 | 6557 | 6594 | 6615 | 6738 | 7295 | | | | | | | |
| SKIP AD.LE.O | 2045 # | 3334 | 3346 | 3437 | 3449 | 4387 | 5021 | 5365 | 5689 | 6336 | 7261 | 7918 | |
| SKIP ADL.EQ.O | 2056 # | 2187 | 3986 | 3991 | 5388 | 5393 | 5507 | 5672 | 5675 | 5743 | 5746 | 5749 | |
| | 5752 | 5812 | 5912 | 5955 | 6094 | 6096 | 6178 | 6198 | 6382 | 6398 | 6452 | 6487 | |
| | 6490 | 6522 | 6986 | 7110 | 7117 | 7153 | 7456 | 7457 | 7648 | 7650 | 7664 | 8248 | |
| | 8511 | 8530 | 8532 | 8693 | 8711 | 8752 | | | | | | | |
| SKIP ADL.LE.O | 2047 # | 7657 | | | | | | | | | | | |
| SKIP ADR.EQ.O | 2057 # | 3085 | 3591 | 3606 | 5119 | 6323 | 6396 | 6751 | 7028 | 7031 | 7056 | 7058 | |
| | 7061 | 7063 | 7065 | 7331 | 7333 | 7337 | 7339 | 7341 | 7343 | 7345 | 7347 | 7444 | |
| | 7497 | 7517 | 7589 | 7616 | 8199 | 8491 | 8680 | 8697 | 8700 | 8705 | | | |
| SKIP CRYO | 2052 # | 3684 | 3727 | 3756 | 5357 | 5481 | 5483 | 5485 | 5487 | | | | |
| SKIP CRY1 | 2053 # | 4084 | 4096 | 5797 | 6213 | 6219 | 6232 | 6423 | 6434 | | | | |
| SKIP CRY2 | 2054 # | 5770 | 5830 | | | | | | | | | | |
| SKIP DPO | 2042 # | 2397 | 2412 | 2571 | 2749 | 2756 | 3039 | 3111 | 3328 | 3340 | 3431 | 3443 | |
| | 3773 | 3776 | 3781 | 4103 | 4128 | 4150 | 4153 | 4208 | 4216 | 4317 | 4353 | 4355 | |
| | 4360 | 4365 | 4369 | 4391 | 4408 | 4420 | 4427 | 4442 | 4461 | 4479 | 4531 | 4538 | |
| | 4553 | 4994 | 5138 | 5162 | 5248 | 5284 | 5288 | 5317 | 5347 | 5355 | 5389 | 5411 | |
| | 5442 | 5549 | 5651 | 5698 | 5702 | 5713 | 5726 | 5963 | 5976 | 5993 | 6006 | 6013 | |
| | 6031 | 6073 | 6075 | 6097 | 6114 | 6117 | 6123 | 6163 | 6255 | 6260 | 6297 | 6306 | |
| | 6339 | 6356 | 6428 | 6478 | 6504 | 6695 | 6725 | 6767 | 6821 | 6940 | 7105 | 7250 | |
| | 8650 | | | | | | | | | | | | |
| SKIP DP18 | 2043 # | 2376 | 2735 | 2742 | 5936 | 6183 | 6387 | 6476 | 6799 | 6800 | 7483 | 8653 | |

Produced on Advanced Information Services Electronic Laser Printer, PKO IES6, DTN: 223-7881

Cross Reference Listing

| | | | | | | | | | | | | | |
|---------------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| | 8663 | 8686 | | | | | | | | | | | |
| SKIP EXECUTE | 2061 # | 7852 | | | | | | | | | | | |
| SKIP FPD | 2049 # | 4170 | 4691 | 4696 | 4710 | 6287 | | | | | | | |
| SKIP IF ACO | 2041 # | 2483 | 3387 | 4688 | | | | | | | | | |
| SKIP IO LEGAL | 2051 # | 2433 | 3536 | 3539 | | | | | | | | | |
| SKIP IRPT | 2058 # | 5142 | 6033 | 8212 | 8733 | 8764 | 8815 | | | | | | |
| SKIP JFCL | 2055 # | 3598 | | | | | | | | | | | |
| SKIP KERNEL | 2050 # | 3529 | 3533 | 3534 | 3541 | 3650 | 6965 | | | | | | |
| SKIP -1MS | 2059 # | 6028 | | | | | | | | | | | |
| SKIP-COMP DISP | 2077 # | 3369 | 3388 | 3389 | 6112 | | | | | | | | |
| SPEC MEM READ | 1811 # | 2409 | | | | | | | | | | | |
| SPEC MEM WRITE | 1812 # | | | | | | | | | | | | |
| START NO TEST WRITE | 1804 # | | | | | | | | | | | | |
| START READ | 1802 # | 2227 | 2228 | 2230 | 2231 | 2250 | 2290 | 2296 | 2301 | 2307 | 2330 | 2334 | |
| | 2363 | 2404 | 2415 | 2418 | 2429 | 3019 | 3325 | 3337 | 3452 | 3453 | 3456 | 3457 | |
| | 3524 | 3525 | 3526 | 3531 | 3535 | 3538 | 3551 | 3565 | 3568 | 3593 | 3615 | 3635 | |
| | 3708 | 3718 | 3752 | 3842 | 3999 | 4043 | 4217 | 4692 | 4701 | 4749 | 4753 | 4757 | |
| | 4761 | 5094 | 5169 | 5252 | 5927 | 5928 | 5952 | 6099 | 6349 | 6376 | 6449 | 6454 | |
| | 6510 | 6630 | 6746 | 6892 | 6893 | 6894 | 6895 | 6970 | 6978 | 7102 | 7220 | 7224 | |
| | 7228 | 7232 | 7236 | 7275 | 7279 | 7315 | 7351 | 7504 | 7622 | 7645 | 7666 | 7823 | |
| | 7858 | 8658 | 8672 | | | | | | | | | | |
| START WRITE | 1803 # | 2588 | 2784 | 2790 | 3404 | 3629 | 3632 | 3640 | 3681 | 3731 | 3736 | 3814 | |
| | 3825 | 3978 | 4025 | 4035 | 4729 | 4735 | 4890 | 5122 | 5130 | 5146 | 5151 | 5245 | |
| | 6542 | 6588 | 7304 | 7308 | 7369 | 7469 | 7831 | 7846 | 7932 | 7939 | 7942 | 7945 | |
| | 7948 | 7951 | 8833 | | | | | | | | | | |
| STATE_[] | 2033 # | 5105 | 5971 | 5997 | 6000 | 6022 | 6070 | 6076 | 6084 | 6115 | 6122 | 6141 | |
| | 6169 | 6351 | 6378 | 6502 | 6559 | 6563 | 6623 | 8200 | 8770 | 8772 | 8774 | 8776 | |
| | 8778 | 8780 | 8784 | | | | | | | | | | |
| STEP SC | 1863 # | 2184 | 2988 | 2990 | 3066 | 3068 | 3070 | 3074 | 3100 | 3107 | 3138 | 3140 | |
| | 3142 | 3146 | 4249 | 4261 | 4267 | 4275 | 4285 | 4293 | 4548 | 4551 | 4591 | 4940 | |
| | 4959 | 5059 | 5296 | 5298 | 5404 | 5428 | 5431 | 5434 | 5438 | 5602 | 5605 | 5612 | |
| | 5630 | 6322 | 6466 | 6472 | 6667 | 7132 | 7148 | 7179 | 7196 | 7386 | 7399 | 7601 | |
| | 7632 | 7700 | 8264 | 8502 | | | | | | | | | |
| SWEEP | 1989 # | 7394 | 7395 | 7398 | | | | | | | | | |
| TAKE INTERRUPT | 2084 # | 8792 | | | | | | | | | | | |
| TEST DISP | 2076 # | 3270 | 3276 | 7532 | | | | | | | | | |
| TL [] | 1711 # | 3986 | 3991 | 5507 | 5672 | 5675 | 5743 | 5746 | 5749 | 5752 | 5812 | 5912 | |
| | 5955 | 6094 | 6096 | 6178 | 6198 | 6382 | 6398 | 6452 | 6487 | 6490 | 7648 | 7650 | |
| | 7664 | 8248 | 8511 | 8530 | 8532 | 8693 | 8711 | 8752 | | | | | |
| TR [] | 1710 # | 3591 | 3606 | 6323 | 6396 | 6751 | 7056 | 7058 | 7061 | 7063 | 7065 | 7331 | |
| | 7333 | 7337 | 7339 | 7341 | 7343 | 7345 | 7347 | 7589 | 7616 | 8199 | 8491 | 8680 | |
| | 8697 | 8700 | 8705 | | | | | | | | | | |
| TURN OFF PXCT | 1986 # | 2239 | 2242 | 2245 | 2257 | 2260 | 2263 | 2277 | | | | | |
| TXXX TEST | 2062 # | 3284 | | | | | | | | | | | |
| UNHALT | 1996 # | 7851 | | | | | | | | | | | |
| UPDATE USER | 2019 # | 3557 | 3561 | 3619 | | | | | | | | | |
| UUO | 2081 # | 2434 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 | 3585 | 3605 | |
| | 3626 | 3639 | 3918 | 3920 | 3922 | 3924 | 3926 | 3928 | 3930 | 3932 | 6907 | 6990 | |
| | 6999 | 7000 | 7001 | 7005 | 7006 | 7007 | 7008 | 7013 | 7014 | 7015 | 7018 | 7019 | |
| | 7020 | 7021 | 7022 | 7023 | 7024 | 7025 | 7217 | 7239 | 7334 | 7793 | 7802 | 7804 | |
| | 7806 | 7808 | 7810 | 7812 | | | | | | | | | |
| VMA | 1851 # | | | | | | | | | | | | |
| VMA EXTENDED | 1789 # | | | | | | | | | | | | |
| VMA PHYSICAL | 1786 # | 3954 | 3978 | 3998 | 4025 | 6971 | 6979 | 7453 | 7495 | 7843 | 7846 | 7929 | |
| | 7932 | 7939 | 7942 | 7945 | 7948 | 7951 | 8657 | 8673 | 8794 | 8830 | 8833 | | |

| | | | | | | | | | | | | | |
|------------------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| VMA PHYSICAL READ | 1788 # | 7453 | 7495 | 8830 | | | | | | | | | |
| VMA PHYSICAL WRITE | 1787 # | 3954 | 7843 | 7929 | 8794 | | | | | | | | |
| VMA_[] | 1831 # | 2227 | 2231 | 2249 | 2306 | 2333 | 2342 | 2429 | 2787 | 3325 | 3452 | 3457 | |
| | 3531 | 3567 | 3629 | 3640 | 3729 | 3734 | 3813 | 3841 | 4217 | 4692 | 4701 | 4752 | |
| | 4760 | 5093 | 5121 | 5129 | 5245 | 5928 | 6454 | 6510 | 6588 | 6630 | 6893 | 6895 | |
| | 7101 | 7351 | 7369 | 7375 | 7453 | 7622 | 7666 | 7822 | 7830 | 7857 | 7958 | 8525 | |
| | 8541 | 8793 | | | | | | | | | | | |
| VMA_[] WITH FLAGS | 1832 # | 7725 | | | | | | | | | | | |
| VMA_[]+1 | 1834 # | 2228 | 2230 | 2275 | 2362 | 3337 | 3631 | 4041 | 5143 | 5148 | 5166 | 5251 | |
| | 6099 | 6449 | 7278 | 7308 | | | | | | | | | |
| VMA_[]+XR | 1836 # | 4748 | 4756 | 6892 | 6894 | | | | | | | | |
| VMA_[]+[] | 1837 # | | | | | | | | | | | | |
| VMA_[]-1 | 1835 # | 3526 | 3535 | 3538 | 7644 | | | | | | | | |
| VMA_[] .OR. [] WITH FL | 1833 # | 7584 | 7619 | | | | | | | | | | |
| WORK[] | 1852 # | 6131 | 6353 | 6552 | 6619 | 6697 | 6728 | 6731 | | | | | |
| WORK[] .NOT. [] | 1762 # | 6173 | | | | | | | | | | | |
| WORK[]_O | 1757 # | 2176 | 7402 | 7403 | | | | | | | | | |
| WORK[]_1 | 1758 # | 2175 | | | | | | | | | | | |
| WORK[]_Q | 1755 # | 4971 | 6103 | 6142 | | | | | | | | | |
| WORK[]_[] | 1759 # | 2159 | 2167 | 2169 | 2170 | 4689 | 4948 | 4961 | 4981 | 5013 | 5921 | 5934 | |
| | 5937 | 5967 | 6029 | 6055 | 6056 | 6057 | 6072 | 6119 | 6155 | 6294 | 6296 | 6358 | |
| | 6371 | 6377 | 6576 | 6724 | 6913 | 6964 | 7071 | 7156 | 7222 | 7226 | 7230 | 7234 | |
| | 7238 | 7252 | 7263 | 7284 | 7287 | 7317 | 7318 | 7685 | 7691 | 7693 | 7838 | 7910 | |
| | 7925 | 7935 | 8197 | 8207 | 8208 | 8210 | 8211 | 8218 | 8228 | 8230 | 8232 | 8234 | |
| | 8238 | 8239 | 8724 | | | | | | | | | | |
| WORK[]_[] CLR LH | 1760 # | 3662 | 7651 | | | | | | | | | | |
| WORK[]_[]-1 | 1761 # | 6352 | | | | | | | | | | | |
| WORK[]_[] .AND. [] | 1763 # | 5984 | | | | | | | | | | | |
| WRITE TEST | 1801 # | 2588 | 2784 | 2790 | 3404 | 3629 | 3632 | 3640 | 3681 | 3731 | 3736 | 3814 | |
| | 3825 | 3978 | 4025 | 4035 | 4729 | 4735 | 4890 | 5122 | 5130 | 5146 | 5151 | 5245 | |
| | 6542 | 6588 | 7304 | 7308 | 7369 | 7469 | 7831 | 7846 | 7932 | 7939 | 7942 | 7945 | |
| | 7948 | 7951 | 8833 | | | | | | | | | | |
| XR | 1850 # | | | | | | | | | | | | |
| [] LEFT_-1 | 1652 # | 2739 | | | | | | | | | | | |
| [] LEFT_O | 1650 # | 2737 | | | | | | | | | | | |
| [] RIGHT_-1 | 1653 # | 2760 | | | | | | | | | | | |
| [] RIGHT_O | 1651 # | 2758 | | | | | | | | | | | |
| []+[] | 1517 # | 2782 | 4228 | 4368 | | | | | | | | | |
| []-# | 1519 # | 6182 | 6475 | 8651 | 8661 | | | | | | | | |
| []-[] | 1518 # | 4364 | 4420 | 4421 | 5118 | 5136 | 5160 | 5248 | 5357 | 5365 | 5713 | 6097 | |
| [] .AND. # | 1520 # | 5388 | 5392 | 6521 | 6984 | 7115 | 7153 | | | | | | |
| [] .AND. NOT. WORK[] | 1764 # | | | | | | | | | | | | |
| [] .AND. NOT. [] | 1523 # | 4125 | | | | | | | | | | | |
| [] .AND. Q | 1521 # | 4392 | | | | | | | | | | | |
| [] .AND. WORK[] | 1765 # | 6737 | | | | | | | | | | | |
| [] .AND. [] | 1522 # | 4153 | 7028 | 7031 | 7517 | | | | | | | | |
| [] .OR. [] | 1524 # | | | | | | | | | | | | |
| [] .XOR. # | 1525 # | 2187 | 7456 | 7457 | | | | | | | | | |
| [] .XOR. [] | 1526 # | 7294 | | | | | | | | | | | |
| []_# | 1528 # | 2148 | 2152 | 2177 | 2178 | 2218 | 2744 | 2746 | 2751 | 2753 | 4537 | 4945 | |
| | 4968 | 4978 | 4982 | 5011 | 5054 | 5105 | 5939 | 5971 | 5997 | 6000 | 6022 | 6070 | |
| | 6076 | 6084 | 6115 | 6122 | 6141 | 6169 | 6351 | 6378 | 6502 | 6559 | 6563 | 6623 | |
| | 7035 | 7037 | 7378 | 7380 | 7391 | 7393 | 7472 | 7514 | 8200 | 8770 | 8772 | 8774 | |
| | 8776 | 8778 | 8780 | 8784 | | | | | | | | | |
| []_#-[] | 1527 # | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | |
|---------------------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|--|--|--|--|--|--|--|--|
| []_ (#-[])*2 | 1614 # | 5049 | | | | | | | | | | | | | | | | | | | |
| []_ (-[])*.5 | 1615 # | | | | | | | | | | | | | | | | | | | | |
| []_ (-[]-.25)*.5 LONG | 1616 # | 4398 | 5707 | | | | | | | | | | | | | | | | | | |
| []_ (-[]-.25)*2 LONG | 1617 # | 4475 | | | | | | | | | | | | | | | | | | | |
| []_ (AC[] .AND. [])*.5 | 1612 # | 4174 | 4176 | 5546 | 5637 | 5692 | | | | | | | | | | | | | | | |
| []_ (MEM.AND.[])*.5 | 1826 # | 2423 | | | | | | | | | | | | | | | | | | | |
| []_ (Q+1)*.5 | 1613 # | 5829 | | | | | | | | | | | | | | | | | | | |
| []_ ([]+#)*2 | 1627 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([]+1)*2 | 1628 # | 6641 | | | | | | | | | | | | | | | | | | | |
| []_ ([]+[])*.5 LONG | 1629 # | 5576 | | | | | | | | | | | | | | | | | | | |
| []_ ([]+[])*2 LONG | 1630 # | 4582 | | | | | | | | | | | | | | | | | | | |
| []_ ([]+[]+.25)*.5 LO | 1633 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([]-[])*.5 LONG | 1631 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([]-[])*2 LONG | 1632 # | 4579 | | | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. #)*.5 | 1619 # | 4942 | 4949 | 7498 | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. #)*2 | 1620 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. NOT.#)*.5 | 1621 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. NOT.#)*2 | 1622 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. [])*.5 | 1623 # | 4168 | | | | | | | | | | | | | | | | | | | |
| []_ ([] .AND. [])*2 | 1624 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([] .OR. #)*.5 | 1625 # | | | | | | | | | | | | | | | | | | | | |
| []_ ([] .OR. #)*2 | 1626 # | | | | | | | | | | | | | | | | | | | | |
| []_ +SIGN | 1656 # | 2421 | 2456 | 5289 | 5293 | 5318 | 5351 | 5412 | | | | | | | | | | | | | |
| []_ +SIGN*.5 | 1661 # | 5550 | 5652 | 5703 | | | | | | | | | | | | | | | | | |
| []_ -1 | 1529 # | 4320 | 6388 | 7360 | | | | | | | | | | | | | | | | | |
| []_ -2 | 1530 # | | | | | | | | | | | | | | | | | | | | |
| []_ -AC | 1545 # | 4624 | | | | | | | | | | | | | | | | | | | |
| []_ -AC[] | 1546 # | 6267 | | | | | | | | | | | | | | | | | | | |
| []_ -Q | 1531 # | | | | | | | | | | | | | | | | | | | | |
| []_ -Q*.5 | 1533 # | 5376 | | | | | | | | | | | | | | | | | | | |
| []_ -Q*2 | 1532 # | 4448 | | | | | | | | | | | | | | | | | | | |
| []_ -SIGN | 1657 # | 2419 | 2457 | 5290 | 5294 | 5319 | 5352 | 5413 | | | | | | | | | | | | | |
| []_ -SIGN*.5 | 1662 # | 5552 | 5653 | 5705 | | | | | | | | | | | | | | | | | |
| []_ -WORK[] | 1772 # | 5995 | 6938 | 6952 | | | | | | | | | | | | | | | | | |
| []_ -[] | 1534 # | 2576 | 2946 | 4517 | 4535 | 4542 | 4636 | 4640 | 4644 | 4646 | 5279 | 5354 | | | | | | | | | |
| []_ -[] | 5389 # | 5498 | 5514 | 5543 | 5695 | 5775 | 5796 | 6264 | | | | | | | | | | | | | |
| []_ -[]*2 | 1536 # | 5358 | | | | | | | | | | | | | | | | | | | |
| []_ -[]-.25 | 1535 # | 5544 | | | | | | | | | | | | | | | | | | | |
| []_ .NOT.AC[] | 1538 # | 6266 | | | | | | | | | | | | | | | | | | | |
| []_ .NOT.AC | 1537 # | 2895 | 2905 | 4623 | | | | | | | | | | | | | | | | | |
| []_ .NOT.Q | 1539 # | 4908 | 5983 | | | | | | | | | | | | | | | | | | |
| []_ .NOT.WORK[] | 1771 # | 6012 | 6040 | 6044 | 6180 | 6184 | 6956 | | | | | | | | | | | | | | |
| []_ .NOT.[] | 1540 # | 2838 | 2875 | 2916 | 2926 | 3287 | 4090 | 4534 | 4638 | 4645 | 5448 | 5496 | | | | | | | | | |
| []_ O | 5511 # | 5513 | 5694 | 5704 | 5773 | 5964 | 5966 | 6518 | 7647 | | | | | | | | | | | | |
| []_ O | 1541 # | 2160 | 2162 | 2165 | 2174 | 2193 | 2215 | 2399 | 2803 | 3270 | 4222 | 4318 | | | | | | | | | |
| []_ O | 4532 | 4812 | 4954 | 5097 | 5139 | 5163 | 5249 | 5515 | 5688 | 5755 | 5771 | 5799 | | | | | | | | | |
| []_ O | 6016 | 6088 | 6100 | 6111 | 6140 | 6261 | 6299 | 6333 | 6389 | 6393 | 6397 | 6400 | | | | | | | | | |
| []_ O | 6455 | 6727 | 7127 | 7137 | 7181 | 7257 | 7332 | 7377 | 7445 | 7674 | 7719 | 7843 | | | | | | | | | |
| []_ O | 8507 | 8764 | 8782 | 8783 | 8814 | | | | | | | | | | | | | | | | |
| []_ O XWD [] | 1543 # | 2154 | 2157 | 2201 | 2237 | 2241 | 2244 | 2256 | 2259 | 2262 | 3942 | 3982 | | | | | | | | | |
| []_ O XWD [] | 4030 | 4394 | 4405 | 5879 | 5881 | 5883 | 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | | | | | | | | | |
| []_ O XWD [] | 6591 | 7248 | 7427 | 7428 | 7429 | 7430 | 7431 | 7432 | 7433 | | | | | | | | | | | | |
| []_ O*.5 LONG | 1542 # | 4248 | | | | | | | | | | | | | | | | | | | |
| []_ AC[] | 1560 # | 2180 | 2370 | 2780 | 5954 | 5972 | 5975 | 6005 | 6021 | 6093 | 6113 | 6145 | | | | | | | | | |
| []_ AC[] | 6160 | 6188 | 6191 | 6193 | 6210 | 6251 | 6253 | 6254 | 6286 | 6292 | 6319 | 6331 | | | | | | | | | |

| | | | | | | | | | | | | |
|--------------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 6366 | 6509 | 6550 | 6587 | 6589 | 6693 | 6806 | 6813 | 6839 | 6841 | 6868 | 6925 |
| | 6933 | 6939 | | | | | | | | | | |
| []_AC[]*.5 | 1562 # | 4188 | | | | | | | | | | |
| []_AC[]*2 | 1561 # | 6200 | | | | | | | | | | |
| []_AC[]-1 | 1557 # | 6360 | 6379 | | | | | | | | | |
| []_AC[]-[] | 1556 # | 4095 | | | | | | | | | | |
| []_AC[]_AND.[] | 1558 # | | | | | | | | | | | |
| []_AC | 1544 # | 2356 | 2369 | 2587 | 2688 | 2690 | 2721 | 2727 | 2964 | 2981 | 3472 | 3716 |
| | 3749 | 3771 | 3827 | 3839 | 4146 | 4325 | 4716 | 5112 | 5200 | 5282 | 5315 | 5347 |
| | 5411 | 5557 | 5957 | 6095 | 6157 | 6291 | 6293 | 6314 | 6451 | 7546 | 7833 | |
| []_AC*.5 | 1547 # | 2987 | 2999 | 3002 | 4193 | 4408 | 5547 | 5649 | 5700 | | | |
| []_AC*.5 LONG | 1548 # | 2383 | 3089 | | | | | | | | | |
| []_AC*2 | 1549 # | 4712 | | | | | | | | | | |
| []_AC+1 | 1550 # | 3486 | | | | | | | | | | |
| []_AC+1000001 | 1551 # | 3508 | 3678 | | | | | | | | | |
| []_AC+[] | 1552 # | 5103 | | | | | | | | | | |
| []_AC-1 | 1553 # | 3500 | | | | | | | | | | |
| []_AC-[] | 1554 # | 3369 | 4070 | 4100 | | | | | | | | |
| []_AC-[]-.25 | 1555 # | 4097 | | | | | | | | | | |
| []_AC.AND.MASK | 1559 # | 2967 | 3015 | | | | | | | | | |
| []_APR | 1563 # | 7002 | 7003 | 7060 | 7092 | 7266 | | | | | | |
| []_CURRENT AC [] | 1564 # | | | | | | | | | | | |
| []_EA | 1567 # | 2319 | | | | | | | | | | |
| []_EA FROM [] | 1565 # | 5925 | | | | | | | | | | |
| []_EXP | 1568 # | 5489 | 5505 | 5835 | | | | | | | | |
| []_FE | 1569 # | 3040 | 6654 | | | | | | | | | |
| []_FLAGS | 1570 # | 3627 | 3641 | 3985 | 7458 | | | | | | | |
| []_IO DATA | 1821 # | 7443 | 7491 | 7587 | 8220 | | | | | | | |
| []_MEM | 1822 # | 2268 | 2338 | 2352 | 2361 | 2365 | 2393 | 2410 | 3548 | 3572 | 3587 | 3612 |
| | 3616 | 3677 | 3722 | 3758 | 3844 | 4001 | 4764 | 5911 | 5930 | 6350 | 6593 | 6621 |
| | 6897 | 6912 | 6982 | 7104 | 7221 | 7225 | 6998 | 7233 | 7237 | 7277 | 7316 | 7454 |
| | 7496 | 7667 | 7712 | 7715 | 7826 | 7865 | 8679 | | | | | |
| []_MEM THEN FETCH | 1823 # | 2404 | | | | | | | | | | |
| []_MEM*.5 | 1824 # | | | | | | | | | | | |
| []_MEM.AND.MASK | 1825 # | 4770 | | | | | | | | | | |
| []_P | 1571 # | 6524 | 6670 | | | | | | | | | |
| []_PC WITH FLAGS | 1572 # | 3688 | 3805 | 3811 | 4014 | 4215 | 6963 | 7470 | 8808 | | | |
| []_Q | 1573 # | 4123 | 4131 | 4351 | 4406 | 4423 | 4429 | 4438 | 5606 | 5613 | 5633 | 5656 |
| | 5715 | 5717 | | | | | | | | | | |
| []_Q*.5 | 1574 # | 4186 | 5370 | 5375 | 5645 | | | | | | | |
| []_Q*2 | 1575 # | | | | | | | | | | | |
| []_Q*2 LONG | 1576 # | 4434 | 4444 | | | | | | | | | |
| []_Q+1 | 1577 # | 5100 | | | | | | | | | | |
| []_RAM | 1578 # | | | | | | | | | | | |
| []_TIME | 1579 # | 7291 | 7292 | 7293 | | | | | | | | |
| []_VMA | 1580 # | 7681 | 7924 | 8194 | 8209 | 8725 | | | | | | |
| []_VMA FLAGS | 1842 # | 7439 | 7486 | 7578 | 7581 | 7617 | 7628 | 7634 | 7860 | | | |
| []_VMA IO READ | 1843 # | 7439 | 7486 | 7578 | 7581 | 7860 | | | | | | |
| []_VMA IO WRITE | 1844 # | 7617 | 7628 | 7634 | | | | | | | | |
| []_WORK[] | 1770 # | 4966 | 4976 | 5175 | 6059 | 6060 | 6061 | 6062 | 6126 | 6337 | 6347 | 6380 |
| | 6381 | 6402 | 6578 | 6614 | 6691 | 6811 | 6822 | 6929 | 7049 | 7083 | 7151 | 7208 |
| | 7210 | 7212 | 7214 | 7216 | 7245 | 7254 | 7259 | 7265 | 7297 | 7301 | 7322 | 7654 |
| | 7659 | 7912 | 7917 | 7921 | 7936 | 7954 | 7956 | 7957 | 7959 | 8198 | 8498 | 8538 |
| | 8539 | 8540 | 8546 | 8735 | 8741 | 8750 | 8754 | 8766 | | | | |
| []_WORK[]+1 | 1773 # | 5991 | 6024 | 6079 | 6177 | 6355 | 6818 | 6824 | | | | |

Cross Reference Listing

| | | | | | | | | | | | | | |
|----------------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| []_XR | 1581 # | 7657 | | | | | | | | | | | |
| []-[] | 1582 # | 2294 | 3018 | 3453 | 3456 | 3524 | 3525 | 3551 | 3578 | 3593 | 3652 | 3706 | |
| | 3824 | 3829 | 4117 | 4145 | 4169 | 4185 | 4315 | 4324 | 4339 | 4345 | 4374 | 4729 | |
| | 5037 | 5285 | 5286 | 5287 | 5325 | 5345 | 5353 | 5562 | 5563 | 5570 | 5571 | 5644 | |
| | 5686 | 5689 | 5731 | 5732 | 5973 | 5993 | 6039 | 6074 | 6334 | 6345 | 6353 | 6420 | |
| | 6520 | 6753 | 6844 | 6867 | 6886 | 6943 | 7009 | 7010 | 7169 | 7186 | 7193 | 7198 | |
| | 7296 | 7328 | 7474 | 7556 | 7558 | 7565 | 7590 | 8260 | 8495 | 8519 | 8816 | | |
| []-[] SWAP | 1583 # | 2388 | 2568 | 2715 | 2717 | 2720 | 2723 | 2726 | 2729 | 3268 | 3273 | 3592 | |
| | 3769 | 3826 | 3840 | 4810 | 4936 | 4952 | 5052 | 5092 | 5101 | 5174 | 6399 | 6456 | |
| | 6461 | 6462 | 6756 | 7084 | 7336 | 7349 | 7359 | 8688 | | | | | |
| []-[] XWD O | 1584 # | 6154 | 6269 | 7707 | 8224 | 8226 | 8240 | | | | | | |
| []-[]*.5 | 1585 # | 2151 | 2425 | 2426 | 2983 | 3004 | 3005 | 3009 | 3092 | 3095 | 4167 | 4224 | |
| | 4386 | 4414 | 4545 | 4546 | 4547 | 4553 | 4814 | 4890 | 4939 | 4958 | 5360 | 5368 | |
| | 5435 | 5484 | 5486 | 5502 | 5503 | 5522 | 5523 | 5528 | 5638 | 5671 | 5840 | 6463 | |
| | 6466 | 6470 | 6472 | 6552 | 6660 | 6667 | 6730 | 7177 | 7179 | 7187 | 7194 | 7196 | |
| | 7501 | 7519 | 7594 | 7600 | 8263 | 8279 | | | | | | | |
| []-[]*.5 LONG | 1586 # | 2384 | 3070 | 3098 | 3142 | 4119 | 4147 | 4226 | 4247 | 4260 | 4388 | 4390 | |
| | 4402 | 4418 | 4433 | 4440 | 4492 | 4540 | 5040 | 5207 | 5227 | 5228 | 5229 | 5243 | |
| | 5298 | 5321 | 5326 | 5401 | 5403 | 5433 | 5601 | 5604 | 5612 | 5615 | 5617 | 5631 | |
| | 5666 | 5721 | 5747 | 5750 | 5784 | 5786 | 5832 | | | | | | |
| []-[]*2 | 1587 # | 2149 | 2971 | 2994 | 3007 | 3010 | 3037 | 4122 | 4149 | 4205 | 5058 | 5242 | |
| | 5355 | 5363 | 5438 | 6201 | 6747 | 7131 | 7146 | 7148 | 7446 | 7482 | 7626 | 7632 | |
| | 7723 | 8501 | | | | | | | | | | | |
| []-[]*2 LONG | 1588 # | 2990 | 3071 | 3074 | 3075 | 3076 | 3101 | 3105 | 3108 | 3143 | 3146 | 3147 | |
| | 4472 | 4552 | 4555 | 4557 | 4903 | 5222 | 5223 | 5480 | 5492 | 5500 | 5504 | 5578 | |
| | 5583 | 5585 | 5597 | 5599 | 5600 | 5629 | 5670 | 5674 | 5723 | 5740 | 5762 | 5779 | |
| | 5807 | | | | | | | | | | | | |
| []-[]*4 | 1589 # | 6457 | | | | | | | | | | | |
| []-[]+# | 1590 # | 3724 | 3753 | 6416 | 6425 | 7447 | 7492 | 8665 | 8675 | 8789 | | | |
| []-[]+.25 | 1591 # | 5836 | | | | | | | | | | | |
| []-[]+0 | 1592 # | | | | | | | | | | | | |
| []-[]+1 | 1593 # | 2179 | 2183 | 2413 | 2416 | 3403 | 3613 | 3634 | 4735 | 5141 | 5165 | 5250 | |
| | 5447 | 5950 | 6051 | 6069 | 6098 | 6241 | 6309 | 6321 | 6328 | 6348 | 6369 | 6383 | |
| | 6438 | 6501 | 6525 | 6619 | 6846 | 6888 | 7255 | 7721 | 8768 | 8829 | | | |
| []-[]+1000001 | 1594 # | 3817 | 3832 | 5115 | | | | | | | | | |
| []-[]+AC | 1595 # | 4056 | 4086 | | | | | | | | | | |
| []-[]+AC[] | 1596 # | 4083 | 6202 | 6212 | 6215 | | | | | | | | |
| []-[]+Q | 1597 # | 4446 | | | | | | | | | | | |
| []-[]+RAM | 1598 # | 6432 | 6434 | | | | | | | | | | |
| []-[]+WORK[] | 1766 # | 5034 | 6373 | 6553 | 6629 | 6736 | 6744 | 7298 | 7671 | 7673 | | | |
| []-[]+XR | 1599 # | 2288 | 2300 | 2315 | 2328 | 3563 | 5924 | 5927 | 7665 | | | | |
| []-[]+[] | 1600 # | 3944 | 3984 | 4199 | 4465 | 4468 | 4556 | 5299 | 5443 | 5526 | 5657 | 6014 | |
| | 6046 | 6218 | 6229 | 6231 | 6668 | 6950 | 6966 | 6974 | 7249 | 7449 | 7484 | 7494 | |
| | 7502 | 8655 | 8670 | | | | | | | | | | |
| []-[]+[]+.25 | 1601 # | | | | | | | | | | | | |
| []-[]-# | 1602 # | | | | | | | | | | | | |
| []-[]-1 | 1603 # | 3419 | 6030 | 6134 | 6135 | 6246 | 6387 | 6424 | 6817 | 8758 | 8818 | | |
| []-[]-1000001 | 1604 # | | | | | | | | | | | | |
| []-[]-AC | 1605 # | | | | | | | | | | | | |
| []-[]-RAM | 1606 # | 6422 | 6427 | | | | | | | | | | |
| []-[]-WORK[] | 1775 # | | | | | | | | | | | | |
| []-[]-[] | 1607 # | 6009 | 6335 | 6946 | 7260 | 7384 | 7397 | | | | | | |
| []-[]-[] REV | 1608 # | 6133 | | | | | | | | | | | |
| []-[] .AND .AC | 1634 # | 2813 | 2936 | 3275 | 3298 | 7532 | | | | | | | |
| []-[] .AND .NOT .# | 1635 # | 3620 | 4021 | 5820 | 6404 | 6772 | 6788 | 7050 | 7076 | 7344 | 8520 | 8695 | |

Cross Reference Listing

| | | | | | | | | | | | | |
|----------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 2634 | 2635 | 2641 | 2642 | 2646 | 2647 | 2651 | 2652 | 2656 | 2657 | 2661 | 2662 |
| | 2666 | 2667 | 2671 | 2672 | 2676 | 2677 | 2766 | 2796 | 2797 | 2806 | 2807 | 2816 |
| | 2817 | 2826 | 2827 | 2831 | 2832 | 2848 | 2849 | 2858 | 2859 | 2868 | 2869 | 2878 |
| | 2879 | 2888 | 2889 | 2898 | 2899 | 2909 | 2910 | 2919 | 2920 | 2929 | 2930 | 2939 |
| | 2940 | 4049 | 4050 | 4063 | 4064 | 4111 | 4112 | 4681 | 8187 | | | |
| AC DISP | 2137 # | 3515 | 6995 | 6996 | 7203 | | | | | | | |
| B | 2125 # | 2799 | 2809 | 2819 | 2829 | 2834 | 2851 | 2861 | 2871 | 2881 | 2891 | 2901 |
| | 2912 | 2922 | 2932 | 2942 | 4052 | 4066 | 4114 | | | | | |
| DAC | 2127 # | 2765 | 4078 | 4079 | 4136 | 4137 | 4162 | 4303 | 4304 | 4308 | 4309 | 4380 |
| | 5622 | 5683 | | | | | | | | | | |
| DBL AC | 2112 # | 2774 | | | | | | | | | | |
| DBL B | 2128 # | 4139 | 4306 | 4311 | | | | | | | | |
| DBL FL-R | 2118 # | 5533 | 5534 | 5622 | 5683 | | | | | | | |
| DBL R | 2111 # | 2765 | 2766 | 4078 | 4079 | 4162 | 4380 | | | | | |
| FL-AC | 2129 # | 5259 | 5262 | 5263 | 5267 | 5270 | 5271 | 5304 | 5308 | 5309 | 5333 | 5337 |
| | 5338 | 5383 | 5384 | 5418 | 5419 | | | | | | | |
| FL-BOTH | 2131 # | 5261 | 5265 | 5269 | 5273 | 5306 | 5311 | 5335 | 5340 | | | |
| FL-I | 2117 # | 5263 | 5271 | 5309 | 5338 | | | | | | | |
| FL-MEM | 2130 # | 5260 | 5264 | 5268 | 5272 | 5305 | 5310 | 5334 | 5339 | | | |
| FL-R | 2115 # | 5259 | 5262 | 5267 | 5270 | 5304 | 5308 | 5333 | 5337 | 5418 | 5419 | |
| FL-RW | 2116 # | 5260 | 5261 | 5264 | 5265 | 5268 | 5269 | 5272 | 5273 | 5305 | 5306 | 5310 |
| | 5311 | 5334 | 5335 | 5339 | 5340 | | | | | | | |
| I | 2103 # | 2842 | 2954 | 3174 | 3175 | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 | 3183 |
| | 3184 | 3192 | 3193 | 3194 | 3195 | 3196 | 3197 | 3198 | 3199 | 3210 | 3211 | 3212 |
| | 3213 | 3214 | 3215 | 3216 | 3217 | 3227 | 3228 | 3229 | 3230 | 3231 | 3232 | 3233 |
| | 3234 | 3349 | 3350 | 3351 | 3352 | 3353 | 3354 | 3355 | 3356 | 3461 | 3462 | 3463 |
| | 3464 | 3465 | 3466 | 3467 | 3468 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 |
| | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 | 3503 | 3504 | 3515 | 3517 | 3668 |
| | 3670 | 3671 | 3798 | 3799 | 3800 | 3801 | 3851 | 3852 | 3853 | 3854 | 3855 | 3856 |
| | 3857 | 3858 | 3862 | 3863 | 3864 | 3865 | 3866 | 3867 | 3868 | 3869 | 3870 | 3871 |
| | 3872 | 3873 | 3874 | 3875 | 3876 | 3877 | 3878 | 3879 | 3880 | 3881 | 3882 | 3883 |
| | 3884 | 3885 | 3886 | 3887 | 3888 | 3889 | 3890 | 3891 | 3892 | 3893 | 3897 | 3898 |
| | 3899 | 3900 | 3904 | 3905 | 3906 | 3907 | 3908 | 3909 | 3910 | 3911 | 3912 | 3913 |
| | 3914 | 5108 | 5384 | 5848 | 5849 | 5850 | 5851 | 5852 | 5853 | 5854 | 5856 | 5857 |
| | 5858 | 5859 | 5861 | 5862 | 5863 | 5864 | 5866 | 5867 | 5868 | 5869 | 5870 | 5871 |
| | 5872 | 5873 | 5907 | 7736 | 7737 | 7738 | 7740 | 7741 | 7743 | 7744 | 7746 | 7747 |
| | 7748 | 7749 | 7750 | 7751 | 7752 | 7753 | 7755 | 7756 | 7757 | 7758 | 7759 | 7760 |
| | 7761 | 7762 | 7764 | 7765 | 7766 | 7767 | 7768 | 7769 | 7770 | 7771 | 7773 | 7774 |
| | 7775 | 7776 | 7777 | 7778 | 7779 | 7780 | 7782 | 7783 | 7784 | 7785 | 7786 | 7787 |
| | 7788 | 7789 | | | | | | | | | | |
| I-PF | 2104 # | 2547 | 2552 | 2557 | 2562 | 2600 | 2605 | 2610 | 2615 | 2620 | 2625 | 2630 |
| | 2635 | 2642 | 2647 | 2652 | 2657 | 2662 | 2667 | 2672 | 2677 | 2796 | 2797 | 2807 |
| | 2817 | 2827 | 2832 | 2849 | 2859 | 2869 | 2879 | 2888 | 2889 | 2899 | 2910 | 2920 |
| | 2930 | 2939 | 2940 | 2955 | 3475 | 3489 | 3765 | 4050 | 4064 | 4112 | 4137 | 4304 |
| | 4309 | | | | | | | | | | | |
| IOT | 2119 # | 6995 | 6996 | 7203 | 7524 | 7525 | 7526 | 7527 | 7535 | 7536 | 7537 | 7538 |
| | 7549 | 7550 | 7551 | 7552 | 7817 | 7818 | 8187 | | | | | |
| IR | 2110 # | 3669 | | | | | | | | | | |
| IW | 2109 # | 2798 | 2799 | 2890 | 2891 | 2941 | 2942 | | | | | |
| M | 2124 # | 2548 | 2553 | 2558 | 2563 | 2601 | 2606 | 2611 | 2616 | 2621 | 2626 | 2631 |
| | 2636 | 2643 | 2648 | 2653 | 2658 | 2663 | 2668 | 2673 | 2678 | 2798 | 2808 | 2818 |
| | 2828 | 2833 | 2843 | 2844 | 2850 | 2860 | 2870 | 2880 | 2890 | 2900 | 2911 | 2921 |
| | 2931 | 2941 | 4051 | 4065 | 4113 | 4138 | 4305 | 4310 | 7203 | | | |
| R | 2105 # | 2583 | 2841 | 3185 | 3186 | 3187 | 3188 | 3189 | 3190 | 3201 | 3202 | 3203 |
| | 3204 | 3205 | 3206 | 3207 | 3208 | 3219 | 3220 | 3221 | 3222 | 3223 | 3224 | 3225 |

| | | | | | | | | | | | | |
|----------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 3226 | 3236 | 3237 | 3238 | 3239 | 3240 | 3241 | 3242 | 3243 | 3358 | 3359 | 3360 |
| | 3361 | 3362 | 3363 | 3364 | 3365 | 3375 | 3376 | 3377 | 3378 | 3379 | 3380 | 3381 |
| | 3382 | 3646 | 4681 | 4682 | 4683 | 4684 | 4685 | 5383 | | | | |
| R-PF | 2106 # | 2546 | 2551 | 2556 | 2561 | 2599 | 2604 | 2609 | 2614 | 2619 | 2624 | 2629 |
| | 2634 | 2641 | 2646 | 2651 | 2656 | 2661 | 2666 | 2671 | 2676 | 2806 | 2816 | 2826 |
| | 2831 | 2848 | 2858 | 2868 | 2878 | 2898 | 2909 | 2919 | 2929 | 4049 | 4063 | 4111 |
| | 4136 | 4303 | 4308 | | | | | | | | | |
| ROUND | 2132 # | 5262 | 5263 | 5264 | 5265 | 5270 | 5271 | 5272 | 5273 | 5308 | 5309 | 5310 |
| | 5311 | 5337 | 5338 | 5339 | 5340 | 5383 | 5419 | | | | | |
| RW | 2108 # | 2549 | 2554 | 2559 | 2564 | 2601 | 2602 | 2606 | 2607 | 2612 | 2617 | 2622 |
| | 2627 | 2632 | 2637 | 2643 | 2644 | 2648 | 2649 | 2654 | 2659 | 2664 | 2669 | 2674 |
| | 2679 | 2808 | 2809 | 2818 | 2819 | 2828 | 2829 | 2833 | 2834 | 2850 | 2851 | 2860 |
| | 2861 | 2870 | 2871 | 2880 | 2881 | 2900 | 2901 | 2911 | 2912 | 2921 | 2922 | 2931 |
| | 2932 | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 | 3408 | 3409 | 3410 |
| | 3411 | 3412 | 3413 | 3414 | 3415 | 4051 | 4052 | 4065 | 4066 | 4113 | 4114 | 4138 |
| | 4139 | 4305 | 4306 | 4310 | 4311 | | | | | | | |
| S | 2126 # | 2549 | 2554 | 2559 | 2564 | 2602 | 2607 | 2612 | 2617 | 2622 | 2627 | 2632 |
| | 2637 | 2644 | 2649 | 2654 | 2659 | 2664 | 2669 | 2674 | 2679 | | | |
| SH | 2113 # | 2951 | 2952 | 2953 | | | | | | | | |
| SHC | 2114 # | 2956 | 2957 | | | | | | | | | |
| SJC- | 3309 # | 3349 | 3358 | 3375 | 3392 | 3408 | 3461 | 3475 | 3489 | | | |
| SJCA | 3313 # | 3353 | 3362 | 3379 | 3396 | 3412 | 3465 | 3479 | 3493 | | | |
| SJCE | 3311 # | 3351 | 3360 | 3377 | 3394 | 3410 | 3463 | 3477 | 3491 | 5849 | | |
| SJCG | 3316 # | 3356 | 3365 | 3382 | 3399 | 3415 | 3468 | 3482 | 3496 | 5854 | | |
| SJCGE | 3314 # | 3354 | 3363 | 3380 | 3397 | 3413 | 3466 | 3480 | 3494 | 3503 | 5852 | |
| SJCL | 3310 # | 3350 | 3359 | 3376 | 3393 | 3409 | 3462 | 3476 | 3490 | 3504 | 5848 | |
| SJCLE | 3312 # | 3352 | 3361 | 3378 | 3395 | 3411 | 3464 | 3478 | 3492 | 5850 | | |
| SJCN | 3315 # | 3355 | 3364 | 3381 | 3398 | 3414 | 3467 | 3481 | 3495 | 5853 | | |
| TC- | 3165 # | 3210 | 3211 | 3219 | 3220 | | | | | | | |
| TCA | 3167 # | 3214 | 3215 | 3223 | 3224 | | | | | | | |
| TCE | 3166 # | 3212 | 3213 | 3221 | 3222 | | | | | | | |
| TCN | 3168 # | 3216 | 3217 | 3225 | 3226 | | | | | | | |
| TN- | 3155 # | | | | | | | | | | | |
| TNA | 3158 # | 3178 | 3179 | 3187 | 3188 | | | | | | | |
| TNE | 3156 # | 3176 | 3177 | 3185 | 3186 | 7526 | | | | | | |
| TNN | 3159 # | 3180 | 3181 | 3189 | 3190 | 7527 | | | | | | |
| TO- | 3169 # | 3227 | 3228 | 3236 | 3237 | | | | | | | |
| TOA | 3171 # | 3231 | 3232 | 3240 | 3241 | | | | | | | |
| TOE | 3170 # | 3229 | 3230 | 3238 | 3239 | | | | | | | |
| TON | 3172 # | 3233 | 3234 | 3242 | 3243 | | | | | | | |
| TZ- | 3161 # | 3192 | 3193 | 3201 | 3202 | | | | | | | |
| TZA | 3163 # | 3196 | 3197 | 3205 | 3206 | | | | | | | |
| TZE | 3162 # | 3194 | 3195 | 3203 | 3204 | | | | | | | |
| TZN | 3164 # | 3198 | 3199 | 3207 | 3208 | | | | | | | |
| W | 2107 # | 2548 | 2553 | 2558 | 2563 | 2611 | 2616 | 2621 | 2626 | 2631 | 2636 | 2653 |
| | 2658 | 2663 | 2668 | 2673 | 2678 | 2775 | 2843 | 2844 | | | | |
| W TEST | 2136 # | 2583 | 4682 | 4684 | | | | | | | | |
| WORD-TNE | 3157 # | 7524 | | | | | | | | | | |
| WORD-TNN | 3160 # | 7525 | | | | | | | | | | |
| (U) MEM | 959 # | 2160 | 2179 | 2183 | 2227 | 2227 | 2227 | 2228 | 2228 | 2230 | 2230 | 2231 |
| | 2231 | 2231 | 2249 | 2250 | 2267 | 2275 | 2290 | 2290 | 2296 | 2296 | 2301 | 2302 |
| | 2306 | 2307 | 2320 | 2329 | 2330 | 2333 | 2334 | 2338 | 2342 | 2351 | 2360 | 2362 |
| | 2363 | 2364 | 2392 | 2403 | 2404 | 2404 | 2414 | 2415 | 2417 | 2418 | 2422 | 2429 |
| | 2429 | 2493 | 2525 | 2531 | 2536 | 2539 | 2568 | 2574 | 2588 | 2589 | 2688 | 2690 |
| | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2783 | 2784 | 2786 | 2787 | 2788 | 2790 |

| | | | | | | | | | | | |
|-------------------------------|--------|------|------|------|------|------|------|------|------|------|------|
| 2803 | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 3019 | 3019 |
| 3325 | 3325 | 3325 | 3337 | 3337 | 3404 | 3405 | 3452 | 3452 | 3452 | 3453 | 3453 |
| 3456 | 3456 | 3457 | 3457 | 3457 | 3524 | 3524 | 3525 | 3525 | 3526 | 3526 | 3531 |
| 3531 | 3535 | 3535 | 3538 | 3538 | 3547 | 3551 | 3551 | 3564 | 3565 | 3567 | 3568 |
| 3571 | 3586 | 3593 | 3593 | 3608 | 3612 | 3614 | 3615 | 3616 | 3629 | 3629 | 3631 |
| 3631 | 3632 | 3635 | 3640 | 3640 | 3676 | 3680 | 3681 | 3693 | 3699 | 3707 | 3708 |
| 3717 | 3718 | 3721 | 3729 | 3731 | 3734 | 3736 | 3738 | 3744 | 3750 | 3752 | 3758 |
| 3813 | 3814 | 3815 | 3825 | 3841 | 3842 | 3843 | 3953 | 3954 | 3954 | 3954 | 3973 |
| 3978 | 3978 | 3978 | 3997 | 3998 | 3999 | 4000 | 4012 | 4016 | 4025 | 4025 | 4025 |
| 4034 | 4035 | 4039 | 4041 | 4042 | 4043 | 4057 | 4071 | 4123 | 4129 | 4152 | 4154 |
| 4217 | 4217 | 4217 | 4692 | 4692 | 4692 | 4701 | 4701 | 4729 | 4735 | 4736 | 4748 |
| 4749 | 4752 | 4753 | 4756 | 4757 | 4760 | 4761 | 4763 | 4769 | 4890 | 4891 | 5093 |
| 5094 | 5121 | 5122 | 5129 | 5130 | 5132 | 5143 | 5144 | 5146 | 5148 | 5149 | 5151 |
| 5155 | 5158 | 5166 | 5167 | 5169 | 5204 | 5245 | 5245 | 5247 | 5251 | 5251 | 5252 |
| 5443 | 5447 | 5493 | 5515 | 5911 | 5925 | 5927 | 5927 | 5928 | 5928 | 5930 | 5951 |
| 5952 | 6099 | 6099 | 6348 | 6349 | 6350 | 6375 | 6376 | 6420 | 6426 | 6433 | 6449 |
| 6449 | 6454 | 6454 | 6510 | 6510 | 6542 | 6544 | 6588 | 6588 | 6590 | 6593 | 6621 |
| 6630 | 6630 | 6745 | 6746 | 6892 | 6892 | 6893 | 6893 | 6894 | 6894 | 6895 | 6895 |
| 6897 | 6911 | 6967 | 6970 | 6971 | 6975 | 6978 | 6979 | 6982 | 7101 | 7102 | 7103 |
| 7220 | 7221 | 7224 | 7225 | 7228 | 7229 | 7232 | 7233 | 7236 | 7237 | 7275 | 7276 |
| 7278 | 7279 | 7303 | 7304 | 7307 | 7308 | 7308 | 7308 | 7309 | 7315 | 7316 | 7351 |
| 7351 | 7351 | 7369 | 7369 | 7370 | 7375 | 7438 | 7442 | 7453 | 7453 | 7453 | 7453 |
| 7454 | 7454 | 7469 | 7490 | 7495 | 7495 | 7495 | 7496 | 7503 | 7504 | 7584 | 7584 |
| 7586 | 7619 | 7619 | 7620 | 7622 | 7622 | 7622 | 7644 | 7645 | 7666 | 7666 | 7667 |
| 7712 | 7715 | 7717 | 7725 | 7725 | 7727 | 7729 | 7822 | 7823 | 7825 | 7830 | 7831 |
| 7840 | 7843 | 7843 | 7843 | 7845 | 7846 | 7846 | 7846 | 7857 | 7858 | 7864 | 7913 |
| 7929 | 7929 | 7929 | 7930 | 7931 | 7932 | 7932 | 7932 | 7933 | 7934 | 7937 | 7938 |
| 7939 | 7939 | 7939 | 7940 | 7941 | 7942 | 7942 | 7942 | 7943 | 7944 | 7945 | 7945 |
| 7945 | 7946 | 7947 | 7948 | 7948 | 7948 | 7949 | 7950 | 7951 | 7951 | 7951 | 7952 |
| 7953 | 7955 | 7958 | 8244 | 8517 | 8525 | 8541 | 8656 | 8657 | 8658 | 8671 | 8672 |
| 8673 | 8678 | 8739 | 8745 | 8793 | 8794 | 8794 | 8794 | 8795 | 8809 | 8817 | 8830 |
| 8830 | 8830 | 8833 | 8833 | 8833 | 8833 | | | | | | |
| (U) MICROCODE OPTION(INH^V@^H | 1239 # | | | | | | | | | | |
| OPT | 1243 # | 7038 | | | | | | | | | |
| (U) MICROCODE OPTION(KIP^W@^H | 1263 # | | | | | | | | | | |
| OPT | 1265 # | 7042 | | | | | | | | | |
| (U) MICROCODE OPTION(KLP^W#^H | 1269 # | | | | | | | | | | |
| OPT | 1273 # | 7043 | | | | | | | | | |
| (U) MICROCODE OPTION(NOC^V#^H | 1245 # | | | | | | | | | | |
| OPT | 1249 # | 7039 | | | | | | | | | |
| (U) MICROCODE OPTION(NON^W | 1251 # | | | | | | | | | | |
| OPT | 1253 # | 7040 | | | | | | | | | |
| (U) MICROCODE OPTION(UBA^W ^H | 1257 # | | | | | | | | | | |
| OPT | 1259 # | 7041 | | | | | | | | | |
| (U) MICROCODE OPTIONS | 1232 # | | | | | | | | | | |
| (U) MICROCODE VERSION | 1276 # | | | | | | | | | | |
| UCV | 1277 # | 7044 | | | | | | | | | |
| (D) MODE | 1337 # | | | | | | | | | | |
| (U) MULTI PREC | 964 # | 4399 | 4451 | 4466 | 4469 | 4476 | 4590 | 5544 | 5577 | 5697 | 5708 |
| (U) MULTI SHIFT | 966 # | 2972 | 2984 | 3006 | 3011 | 4816 | 4821 | 4897 | 4905 | 5242 | 5243 |
| (U) PHYSICAL | 1133 # | 3954 | 3978 | 3998 | 4025 | 6971 | 6979 | 7381 | 7382 | 7385 | 7394 |
| | 7398 | 7439 | 7453 | 7486 | 7495 | 7578 | 7581 | 7617 | 7628 | 7634 | 7843 |
| | 7860 | 7929 | 7932 | 7939 | 7942 | 7945 | 7948 | 7951 | 8657 | 8673 | 8794 |
| | 8833 | | | | | | | | | | 8830 |
| (U) PI.CLR | 1192 # | 7331 | | | | | | | | | |

| | | | | | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| (U) PI.CO1 | 1181 # | | | | | | | | | | | | |
| (U) PI.CO2 | 1182 # | | | | | | | | | | | | |
| (U) PI.DIR | 1191 # | 7337 | | | | | | | | | | | |
| (U) PI.IP1 | 1173 # | 7514 | | | | | | | | | | | |
| (U) PI.IP2 | 1174 # | | | | | | | | | | | | |
| (U) PI.IP3 | 1175 # | | | | | | | | | | | | |
| (U) PI.IP4 | 1176 # | | | | | | | | | | | | |
| (U) PI.IP5 | 1177 # | | | | | | | | | | | | |
| (U) PI.IP6 | 1178 # | | | | | | | | | | | | |
| (U) PI.IP7 | 1179 # | | | | | | | | | | | | |
| (U) PI.MBZ | 1190 # | 7333 | | | | | | | | | | | |
| (U) PI.ON | 1180 # | 7342 | 7344 | | | | | | | | | | |
| (U) PI.REQ | 1193 # | 7339 | | | | | | | | | | | |
| (U) PI.SC1 | 1198 # | | | | | | | | | | | | |
| (U) PI.SC2 | 1199 # | | | | | | | | | | | | |
| (U) PI.SC3 | 1200 # | | | | | | | | | | | | |
| (U) PI.SC4 | 1201 # | | | | | | | | | | | | |
| (U) PI.SC5 | 1202 # | | | | | | | | | | | | |
| (U) PI.SC6 | 1203 # | | | | | | | | | | | | |
| (U) PI.SC7 | 1204 # | | | | | | | | | | | | |
| (U) PI.TCF | 1195 # | 7347 | | | | | | | | | | | |
| (U) PI.TCN | 1194 # | 7345 | | | | | | | | | | | |
| (U) PI.TSF | 1196 # | 7343 | | | | | | | | | | | |
| (U) PI.TSN | 1197 # | 7341 | | | | | | | | | | | |
| (U) PI.ZER | 1172 # | | | | | | | | | | | | |
| (U) PXCT | 1137 # | | | | | | | | | | | | |
| BIS-DST-EA | 1143 # | 6886 | 6888 | 6894 | 6895 | 6897 | | | | | | | |
| BIS-SRC-EA | 1141 # | | | | | | | | | | | | |
| CURRENT | 1138 # | 2227 | 2228 | 2230 | 2231 | 2250 | 2290 | 2296 | 2404 | 2429 | 3019 | 3325 | |
| | 3337 | 3452 | 3453 | 3456 | 3457 | 3524 | 3525 | 3549 | 3551 | 3573 | 3588 | 3593 | |
| | 3708 | 4217 | 4692 | 4701 | 7351 | 7622 | 7663 | 7858 | | | | | |
| D1 | 1140 # | 2320 | 2362 | 2414 | 2417 | 2785 | 2789 | 3730 | 3735 | 5123 | 5131 | 5145 | |
| | 5150 | 5246 | | | | | | | | | | | |
| D2 | 1144 # | 3682 | 3719 | 3751 | 4750 | 4754 | 5095 | 5168 | 5252 | 6892 | 6893 | | |
| E1 | 1139 # | 2329 | 2334 | 2340 | 3569 | 3655 | | | | | | | |
| E2 | 1142 # | 4699 | 4718 | 4758 | 4762 | 4766 | 5918 | 5927 | 5928 | 5930 | 6849 | | |
| (U) RAMADR | 676 # | | | | | | | | | | | | |
| AC# | 677 # | 2356 | 2369 | 2383 | 2520 | 2587 | 2688 | 2690 | 2721 | 2727 | 2813 | 2823 | |
| | 2855 | 2865 | 2885 | 2895 | 2905 | 2936 | 2964 | 2967 | 2981 | 2987 | 2999 | 3002 | |
| | 3015 | 3077 | 3089 | 3110 | 3275 | 3290 | 3293 | 3298 | 3299 | 3369 | 3388 | 3428 | |
| | 3431 | 3434 | 3437 | 3440 | 3443 | 3446 | 3449 | 3472 | 3486 | 3500 | 3508 | 3678 | |
| | 3709 | 3711 | 3716 | 3749 | 3771 | 3774 | 3779 | 3807 | 3827 | 3839 | 4056 | 4070 | |
| | 4086 | 4097 | 4100 | 4117 | 4118 | 4144 | 4146 | 4193 | 4208 | 4315 | 4316 | 4325 | |
| | 4406 | 4408 | 4450 | 4452 | 4623 | 4624 | 4628 | 4706 | 4712 | 4716 | 4992 | 5064 | |
| | 5103 | 5112 | 5113 | 5177 | 5200 | 5202 | 5282 | 5315 | 5345 | 5347 | 5410 | 5411 | |
| | 5439 | 5445 | 5547 | 5551 | 5553 | 5557 | 5649 | 5700 | 5801 | 5802 | 5957 | 6048 | |
| | 6085 | 6087 | 6095 | 6121 | 6157 | 6259 | 6291 | 6293 | 6305 | 6314 | 6384 | 6451 | |
| | 6605 | 6678 | 6930 | 6954 | 7532 | 7543 | 7546 | 7560 | 7562 | 7731 | 7833 | 8201 | |
| AC*# | 678 # | 2172 | 2173 | 2180 | 2370 | 2382 | 2488 | 2771 | 2780 | 3021 | 3042 | 3086 | |
| | 3116 | 4083 | 4095 | 4174 | 4176 | 4178 | 4186 | 4188 | 4195 | 4197 | 4209 | 4210 | |
| | 4211 | 4212 | 4213 | 4214 | 4326 | 4410 | 4432 | 4436 | 4439 | 4453 | 4454 | 4456 | |
| | 4459 | 4478 | 4480 | 4482 | 4483 | 4609 | 4610 | 4613 | 4614 | 4615 | 4618 | 4619 | |
| | 4620 | 4626 | 4627 | 5546 | 5637 | 5690 | 5692 | 5698 | 5757 | 5765 | 5803 | 5810 | |
| | 5954 | 5962 | 5972 | 5975 | 6005 | 6008 | 6021 | 6028 | 6035 | 6045 | 6083 | 6093 | |
| | 6113 | 6128 | 6145 | 6160 | 6162 | 6188 | 6191 | 6193 | 6200 | 6201 | 6202 | 6204 | |

RAM
VMA

XR#

#

(D) READ

| | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|
| 6210 | 6212 | 6215 | 6221 | 6223 | 6251 | 6253 | 6254 | 6258 | 6265 | 6266 | 6267 |
| 6271 | 6272 | 6279 | 6281 | 6286 | 6292 | 6308 | 6319 | 6331 | 6340 | 6346 | 6357 |
| 6360 | 6362 | 6366 | 6379 | 6385 | 6386 | 6509 | 6540 | 6545 | 6550 | 6587 | 6589 |
| 6693 | 6694 | 6806 | 6813 | 6820 | 6839 | 6841 | 6848 | 6868 | 6889 | 6925 | 6927 |
| 6933 | 6935 | 6939 | 6942 | 6947 | | | | | | | |
| 681 # | 2182 | 2184 | 6422 | 6427 | 6432 | 6434 | | | | | |
| 680 # | 2160 | 2268 | 2338 | 2352 | 2361 | 2365 | 2393 | 2404 | 2410 | 2423 | 2494 |
| 2526 | 2532 | 2537 | 2540 | 2590 | 2786 | 3405 | 3548 | 3572 | 3587 | 3608 | 3612 |
| 3616 | 3677 | 3694 | 3700 | 3722 | 3739 | 3745 | 3758 | 3816 | 3844 | 3974 | 4001 |
| 4013 | 4017 | 4040 | 4736 | 4764 | 4770 | 4891 | 5133 | 5156 | 5159 | 5205 | 5247 |
| 5911 | 5930 | 6350 | 6544 | 6590 | 6593 | 6621 | 6897 | 6912 | 6982 | 7104 | 7221 |
| 7225 | 7229 | 7233 | 7237 | 7277 | 7307 | 7316 | 7309 | 7316 | 7438 | 7443 | 7454 |
| 7491 | 7496 | 7587 | 7620 | 7667 | 7712 | 7715 | 7717 | 7727 | 7729 | 7826 | 7840 |
| 7845 | 7865 | 7913 | 7930 | 7931 | 7933 | 7934 | 7937 | 7938 | 7940 | 7941 | 7943 |
| 7944 | 7946 | 7947 | 7949 | 7950 | 7952 | 7953 | 7955 | 8220 | 8244 | 8679 | 8739 |
| 8745 | 8796 | 8810 | | | | | | | | | |
| 679 # | 2278 | 2288 | 2300 | 2315 | 2328 | 2344 | 3553 | 3555 | 3563 | 4743 | 4746 |
| 4748 | 4756 | 5922 | 5924 | 5927 | 6892 | 6894 | 6898 | 7656 | 7657 | 7665 | |
| 682 # | 2159 | 2167 | 2169 | 2170 | 2175 | 2176 | 3662 | 4689 | 4948 | 4961 | 4966 |
| 4971 | 4975 | 4976 | 4981 | 4987 | 5013 | 5026 | 5034 | 5039 | 5045 | 5175 | 5921 |
| 5934 | 5937 | 5967 | 5984 | 5991 | 5995 | 6012 | 6024 | 6029 | 6040 | 6044 | 6055 |
| 6056 | 6057 | 6059 | 6060 | 6061 | 6062 | 6072 | 6079 | 6103 | 6119 | 6126 | 6131 |
| 6132 | 6139 | 6142 | 6155 | 6173 | 6177 | 6180 | 6184 | 6294 | 6296 | 6337 | 6347 |
| 6352 | 6353 | 6355 | 6358 | 6371 | 6373 | 6377 | 6380 | 6381 | 6402 | 6552 | 6553 |
| 6576 | 6578 | 6614 | 6619 | 6629 | 6691 | 6697 | 6724 | 6728 | 6731 | 6736 | 6737 |
| 6744 | 6811 | 6818 | 6822 | 6824 | 6913 | 6929 | 6938 | 6944 | 6952 | 6953 | 6956 |
| 6964 | 7049 | 7071 | 7083 | 7151 | 7156 | 7208 | 7210 | 7212 | 7214 | 7216 | 7222 |
| 7226 | 7230 | 7234 | 7238 | 7245 | 7252 | 7254 | 7259 | 7263 | 7265 | 7284 | 7287 |
| 7297 | 7298 | 7301 | 7317 | 7318 | 7322 | 7402 | 7403 | 7651 | 7654 | 7659 | 7671 |
| 7673 | 7685 | 7691 | 7693 | 7838 | 7910 | 7912 | 7917 | 7921 | 7925 | 7935 | 7936 |
| 7954 | 7956 | 7957 | 7959 | 8197 | 8198 | 8207 | 8208 | 8210 | 8211 | 8218 | 8228 |
| 8230 | 8232 | 8234 | 8238 | 8239 | 8498 | 8538 | 8539 | 8540 | 8546 | 8724 | 8735 |
| 8741 | 8750 | 8754 | 8766 | | | | | | | | |
| 1348 # | 2546 | 2547 | 2549 | 2551 | 2552 | 2554 | 2556 | 2557 | 2559 | 2561 | 2562 |
| 2564 | 2583 | 2599 | 2600 | 2601 | 2602 | 2604 | 2605 | 2606 | 2607 | 2609 | 2610 |
| 2612 | 2614 | 2615 | 2617 | 2619 | 2620 | 2622 | 2624 | 2625 | 2627 | 2629 | 2630 |
| 2632 | 2634 | 2635 | 2637 | 2641 | 2642 | 2643 | 2644 | 2646 | 2647 | 2648 | 2649 |
| 2651 | 2652 | 2654 | 2656 | 2657 | 2659 | 2661 | 2662 | 2664 | 2666 | 2667 | 2669 |
| 2671 | 2672 | 2674 | 2676 | 2677 | 2679 | 2765 | 2766 | 2796 | 2797 | 2806 | 2807 |
| 2808 | 2809 | 2816 | 2817 | 2818 | 2819 | 2826 | 2827 | 2828 | 2829 | 2831 | 2832 |
| 2833 | 2834 | 2841 | 2848 | 2849 | 2850 | 2851 | 2858 | 2859 | 2860 | 2861 | 2868 |
| 2869 | 2870 | 2871 | 2878 | 2879 | 2880 | 2881 | 2888 | 2889 | 2898 | 2899 | 2900 |
| 2901 | 2909 | 2910 | 2911 | 2912 | 2919 | 2920 | 2921 | 2922 | 2929 | 2930 | 2931 |
| 2932 | 2939 | 2940 | 2951 | 2952 | 2953 | 2955 | 2956 | 2957 | 3185 | 3186 | 3187 |
| 3188 | 3189 | 3190 | 3201 | 3202 | 3203 | 3204 | 3205 | 3206 | 3207 | 3208 | 3219 |
| 3220 | 3221 | 3222 | 3223 | 3224 | 3225 | 3226 | 3236 | 3237 | 3238 | 3239 | 3240 |
| 3241 | 3242 | 3243 | 3358 | 3359 | 3360 | 3361 | 3362 | 3363 | 3364 | 3365 | 3375 |
| 3376 | 3377 | 3378 | 3379 | 3380 | 3381 | 3382 | 3392 | 3393 | 3394 | 3395 | 3396 |
| 3397 | 3398 | 3399 | 3408 | 3409 | 3410 | 3411 | 3412 | 3413 | 3414 | 3415 | 3475 |
| 3489 | 3646 | 3669 | 3765 | 4049 | 4050 | 4051 | 4052 | 4063 | 4064 | 4065 | 4066 |
| 4078 | 4079 | 4111 | 4112 | 4113 | 4114 | 4136 | 4137 | 4138 | 4139 | 4162 | 4303 |
| 4304 | 4305 | 4306 | 4308 | 4309 | 4310 | 4311 | 4380 | 4681 | 4682 | 4683 | 4684 |
| 4685 | 5259 | 5260 | 5261 | 5262 | 5263 | 5264 | 5265 | 5267 | 5268 | 5269 | 5270 |
| 5271 | 5272 | 5273 | 5304 | 5305 | 5306 | 5308 | 5309 | 5310 | 5311 | 5333 | 5334 |
| 5335 | 5337 | 5338 | 5339 | 5340 | 5383 | 5418 | 5419 | 5533 | 5534 | 5622 | 5683 |

| | | | | | | | | | | | | |
|----------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 5907 | | | | | | | | | | | |
| (U) READ CYCLE | 1125 # | 2227 | 2228 | 2230 | 2231 | 2250 | 2290 | 2296 | 2301 | 2307 | 2330 | 2334 |
| | 2363 | 2404 | 2415 | 2418 | 2429 | 3019 | 3325 | 3337 | 3452 | 3453 | 3456 | 3457 |
| | 3524 | 3525 | 3526 | 3531 | 3535 | 3538 | 3551 | 3565 | 3568 | 3593 | 3615 | 3635 |
| | 3708 | 3718 | 3752 | 3842 | 3999 | 4043 | 4217 | 4692 | 4701 | 4749 | 4753 | 4757 |
| | 4761 | 5094 | 5169 | 5252 | 5927 | 5928 | 5952 | 6099 | 6349 | 6376 | 6449 | 6454 |
| | 6510 | 6630 | 6746 | 6892 | 6893 | 6894 | 6895 | 6970 | 6978 | 7102 | 7220 | 7224 |
| | 7228 | 7232 | 7236 | 7275 | 7279 | 7315 | 7351 | 7439 | 7453 | 7486 | 7495 | 7504 |
| | 7578 | 7581 | 7622 | 7645 | 7666 | 7823 | 7858 | 7860 | 8521 | 8658 | 8672 | 8830 |
| (D) ROUND | 1336 # | 5262 | 5263 | 5264 | 5265 | 5270 | 5271 | 5272 | 5273 | 5308 | 5309 | 5310 |
| | 5311 | 5337 | 5338 | 5339 | 5340 | 5383 | 5419 | | | | | |
| (U) RSRC | 610 # | | | | | | | | | | | |
| AB | 612 # | 3662 | 7651 | | | | | | | | | |
| AQ | 611 # | | | | | | | | | | | |
| DO | 618 # | 2758 | 6154 | 6269 | 7439 | 7486 | 7578 | 7581 | 7617 | 7628 | 7634 | 7707 |
| | 7860 | 8224 | 8226 | 8240 | | | | | | | | |
| DA | 616 # | 2154 | 2157 | 2190 | 2201 | 2237 | 2241 | 2244 | 2247 | 2256 | 2259 | 2262 |
| | 2265 | 2516 | 3041 | 3579 | 3942 | 3982 | 4030 | 4394 | 4405 | 4799 | 5879 | 5881 |
| | 5883 | 5885 | 5887 | 5889 | 5891 | 5893 | 5895 | 6579 | 6591 | 6617 | 6657 | 6802 |
| | 7027 | 7030 | 7093 | 7248 | 7291 | 7292 | 7293 | 7427 | 7428 | 7429 | 7430 | 7431 |
| | 7432 | 7433 | 7435 | 7468 | 7500 | 8266 | 8527 | 8749 | | | | |
| DQ | 617 # | | | | | | | | | | | |
| OA | 615 # | 2319 | 2760 | 3688 | 3805 | 3811 | 4014 | 4215 | 5220 | 5550 | 5552 | 5652 |
| | 5653 | 5703 | 5705 | 5925 | 6963 | 7470 | 8808 | | | | | |
| OB | 614 # | 2737 | 2739 | | | | | | | | | |
| OO | 613 # | 3627 | 3641 | 3985 | 5224 | 5914 | 5958 | 6158 | 7458 | | | |
| (U) S# | 1000 # | 2174 | 2377 | 2396 | 2396 | 2411 | 2411 | 2427 | 2968 | 2972 | 2982 | |
| | 2984 | 3000 | 3003 | 3006 | 3011 | 3023 | 3038 | 3040 | 3067 | 3084 | 3093 | 3096 |
| | 3139 | 4118 | 4146 | 4183 | 4203 | 4223 | 4336 | 4433 | 4441 | 4703 | 4785 | 4808 |
| | 4813 | 4815 | 4817 | 4820 | 4883 | 4884 | 4885 | 4886 | 4887 | 4888 | 4896 | 4900 |
| | 4902 | 4904 | 4937 | 4953 | 4964 | 4977 | 4993 | 5038 | 5053 | 5212 | 5214 | 5232 |
| | 5234 | 5242 | 5243 | 5286 | 5288 | 5320 | 5325 | 5327 | 5353 | 5354 | 5367 | 5368 |
| | 5390 | 5394 | 5396 | 5397 | 5401 | 5409 | 5424 | 5427 | 5429 | 5480 | 5484 | 5486 |
| | 5489 | 5492 | 5500 | 5502 | 5503 | 5504 | 5505 | 5523 | 5528 | 5558 | 5569 | 5584 |
| | 5586 | 5627 | 5638 | 5642 | 5645 | 5655 | 5661 | 5667 | 5678 | 5716 | 5717 | 5727 |
| | 5741 | 5748 | 5751 | 5763 | 5780 | 5785 | 5787 | 5808 | 5833 | 5835 | 5841 | 5979 |
| | 5980 | 5982 | 6148 | 6293 | 6461 | 6463 | 6470 | 6519 | 6524 | 6526 | 6649 | 6651 |
| | 6653 | 6654 | 6660 | 6670 | 6853 | 7128 | 7146 | 7177 | 7187 | 7194 | 7406 | 7594 |
| | 7626 | 7680 | 7696 | 8261 | 8496 | | | | | | | |
| (U) SCAD | 977 # | | | | | | | | | | | |
| A | 985 # | 2174 | 3023 | 4118 | 4146 | 4183 | 4203 | 4223 | 4336 | 4433 | 4441 | 4699 |
| | 4713 | 4790 | 4791 | 4792 | 4793 | 4794 | 4899 | 4937 | 4953 | 4964 | 4977 | 4993 |
| | 5038 | 5053 | 5212 | 5214 | 5232 | 5234 | 5320 | 5367 | 5390 | 5394 | 5396 | 5397 |
| | 5401 | 5627 | 5638 | 5642 | 5655 | 5661 | 5716 | 5717 | 6293 | 6461 | 6463 | 6470 |
| | 6519 | 6649 | 6650 | 6660 | 6850 | 6871 | 6890 | 7128 | 7146 | 7177 | 7187 | 7194 |
| | 7406 | 7594 | 7626 | 7680 | 7696 | 8261 | 8496 | | | | | |
| A*2 | 978 # | 6642 | | | | | | | | | | |
| A+B | 982 # | 2377 | 2396 | 2396 | 2411 | 2411 | 2968 | 2972 | 2982 | 2984 | 3003 | 3006 |
| | 3011 | 3038 | 3040 | 3084 | 3093 | 4815 | 4820 | 4884 | 4885 | 4886 | 4887 | 4888 |
| | 4896 | 4904 | 5242 | 5243 | 5286 | 5288 | 5316 | 5325 | 5327 | 5368 | 5409 | 5411 |
| | 5423 | 5427 | 5480 | 5484 | 5486 | 5489 | 5492 | 5500 | 5502 | 5503 | 5504 | 5505 |
| | 5523 | 5528 | 5558 | 5569 | 5584 | 5586 | 5650 | 5667 | 5678 | 5741 | 5748 | 5751 |
| | 5763 | 5780 | 5785 | 5787 | 5808 | 5833 | 5835 | 5841 | 5979 | 5980 | 5982 | 6520 |
| | 6524 | 6652 | 6653 | 6654 | 6670 | 6958 | | | | | | |
| A-1 | 984 # | 2184 | 2988 | 2990 | 3066 | 3068 | 3070 | 3074 | 3100 | 3107 | 3138 | 3140 |

| | | | | | | | | | | | | |
|------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | 3142 | 3146 | 4249 | 4261 | 4267 | 4275 | 4285 | 4293 | 4548 | 4551 | 4591 | 4940 |
| | 4959 | 5059 | 5296 | 5298 | 5404 | 5428 | 5431 | 5434 | 5438 | 5602 | 5605 | 5612 |
| | 5630 | 6322 | 6466 | 6472 | 6667 | 7132 | 7148 | 7179 | 7196 | 7386 | 7399 | 7601 |
| | 7632 | 7687 | 7700 | 8264 | 8502 | | | | | | | |
| A-B | 981 # | 2965 | 3000 | 3067 | 3096 | 3139 | 4691 | 4696 | 4710 | 4731 | 4785 | 4808 |
| | 4813 | 4817 | 4883 | 4902 | 5347 | 5353 | 5354 | 5429 | 5701 | 5727 | 6032 | 6042 |
| | 6842 | 6845 | 6885 | 6887 | | | | | | | | |
| A-B-1 | 980 # | 5282 | 5287 | 5548 | 5568 | | | | | | | |
| A.AND.B | 983 # | 4703 | 4900 | 6148 | 6526 | 6651 | 6853 | | | | | |
| A.OR.B | 979 # | 2427 | 5645 | | | | | | | | | |
| (U) SCADA | 986 # | | | | | | | | | | | |
| BYTE1 | 990 # | 4691 | 4696 | 4699 | 4710 | 4790 | 4899 | 6032 | 6520 | 6650 | 6842 | 6850 |
| | 6885 | 6890 | 6958 | | | | | | | | | |
| BYTE2 | 991 # | 4791 | | | | | | | | | | |
| BYTE3 | 992 # | 4792 | | | | | | | | | | |
| BYTE4 | 993 # | 4793 | | | | | | | | | | |
| BYTE5 | 994 # | 4714 | 4794 | 6643 | 6872 | | | | | | | |
| PTR44 | 989 # | 4731 | 6042 | 6845 | 6887 | | | | | | | |
| S# | 988 # | 2174 | 2377 | 2396 | 2396 | 2411 | 2411 | 2427 | 2965 | 2968 | 2972 | 2982 |
| | 2984 | 3000 | 3003 | 3006 | 3011 | 3023 | 3038 | 3040 | 3067 | 3084 | 3093 | 3096 |
| | 3139 | 4118 | 4146 | 4183 | 4203 | 4223 | 4336 | 4433 | 4441 | 4703 | 4785 | 4808 |
| | 4813 | 4815 | 4817 | 4820 | 4883 | 4884 | 4885 | 4886 | 4887 | 4888 | 4896 | 4900 |
| | 4902 | 4904 | 4937 | 4953 | 4964 | 4977 | 4993 | 5038 | 5053 | 5212 | 5214 | 5232 |
| | 5234 | 5242 | 5243 | 5286 | 5288 | 5320 | 5325 | 5327 | 5353 | 5354 | 5367 | 5368 |
| | 5390 | 5394 | 5396 | 5397 | 5401 | 5409 | 5423 | 5427 | 5429 | 5480 | 5484 | 5486 |
| | 5489 | 5492 | 5500 | 5502 | 5503 | 5504 | 5505 | 5523 | 5528 | 5558 | 5569 | 5584 |
| | 5586 | 5627 | 5638 | 5642 | 5645 | 5655 | 5661 | 5667 | 5678 | 5716 | 5717 | 5727 |
| | 5741 | 5748 | 5751 | 5763 | 5780 | 5785 | 5787 | 5808 | 5833 | 5835 | 5841 | 5979 |
| | 5980 | 5982 | 6148 | 6293 | 6461 | 6463 | 6470 | 6519 | 6524 | 6526 | 6649 | 6651 |
| | 6653 | 6654 | 6660 | 6670 | 6853 | 7128 | 7146 | 7177 | 7187 | 7194 | 7406 | 7594 |
| | 7626 | 7680 | 7696 | 8261 | 8496 | | | | | | | |
| SC | 987 # | 2184 | 2988 | 2990 | 3066 | 3068 | 3070 | 3074 | 3100 | 3107 | 3138 | 3140 |
| | 3142 | 3146 | 4249 | 4261 | 4267 | 4275 | 4285 | 4293 | 4548 | 4551 | 4591 | 4940 |
| | 4959 | 5059 | 5282 | 5287 | 5296 | 5298 | 5316 | 5347 | 5404 | 5411 | 5428 | 5431 |
| | 5434 | 5438 | 5548 | 5568 | 5602 | 5605 | 5612 | 5630 | 5650 | 5701 | 6322 | 6466 |
| | 6472 | 6652 | 6667 | 7132 | 7148 | 7179 | 7196 | 7386 | 7399 | 7601 | 7632 | 7687 |
| | 7700 | 8264 | 8502 | | | | | | | | | |
| (U) SCADB | 995 # | | | | | | | | | | | |
| EXP | 997 # | 2396 | 2396 | 2411 | 2411 | 5282 | 5286 | 5288 | 5316 | 5347 | 5411 | 5548 |
| | 5558 | 5569 | 5650 | 5701 | | | | | | | | |
| FE | 996 # | 2427 | 2965 | 2968 | 2972 | 2982 | 2984 | 3000 | 3003 | 3006 | 3011 | 3038 |
| | 3040 | 3093 | 3096 | 4703 | 4785 | 4808 | 4813 | 4815 | 4820 | 4884 | 4885 | 4886 |
| | 4887 | 4888 | 4896 | 4900 | 4902 | 4904 | 5242 | 5243 | 5287 | 5325 | 5327 | 5353 |
| | 5354 | 5368 | 5424 | 5427 | 5429 | 5480 | 5484 | 5486 | 5489 | 5492 | 5500 | 5502 |
| | 5503 | 5504 | 5505 | 5523 | 5528 | 5568 | 5584 | 5586 | 5645 | 5667 | 5678 | 5727 |
| | 5741 | 5748 | 5751 | 5763 | 5780 | 5785 | 5787 | 5808 | 5833 | 5835 | 5841 | 5980 |
| | 5982 | 6148 | 6520 | 6524 | 6526 | 6651 | 6652 | 6653 | 6654 | 6670 | 6853 | |
| SHIFT | 998 # | 2377 | 3067 | 3084 | 3139 | 5409 | | | | | | |
| SIZE | 999 # | 4691 | 4696 | 4710 | 4731 | 4817 | 4883 | 5979 | 6032 | 6042 | 6842 | 6845 |
| | 6885 | 6887 | 6958 | | | | | | | | | |
| (U) SETFOV | 1081 # | 5349 | 5361 | 5714 | | | | | | | | |
| (U) SETFPD | 1087 # | 4231 | 4745 | 6388 | | | | | | | | |
| (U) SETNDV | 1082 # | 4338 | 4373 | 4404 | 4490 | 4997 | 5349 | 5361 | 5714 | | | |
| (U) SETOV | 1079 # | 4131 | 4132 | 4156 | 4218 | 4338 | 4373 | 4404 | 4490 | 4997 | 5349 | 5361 |
| | 5426 | 5714 | | | | | | | | | | |

| | | | | | | | | | | | | | |
|-------------|-------|------|------|------|------|------|------|------|------|------|------|------|--|
| (U) SHSTYLE | 824 # | | | | | | | | | | | | |
| ASHC | 829 # | 2990 | 3099 | 3102 | 3106 | 3109 | 4267 | 4275 | 4285 | 4293 | 4399 | 4403 | |
| | 4472 | 4476 | 4492 | 5298 | 5401 | 5403 | 5433 | 5577 | 5579 | 5583 | 5585 | 5601 | |
| | 5605 | 5612 | 5615 | 5617 | 5629 | 5670 | 5674 | 5709 | 5723 | 5741 | 5747 | 5750 | |
| | 5763 | 5780 | 5784 | 5786 | 5808 | 5833 | | | | | | | |
| DIV | 831 # | 4548 | 4551 | 4552 | 4595 | 4599 | 5480 | 5492 | 5500 | 5504 | 5597 | 5599 | |
| | 5600 | | | | | | | | | | | | |
| LSHC | 830 # | 3070 | 3074 | 4580 | 4583 | | | | | | | | |
| NORM | 825 # | 2424 | 2984 | 4226 | 4260 | 4433 | 4589 | 5207 | 5222 | 5223 | 5227 | 5228 | |
| | 5229 | 5243 | 5631 | 5721 | | | | | | | | | |
| ONES | 827 # | 4819 | 4822 | 4895 | 4898 | 5982 | | | | | | | |
| ROT | 828 # | 3006 | 3011 | | | | | | | | | | |
| ROTC | 832 # | 3142 | 3146 | | | | | | | | | | |
| ZERO | 826 # | | | | | | | | | | | | |
| (U) SKIP | 902 # | | | | | | | | | | | | |
| ACO | 910 # | 2483 | 3387 | 4688 | | | | | | | | | |
| ADEQO | 922 # | 3016 | 3331 | 3343 | 3434 | 3446 | 4121 | 4126 | 4328 | 4330 | 4334 | 4348 | |
| | 4393 | 4422 | 4452 | 4533 | 4611 | 4616 | 4621 | 4637 | 4644 | 4962 | 4991 | 5345 | |
| | 5371 | 5373 | 5446 | 5488 | 5512 | 5668 | 5719 | 5761 | 5806 | 6136 | 6189 | 6265 | |
| | 6268 | 6557 | 6594 | 6615 | 6738 | 7295 | | | | | | | |
| ADLEQO | 906 # | 2187 | 3986 | 3991 | 5388 | 5393 | 5507 | 5672 | 5675 | 5743 | 5746 | 5749 | |
| | 5752 | 5812 | 5912 | 5955 | 6094 | 6096 | 6178 | 6198 | 6382 | 6398 | 6452 | 6487 | |
| | 6490 | 6522 | 6986 | 7110 | 7117 | 7153 | 7456 | 7457 | 7648 | 7650 | 7664 | 8248 | |
| | 8511 | 8530 | 8532 | 8693 | 8711 | 8752 | | | | | | | |
| ADREQO | 907 # | 3085 | 3591 | 3606 | 5119 | 6323 | 6396 | 6751 | 7028 | 7031 | 7056 | 7058 | |
| | 7061 | 7063 | 7065 | 7331 | 7333 | 7337 | 7339 | 7341 | 7343 | 7345 | 7347 | 7444 | |
| | 7497 | 7517 | 7589 | 7616 | 8199 | 8491 | 8680 | 8697 | 8700 | 8705 | | | |
| CRYO | 905 # | 3684 | 3727 | 3756 | 5357 | 5481 | 5483 | 5485 | 5487 | 5487 | | | |
| CRY1 | 918 # | 4084 | 4096 | 5797 | 6213 | 6219 | 6232 | 6423 | 6434 | | | | |
| CRY2 | 913 # | 5770 | 5830 | | | | | | | | | | |
| DPO | 914 # | 2397 | 2412 | 2571 | 2749 | 2756 | 3039 | 3111 | 3328 | 3340 | 3431 | 3443 | |
| | 3773 | 3776 | 3781 | 4103 | 4128 | 4150 | 4153 | 4208 | 4216 | 4317 | 4353 | 4355 | |
| | 4360 | 4365 | 4369 | 4391 | 4408 | 4420 | 4427 | 4442 | 4461 | 4479 | 4531 | 4538 | |
| | 4553 | 4994 | 5138 | 5162 | 5248 | 5284 | 5288 | 5317 | 5347 | 5355 | 5389 | 5411 | |
| | 5442 | 5549 | 5651 | 5698 | 5702 | 5713 | 5726 | 5963 | 5976 | 5993 | 6006 | 6013 | |
| | 6031 | 6073 | 6075 | 6097 | 6114 | 6117 | 6123 | 6163 | 6255 | 6260 | 6297 | 6306 | |
| | 6339 | 6356 | 6428 | 6478 | 6504 | 6695 | 6725 | 6767 | 6821 | 6940 | 7105 | 7250 | |
| | 8650 | | | | | | | | | | | | |
| DP18 | 915 # | 2376 | 2735 | 2742 | 5936 | 6183 | 6387 | 6476 | 6799 | 6800 | 7483 | 8653 | |
| | 8663 | 8686 | | | | | | | | | | | |
| EXECUTE | 924 # | 7852 | | | | | | | | | | | |
| FPD | 909 # | 4170 | 4691 | 4696 | 4710 | 6287 | | | | | | | |
| INT | 911 # | 5142 | 6033 | 8212 | 8733 | 8764 | 8815 | | | | | | |
| IOLGL | 903 # | 2433 | 3536 | 3539 | | | | | | | | | |
| IOT | 916 # | | | | | | | | | | | | |
| JFCL | 917 # | 3598 | | | | | | | | | | | |
| KERNEL | 908 # | 3529 | 3533 | 3534 | 3541 | 3650 | 6965 | | | | | | |
| LE | 912 # | 3334 | 3346 | 3437 | 3449 | 4387 | 5021 | 5365 | 5689 | 6336 | 7261 | 7918 | |
| LLE | 904 # | 7657 | | | | | | | | | | | |
| SC | 923 # | 2184 | 2988 | 2990 | 3066 | 3068 | 3070 | 3074 | 3091 | 3100 | 3107 | 3138 | |
| | 3140 | 3142 | 3146 | 4249 | 4261 | 4267 | 4275 | 4285 | 4293 | 4548 | 4551 | 4591 | |
| | 4940 | 4959 | 5059 | 5296 | 5298 | 5404 | 5428 | 5431 | 5434 | 5438 | 5551 | 5553 | |
| | 5602 | 5605 | 5612 | 5630 | 6322 | 6466 | 6472 | 6658 | 6667 | 7132 | 7148 | 7179 | |
| | 7196 | 7386 | 7399 | 7601 | 7632 | 7700 | 8264 | 8502 | | | | | |
| TRAP CYCLE | 920 # | 8751 | | | | | | | | | | | |

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES, DTN: 223-7881

| | | | | | | | | | | | | | | |
|-------------|--------------|--------|------|------|------|------|------|------|------|------|------|------|------|--|
| | TXXX | 919 # | 3284 | | | | | | | | | | | |
| | -1 MS | 927 # | 5134 | 6028 | 7299 | | | | | | | | | |
| | -CONTINUE | 926 # | 7854 | | | | | | | | | | | |
| | -IO BUSY | 925 # | 7682 | 7688 | 7697 | 7705 | | | | | | | | |
| (U) SPEC | | 788 # | 2568 | 2574 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 | 2803 | |
| | | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 | 4123 | 4129 | 4152 | |
| | | 4154 | 8517 | 8817 | | | | | | | | | | |
| | APR EN | 802 # | 2165 | 7070 | 7155 | | | | | | | | | |
| | APR FLAGS | 800 # | 7075 | 7079 | 7269 | | | | | | | | | |
| | ASHOV | 809 # | 2990 | 3106 | 3109 | | | | | | | | | |
| | CLR IO BUSY | 792 # | 7575 | 7613 | | | | | | | | | | |
| | CLR IO LATCH | 791 # | 7660 | 7668 | 7672 | 7675 | 7684 | 7690 | 7699 | 7704 | | | | |
| | CLRCLK | 790 # | 7246 | 7300 | 8736 | | | | | | | | | |
| | CLRCSH | 801 # | 7381 | 7382 | 7385 | | | | | | | | | |
| | EXPTST | 810 # | 5489 | 5505 | 5835 | | | | | | | | | |
| | FLAGS | 811 # | 2216 | 2217 | 2577 | 3403 | 3419 | 3486 | 3500 | 3556 | 3557 | 3560 | 3561 | |
| | | 3597 | 3618 | 3619 | 3689 | 3701 | 3757 | 3787 | 3791 | 3806 | 3812 | 4003 | 4004 | |
| | | 4005 | 4057 | 4071 | 4088 | 4098 | 4101 | 4131 | 4132 | 4156 | 4206 | 4218 | 4228 | |
| | | 4231 | 4338 | 4373 | 4404 | 4490 | 4639 | 4641 | 4706 | 4723 | 4745 | 4997 | 5349 | |
| | | 5361 | 5426 | 5714 | 6388 | 6389 | 6987 | 7459 | 7474 | 8755 | | | | |
| | INHCRY18 | 806 # | 3509 | 3679 | 3726 | 3755 | 3775 | 3780 | 5029 | 5104 | | | | |
| | LDACBLK | 812 # | 2162 | 7125 | 7142 | | | | | | | | | |
| | LDINST | 813 # | 2269 | 3654 | 6983 | 7866 | | | | | | | | |
| | LDPAGE | 793 # | 7376 | 8533 | | | | | | | | | | |
| | LDPI | 808 # | 3607 | 7362 | 7437 | 7919 | | | | | | | | |
| | LDPXCT | 795 # | 3659 | | | | | | | | | | | |
| | LOADIR | 807 # | 5917 | | | | | | | | | | | |
| | LOADXR | 798 # | 2340 | 3549 | 3573 | 3588 | 4699 | 4718 | 4766 | 5918 | 5930 | 6849 | 6886 | |
| | | 6888 | 6897 | 7663 | | | | | | | | | | |
| | MEMCLR | 803 # | 2160 | 3608 | 7438 | 7840 | 7913 | 8244 | 8739 | 8745 | | | | |
| | NICOND | 794 # | 2227 | 2228 | 2230 | 2231 | 2291 | 2297 | 2521 | 3042 | 3088 | 3116 | 3325 | |
| | | 3337 | 3428 | 3452 | 3453 | 3456 | 3457 | 3524 | 3525 | 3551 | 3593 | 3710 | 3786 | |
| | | 3793 | 4217 | 4481 | 4484 | 4692 | 5439 | 5445 | 5804 | 7351 | 7622 | 8201 | | |
| | PREV | 797 # | 7824 | 7832 | | | | | | | | | | |
| | PXCT OFF | 805 # | 2239 | 2242 | 2245 | 2257 | 2260 | 2263 | 2277 | | | | | |
| | SWEEP | 804 # | 7394 | 7395 | 7398 | | | | | | | | | |
| | WAIT | 796 # | 2409 | | | | | | | | | | | |
| | # | 789 # | 7848 | 7851 | 7914 | | | | | | | | | |
| (U) STATE | | 1003 # | | | | | | | | | | | | |
| | BLT | 1005 # | 5105 | | | | | | | | | | | |
| | COMP-DST | 1012 # | 6122 | | | | | | | | | | | |
| | CVTDB | 1011 # | 6169 | | | | | | | | | | | |
| | DST | 1008 # | 6084 | | | | | | | | | | | |
| | DSTF | 1010 # | 6022 | | | | | | | | | | | |
| | EDIT-DST | 1014 # | 6141 | 6351 | 6378 | 6597 | 6623 | 8772 | 8776 | | | | | |
| | EDIT-S+D | 1015 # | 6563 | | | | | | | | | | | |
| | EDIT-SRC | 1013 # | 6115 | 6502 | 6559 | 8770 | 8778 | 8780 | 8784 | | | | | |
| | MAP | 1006 # | 8200 | | | | | | | | | | | |
| | SIMPLE | 1004 # | | | | | | | | | | | | |
| | SRC | 1007 # | 5971 | 6000 | 6070 | 8774 | | | | | | | | |
| | SRC+DST | 1009 # | 5997 | 6076 | | | | | | | | | | |
| (U) SWITCH% | | | | | | | | | | | | | | |
| | FULL | 397 | 1294 | 1296 | 2171 | 2191 | 2196 | 2200 | 4165 | 4233 | 4236 | 4383 | 4493 | |
| | | 4496 | 4564 | 4629 | 4915 | 5066 | 5070 | 7149 | 7157 | | | | | |
| | INH CST | 400 | 1240 | 1242 | 1244 | 8003 | 8005 | 8352 | 8355 | 8356 | 8359 | 8477 | 8480 | |

| | | | | | | | | | | | | | |
|------------------|--------|--------|------|------|------|------|------|------|------|------|------|------|------|
| | | 8481 | 8484 | 8555 | 8564 | 8565 | 8581 | | | | | | |
| | KIPAGE | 409 | 419 | 421 | 1264 | 1266 | 1268 | 3946 | 3956 | 4008 | 4019 | 7159 | 7166 |
| | | 7172 | 7174 | 8269 | 8273 | 8278 | 8281 | 8645 | 8691 | 8800 | 8803 | 8807 | 8812 |
| | KLPAGE | 8 | 412 | 416 | 418 | 420 | 1270 | 1272 | 1274 | 3947 | 3951 | 3957 | 3971 |
| | | 7160 | 7165 | 7168 | 7170 | 8268 | 8277 | 8283 | 8486 | 8549 | 8643 | 8799 | 8805 |
| | | 8822 | 8827 | | | | | | | | | | |
| | NOCST | 403 | 424 | 426 | 1246 | 1248 | 1250 | 8347 | 8422 | 8426 | 8440 | 8443 | 8444 |
| | | 8448 | 8474 | 8493 | 8550 | 8582 | | | | | | | |
| | NONSTD | 9 | 428 | 1252 | 1254 | 1256 | | | | | | | |
| | SIM | 394 | 2209 | 2214 | 2221 | | | | | | | | |
| | UBABLT | 7 | 406 | 415 | 422 | 1258 | 1260 | 1262 | 5180 | 5254 | 7796 | 7798 | 7800 |
| (U) T | | 940 # | | | | | | | | | | | |
| | 2T | 943 # | 4408 | 4421 | 5288 | 5558 | 6132 | 6202 | 6266 | 6554 | 6691 | 6736 | 7301 |
| | | 7658 | | | | | | | | | | | |
| | 3T | 944 # | 2187 | 2412 | 2422 | 2577 | 3110 | 3284 | 3369 | 3403 | 3419 | 3509 | 3599 |
| | | 3683 | 3718 | 3726 | 3751 | 3755 | 3772 | 3777 | 3782 | 3819 | 3833 | 4057 | 4071 |
| | | 4174 | 4176 | 4195 | 4229 | 4366 | 4370 | 4420 | 4426 | 4427 | 4428 | 4445 | 4449 |
| | | 4470 | 4478 | 4559 | 4624 | 4641 | 4699 | 4714 | 4785 | 4803 | 4899 | 4943 | 4950 |
| | | 5030 | 5046 | 5050 | 5116 | 5118 | 5137 | 5161 | 5248 | 5282 | 5316 | 5357 | 5388 |
| | | 5389 | 5392 | 5424 | 5427 | 5481 | 5483 | 5485 | 5487 | 5546 | 5547 | 5550 | 5552 |
| | | 5651 | 5652 | 5653 | 5697 | 5701 | 5703 | 5705 | 5713 | 5796 | 5802 | 5962 | 5980 |
| | | 5982 | 6030 | 6034 | 6097 | 6204 | 6218 | 6232 | 6264 | 6267 | 6352 | 6355 | 6387 |
| | | 6388 | 6416 | 6422 | 6425 | 6434 | 6459 | 6473 | 6500 | 6520 | 6521 | 6643 | 6650 |
| | | 6694 | 6753 | 6843 | 6850 | 6871 | 6885 | 6889 | 6892 | 6894 | 6930 | 6985 | 7104 |
| | | 7110 | 7116 | 7153 | 7418 | 7447 | 7455 | 7456 | 7457 | 7492 | 7496 | 7498 | |
| | | 7665 | 8525 | 8542 | 8665 | 8676 | 8760 | 8790 | | | | | |
| | 4T | 945 # | 3015 | 3486 | 3500 | 4083 | 4088 | 4095 | 4098 | 4100 | 4990 | 4995 | 5976 |
| | | 6006 | 6183 | 6188 | 6213 | 6254 | 6427 | 6475 | 6615 | 6738 | 6939 | 7918 | 8215 |
| | | 8653 | 8663 | | | | | | | | | | |
| | 5T | 946 # | 7687 | 7704 | | | | | | | | | |
| (D) TEST | | 1349 # | 2548 | 2548 | 2549 | 2549 | 2553 | 2553 | 2554 | 2554 | 2558 | 2558 | 2559 |
| | | 2559 | 2563 | 2563 | 2564 | 2564 | 2583 | 2601 | 2601 | 2602 | 2602 | 2606 | 2606 |
| | | 2607 | 2607 | 2611 | 2611 | 2612 | 2612 | 2616 | 2616 | 2617 | 2617 | 2621 | 2621 |
| | | 2622 | 2622 | 2626 | 2626 | 2627 | 2627 | 2631 | 2631 | 2632 | 2632 | 2636 | 2636 |
| | | 2637 | 2637 | 2643 | 2643 | 2644 | 2644 | 2648 | 2648 | 2649 | 2649 | 2653 | 2653 |
| | | 2654 | 2654 | 2658 | 2658 | 2659 | 2659 | 2663 | 2663 | 2664 | 2664 | 2668 | 2668 |
| | | 2669 | 2669 | 2673 | 2673 | 2674 | 2674 | 2678 | 2678 | 2679 | 2679 | 2775 | 2798 |
| | | 2798 | 2799 | 2799 | 2808 | 2808 | 2809 | 2809 | 2818 | 2818 | 2819 | 2819 | 2828 |
| | | 2828 | 2829 | 2829 | 2833 | 2833 | 2834 | 2834 | 2843 | 2843 | 2844 | 2844 | 2850 |
| | | 2850 | 2851 | 2851 | 2860 | 2860 | 2861 | 2861 | 2870 | 2870 | 2871 | 2871 | 2880 |
| | | 2880 | 2881 | 2881 | 2890 | 2890 | 2891 | 2891 | 2900 | 2900 | 2901 | 2901 | 2911 |
| | | 2911 | 2912 | 2912 | 2921 | 2921 | 2922 | 2922 | 2931 | 2931 | 2932 | 2932 | 2941 |
| | | 2941 | 2942 | 2942 | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 | 3408 |
| | | 3409 | 3410 | 3411 | 3412 | 3413 | 3414 | 3415 | 4051 | 4051 | 4052 | 4052 | 4065 |
| | | 4065 | 4066 | 4066 | 4113 | 4113 | 4114 | 4114 | 4138 | 4138 | 4139 | 4139 | 4305 |
| | | 4305 | 4306 | 4306 | 4310 | 4310 | 4311 | 4311 | 4682 | 4684 | 5260 | 5260 | 5261 |
| | | 5261 | 5264 | 5264 | 5265 | 5265 | 5268 | 5268 | 5269 | 5269 | 5272 | 5272 | 5273 |
| | | 5273 | 5305 | 5305 | 5306 | 5306 | 5310 | 5310 | 5311 | 5311 | 5334 | 5334 | 5335 |
| | | 5335 | 5339 | 5339 | 5340 | 5340 | 7203 | | | | | | |
| (U) TRAP1 | | 1096 # | 4131 | 4132 | 4156 | 4218 | 4338 | 4373 | 4404 | 4490 | 4997 | 5349 | 5361 |
| | | 5426 | 5714 | | | | | | | | | | |
| (U) TRAP2 | | 1095 # | 3701 | 3757 | 3787 | 3791 | | | | | | | |
| (U) VECTOR CYCLE | | 1167 # | 7487 | | | | | | | | | | |
| (D) VMA | | 1351 # | 2547 | 2552 | 2557 | 2562 | 2600 | 2605 | 2610 | 2615 | 2620 | 2625 | 2630 |
| | | 2635 | 2642 | 2647 | 2652 | 2657 | 2662 | 2667 | 2672 | 2677 | 2796 | 2797 | 2807 |

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES6, DTN: 223-7881

(U) WAIT

| | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|
| 2817 | 2827 | 2832 | 2849 | 2859 | 2869 | 2879 | 2888 | 2889 | 2899 | 2910 | 2920 |
| 2930 | 2939 | 2940 | 2951 | 2952 | 2953 | 2955 | 2956 | 2957 | 3475 | 3489 | 3515 |
| 3765 | 4050 | 4064 | 4112 | 4137 | 4304 | 4309 | | | | | |
| 1156 # | 2227 | 2227 | 2228 | 2228 | 2230 | 2230 | 2231 | 2231 | 2250 | 2250 | 2267 |
| 2290 | 2290 | 2296 | 2296 | 2301 | 2307 | 2320 | 2330 | 2334 | 2338 | 2351 | 2360 |
| 2363 | 2364 | 2392 | 2403 | 2404 | 2404 | 2415 | 2418 | 2422 | 2429 | 2429 | 2493 |
| 2525 | 2531 | 2536 | 2539 | 2588 | 2588 | 2589 | 2784 | 2784 | 2786 | 2790 | 2790 |
| 3019 | 3019 | 3325 | 3325 | 3337 | 3337 | 3404 | 3404 | 3405 | 3452 | 3452 | 3453 |
| 3453 | 3456 | 3456 | 3457 | 3457 | 3524 | 3524 | 3525 | 3525 | 3526 | 3531 | 3535 |
| 3538 | 3547 | 3551 | 3551 | 3565 | 3568 | 3571 | 3586 | 3593 | 3593 | 3612 | 3615 |
| 3616 | 3629 | 3629 | 3632 | 3632 | 3635 | 3640 | 3640 | 3676 | 3681 | 3681 | 3693 |
| 3699 | 3708 | 3708 | 3718 | 3721 | 3731 | 3731 | 3736 | 3736 | 3738 | 3744 | 3752 |
| 3758 | 3814 | 3814 | 3815 | 3825 | 3825 | 3842 | 3843 | 3954 | 3973 | 3978 | 3978 |
| 3999 | 4000 | 4012 | 4016 | 4025 | 4025 | 4035 | 4035 | 4039 | 4043 | 4217 | 4217 |
| 4692 | 4692 | 4701 | 4701 | 4729 | 4729 | 4735 | 4735 | 4736 | 4749 | 4753 | 4757 |
| 4761 | 4763 | 4769 | 4890 | 4890 | 4891 | 5094 | 5122 | 5122 | 5130 | 5130 | 5132 |
| 5146 | 5146 | 5151 | 5151 | 5155 | 5158 | 5169 | 5204 | 5245 | 5245 | 5247 | 5252 |
| 5911 | 5927 | 5928 | 5930 | 5952 | 6099 | 6349 | 6350 | 6376 | 6449 | 6454 | 6510 |
| 6542 | 6542 | 6544 | 6588 | 6588 | 6590 | 6593 | 6621 | 6630 | 6746 | 6892 | 6893 |
| 6894 | 6895 | 6897 | 6911 | 6967 | 6970 | 6975 | 6978 | 6982 | 7102 | 7103 | 7220 |
| 7221 | 7224 | 7225 | 7228 | 7229 | 7232 | 7233 | 7236 | 7237 | 7275 | 7276 | 7279 |
| 7304 | 7304 | 7307 | 7308 | 7308 | 7309 | 7315 | 7316 | 7351 | 7351 | 7369 | 7369 |
| 7370 | 7442 | 7453 | 7454 | 7469 | 7469 | 7490 | 7495 | 7496 | 7504 | 7584 | 7586 |
| 7619 | 7620 | 7622 | 7622 | 7645 | 7666 | 7667 | 7712 | 7715 | 7717 | 7725 | 7727 |
| 7729 | 7823 | 7825 | 7831 | 7831 | 7843 | 7845 | 7846 | 7846 | 7858 | 7858 | 7864 |
| 7929 | 7930 | 7931 | 7932 | 7932 | 7933 | 7934 | 7937 | 7938 | 7939 | 7939 | 7940 |
| 7941 | 7942 | 7942 | 7943 | 7944 | 7945 | 7945 | 7946 | 7947 | 7948 | 7948 | 7949 |
| 7950 | 7951 | 7951 | 7952 | 7953 | 7955 | 8543 | 8658 | 8672 | 8678 | 8794 | 8795 |
| 8809 | 8830 | 8833 | 8833 | | | | | | | | |

(U) WORK

| | | | | | | | | | | | |
|----------|--------|------|------|------|------|------|------|------|--|--|--|
| | 1019 # | | | | | | | | | | |
| ACO | 1060 # | | | | | | | | | | |
| AC1 | 1061 # | | | | | | | | | | |
| AC2 | 1062 # | | | | | | | | | | |
| AC3 | 1063 # | | | | | | | | | | |
| ADJBPW | 1038 # | 5013 | 5034 | | | | | | | | |
| ADJP | 1033 # | 4948 | 4975 | | | | | | | | |
| ADJPTR | 1035 # | 4689 | 4966 | 5028 | | | | | | | |
| ADJQ1 | 1036 # | 4971 | 4989 | | | | | | | | |
| ADJR2 | 1037 # | 4981 | 5048 | | | | | | | | |
| ADJS | 1034 # | 4961 | 4976 | 5039 | | | | | | | |
| APR | 1040 # | 2167 | 7049 | 7071 | 7083 | 7151 | 7156 | 8198 | | | |
| BADWO | 1020 # | 8238 | | | | | | | | | |
| BADW1 | 1021 # | 8239 | | | | | | | | | |
| BDH | 1050 # | 6296 | 6371 | 6380 | 6402 | | | | | | |
| BDL | 1051 # | 6294 | 6337 | 6377 | 6381 | | | | | | |
| CBR | 1030 # | 7210 | 7226 | | | | | | | | |
| CMS | 1047 # | 6119 | 6131 | 6132 | 6142 | | | | | | |
| CSTM | 1031 # | 7214 | 7234 | | | | | | | | |
| DDIV SGN | 1064 # | | | | | | | | | | |
| DECHI | 1069 # | 2176 | 2178 | 6425 | | | | | | | |
| DECLO | 1068 # | 2175 | 2177 | 6416 | | | | | | | |
| DIV | 1023 # | | | | | | | | | | |
| DVSOR H | 1065 # | | | | | | | | | | |
| DVSOR L | 1066 # | | | | | | | | | | |
| EO | 1042 # | 5921 | 6347 | 6619 | 6629 | | | | | | |

| | | | | | | | | | | | | |
|-----------------|--------|------|------|------|------|------|------|------|------|------|------|------|
| E1 | 1043 # | 5934 | 5937 | 6373 | 6552 | 6553 | 6728 | 6731 | 6736 | 6744 | | |
| FILL | 1046 # | 6103 | 6126 | 6139 | 6614 | 6691 | 6697 | 6811 | 6913 | | | |
| FSIG | 1048 # | 6576 | 6578 | | | | | | | | | |
| HSBADR | 1039 # | 2159 | 7216 | 7238 | 7912 | 7917 | | | | | | |
| MSK | 1045 # | 5984 | 6155 | 6737 | | | | | | | | |
| MUL | 1022 # | | | | | | | | | | | |
| PERIOD | 1056 # | 7265 | 7317 | 7322 | | | | | | | | |
| PTA.E | 1072 # | 7403 | | | | | | | | | | |
| PTA.U | 1073 # | 7402 | | | | | | | | | | |
| PUR | 1032 # | 7212 | 7230 | | | | | | | | | |
| SBR | 1029 # | 7208 | 7222 | | | | | | | | | |
| SLEN | 1044 # | 5967 | 5991 | 5995 | 6012 | 6024 | 6040 | 6044 | 6072 | 6079 | 6173 | 6177 |
| | 6180 | 6184 | 6352 | 6353 | 6355 | 6358 | 6724 | 6818 | 6822 | 6824 | 6938 | 6952 |
| | 6956 | | | | | | | | | | | |
| SV.ARX | 1026 # | 6056 | 6061 | 7838 | 7910 | 7921 | 7936 | 7959 | 8211 | 8538 | 8766 | |
| SV.AR | 1025 # | 6029 | 6059 | 7935 | 7956 | 8207 | 8546 | | | | | |
| SV.AR1 | 1075 # | | | | | | | | | | | |
| SV.BRX | 1028 # | 5175 | 6057 | 6062 | 6929 | 6944 | 6953 | 8208 | 8540 | | | |
| SV.BR | 1027 # | 6055 | 6060 | 8218 | 8228 | 8230 | 8232 | 8234 | 8539 | 8724 | | |
| SV.VMA | 1024 # | 7685 | 7691 | 7693 | 7925 | 7954 | 7957 | 8197 | 8210 | 8498 | 8741 | 8750 |
| TIME0 | 1054 # | 2169 | 7254 | 7287 | 7297 | | | | | | | |
| TIME1 | 1055 # | 2170 | 7245 | 7252 | 7284 | 7298 | 7301 | 8735 | | | | |
| TRAPPC | 1074 # | 6964 | 8754 | | | | | | | | | |
| TTG | 1057 # | 7259 | 7263 | 7318 | | | | | | | | |
| YSAVE | 1071 # | 3662 | 7651 | 7654 | 7659 | 7671 | 7673 | | | | | |
| (D) WRITE | 1352 # | | | | | | | | | | | |
| (U) WRITE CYCLE | 1127 # | 2568 | 2574 | 2588 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 |
| | 2784 | 2790 | 2803 | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 |
| | 3404 | 3629 | 3632 | 3640 | 3681 | 3731 | 3736 | 3814 | 3825 | 3954 | 3978 | 4025 |
| | 4035 | 4057 | 4071 | 4123 | 4129 | 4152 | 4154 | 4729 | 4735 | 4890 | 5122 | 5130 |
| | 5146 | 5151 | 5245 | 5443 | 5447 | 5493 | 5515 | 6542 | 6588 | 7304 | 7308 | 7369 |
| | 7469 | 7617 | 7628 | 7634 | 7831 | 7843 | 7846 | 7929 | 7932 | 7939 | 7942 | 7945 |
| | 7948 | 7951 | 8517 | 8522 | 8712 | 8794 | 8817 | 8833 | | | | |
| (U) WRITE TEST | 1126 # | 2568 | 2574 | 2588 | 2688 | 2690 | 2723 | 2729 | 2737 | 2739 | 2758 | 2760 |
| | 2784 | 2790 | 2803 | 2813 | 2823 | 2855 | 2865 | 2885 | 2895 | 2906 | 2916 | 2946 |
| | 3404 | 3629 | 3632 | 3640 | 3681 | 3731 | 3736 | 3814 | 3825 | 3954 | 3978 | 4025 |
| | 4035 | 4057 | 4071 | 4123 | 4129 | 4152 | 4154 | 4729 | 4735 | 4890 | 5122 | 5130 |
| | 5146 | 5151 | 5245 | 5443 | 5447 | 5493 | 5515 | 6542 | 6588 | 7304 | 7308 | 7369 |
| | 7439 | 7453 | 7469 | 7486 | 7495 | 7578 | 7581 | 7617 | 7628 | 7634 | 7831 | 7843 |
| | 7846 | 7860 | 7929 | 7932 | 7939 | 7942 | 7945 | 7948 | 7951 | 8248 | 8517 | 8523 |
| | 8794 | 8817 | 8830 | 8833 | | | | | | | | |
| (U) WRU CYCLE | 1163 # | 7440 | | | | | | | | | | |
| (U) # | 974 # | 2148 | 2150 | 2152 | 2154 | 2157 | 2187 | 2201 | 2237 | 2241 | 2244 | 2256 |
| | 2259 | 2262 | 2419 | 2421 | 2434 | 2456 | 2457 | 2744 | 2746 | 2751 | 2753 | 2770 |
| | 3041 | 3112 | 3114 | 3528 | 3537 | 3540 | 3542 | 3543 | 3544 | 3577 | 3582 | 3585 |
| | 3591 | 3605 | 3606 | 3626 | 3639 | 3725 | 3754 | 3817 | 3832 | 3918 | 3920 | 3922 |
| | 3924 | 3926 | 3928 | 3930 | 3932 | 3939 | 3942 | 3982 | 3987 | 3989 | 3992 | 3994 |
| | 4022 | 4030 | 4037 | 4104 | 4105 | 4357 | 4394 | 4395 | 4405 | 4413 | 4458 | 4511 |
| | 4537 | 4544 | 4635 | 4800 | 4944 | 4946 | 4951 | 4969 | 4973 | 4979 | 4983 | 5012 |
| | 5017 | 5051 | 5055 | 5062 | 5115 | 5211 | 5216 | 5225 | 5231 | 5236 | 5289 | 5290 |
| | 5293 | 5294 | 5318 | 5319 | 5323 | 5351 | 5352 | 5388 | 5392 | 5412 | 5413 | 5550 |
| | 5552 | 5632 | 5652 | 5653 | 5672 | 5675 | 5703 | 5705 | 5879 | 5881 | 5883 | 5885 |
| | 5887 | 5889 | 5891 | 5893 | 5895 | 5912 | 5915 | 5939 | 5955 | 5959 | 6094 | 6096 |
| | 6154 | 6159 | 6166 | 6171 | 6178 | 6182 | 6198 | 6240 | 6256 | 6257 | 6263 | 6269 |
| | 6289 | 6302 | 6304 | 6311 | 6316 | 6323 | 6326 | 6382 | 6396 | 6398 | 6405 | 6430 |

| | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| 6436 | 6452 | 6464 | 6467 | 6475 | 6487 | 6490 | 6507 | 6522 | 6580 | 6591 | 6600 |
| 6602 | 6618 | 6637 | 6658 | 6663 | 6672 | 6675 | 6751 | 6773 | 6777 | 6781 | 6785 |
| 6789 | 6793 | 6803 | 6809 | 6907 | 6918 | 6985 | 6990 | 6999 | 7000 | 7001 | 7005 |
| 7006 | 7007 | 7008 | 7013 | 7014 | 7015 | 7018 | 7019 | 7020 | 7021 | 7022 | 7023 |
| 7024 | 7025 | 7027 | 7030 | 7036 | 7051 | 7052 | 7055 | 7056 | 7058 | 7061 | 7063 |
| 7065 | 7077 | 7087 | 7090 | 7094 | 7108 | 7113 | 7116 | 7120 | 7135 | 7152 | 7153 |
| 7154 | 7180 | 7189 | 7197 | 7217 | 7239 | 7248 | 7267 | 7282 | 7334 | 7335 | 7378 |
| 7380 | 7391 | 7393 | 7420 | 7421 | 7422 | 7423 | 7424 | 7425 | 7426 | 7427 | 7428 |
| 7429 | 7430 | 7431 | 7432 | 7433 | 7436 | 7447 | 7456 | 7457 | 7472 | 7492 | 7499 |
| 7589 | 7597 | 7603 | 7616 | 7648 | 7650 | 7664 | 7707 | 7793 | 7802 | 7804 | 7806 |
| 7808 | 7810 | 7812 | 7848 | 7851 | 7862 | 7914 | 8192 | 8196 | 8199 | 8224 | 8226 |
| 8240 | 8250 | 8253 | 8259 | 8267 | 8492 | 8505 | 8509 | 8513 | 8515 | 8528 | 8529 |
| 8531 | 8536 | 8652 | 8662 | 8666 | 8676 | 8681 | 8697 | 8700 | 8703 | 8706 | 8715 |
| 8720 | 8728 | 8790 | | | | | | | | | |

| | Dcode | Loc'n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|-------|-------|------|------|------|------|------|------|------|------|
| D | 0000 | | 3904 | 5848 | 5849 | 5850 | 5851 | 5852 | 5853 | 5854 |
| D | 0010 | | 5856 | 5857 | 5858 | 5859 | 5861 | 5862 | 5863 | 5864 |
| D | 0020 | | 5866 | 5867 | 5868 | 5869 | 5870 | 5871 | 5872 | 5873 |
| D | 0030 | | 3851 | 3852 | 3853 | 3854 | 3855 | 3856 | 3857 | 3858 |
| D | 0040 | | 3862 | 3863 | 3864 | 3865 | 3866 | 3867 | 3868 | 3869 |
| D | 0050 | | 3870 | 3871 | 3872 | 3873 | 3874 | 3875 | 3876 | 3877 |
| D | 0060 | | 3878 | 3879 | 3880 | 3881 | 3882 | 3883 | 3884 | 3885 |
| D | 0070 | | 3886 | 3887 | 3888 | 3889 | 3890 | 3891 | 3892 | 3893 |
| | | | | | | | | | | |
| D | 0100 | | 3897 | 3898 | 3899 | 3900 | 3905 | 3765 | 3906 | 3907 |
| D | 0110 | | 5533 | 5534 | 5622 | 5683 | 4078 | 4079 | 4162 | 4380 |
| D | 0120 | | 2765 | 2766 | 5418 | 5907 | 2774 | 2775 | 5419 | 5383 |
| D | 0130 | | 3908 | 3909 | 5384 | 4681 | 4682 | 4683 | 4684 | 4685 |
| D | 0140 | | 5259 | 3910 | 5260 | 5261 | 5262 | 5263 | 5264 | 5265 |
| D | 0150 | | 5267 | 3911 | 5268 | 5269 | 5270 | 5271 | 5272 | 5273 |
| D | 0160 | | 5304 | 3912 | 5305 | 5306 | 5308 | 5309 | 5310 | 5311 |
| D | 0170 | | 5333 | 3913 | 5334 | 5335 | 5337 | 5338 | 5339 | 5340 |
| | | | | | | | | | | |
| D | 0200 | | 2546 | 2547 | 2548 | 2549 | 2551 | 2552 | 2553 | 2554 |
| D | 0210 | | 2556 | 2557 | 2558 | 2559 | 2561 | 2562 | 2563 | 2564 |
| D | 0220 | | 4111 | 4112 | 4113 | 4114 | 4136 | 4137 | 4138 | 4139 |
| D | 0230 | | 4303 | 4304 | 4305 | 4306 | 4308 | 4309 | 4310 | 4311 |
| D | 0240 | | 2951 | 2952 | 2953 | 2954 | 2955 | 2956 | 2957 | 3914 |
| D | 0250 | | 2583 | 5108 | 3503 | 3504 | 3515 | 3517 | 3646 | 8187 |
| D | 0260 | | 3668 | 3669 | 3670 | 3671 | 3798 | 3799 | 3800 | 3801 |
| D | 0270 | | 4049 | 4050 | 4051 | 4052 | 4063 | 4064 | 4065 | 4066 |
| | | | | | | | | | | |
| D | 0300 | | 3349 | 3350 | 3351 | 3352 | 3353 | 3354 | 3355 | 3356 |
| D | 0310 | | 3358 | 3359 | 3360 | 3361 | 3362 | 3363 | 3364 | 3365 |
| D | 0320 | | 3461 | 3462 | 3463 | 3464 | 3465 | 3466 | 3467 | 3468 |
| D | 0330 | | 3375 | 3376 | 3377 | 3378 | 3379 | 3380 | 3381 | 3382 |
| D | 0340 | | 3475 | 3476 | 3477 | 3478 | 3479 | 3480 | 3481 | 3482 |
| D | 0350 | | 3392 | 3393 | 3394 | 3395 | 3396 | 3397 | 3398 | 3399 |
| D | 0360 | | 3489 | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 |
| D | 0370 | | 3408 | 3409 | 3410 | 3411 | 3412 | 3413 | 3414 | 3415 |
| | | | | | | | | | | |
| D | 0400 | | 2796 | 2797 | 2798 | 2799 | 2806 | 2807 | 2808 | 2809 |
| D | 0410 | | 2816 | 2817 | 2818 | 2819 | 2826 | 2827 | 2828 | 2829 |
| D | 0420 | | 2831 | 2832 | 2833 | 2834 | 2841 | 2842 | 2843 | 2844 |
| D | 0430 | | 2848 | 2849 | 2850 | 2851 | 2858 | 2859 | 2860 | 2861 |
| D | 0440 | | 2868 | 2869 | 2870 | 2871 | 2878 | 2879 | 2880 | 2881 |
| D | 0450 | | 2888 | 2889 | 2890 | 2891 | 2898 | 2899 | 2900 | 2901 |
| D | 0460 | | 2909 | 2910 | 2911 | 2912 | 2919 | 2920 | 2921 | 2922 |
| D | 0470 | | 2929 | 2930 | 2931 | 2932 | 2939 | 2940 | 2941 | 2942 |
| | | | | | | | | | | |
| D | 0500 | | 2599 | 2600 | 2601 | 2602 | 2604 | 2605 | 2606 | 2607 |
| D | 0510 | | 2609 | 2610 | 2611 | 2612 | 2614 | 2615 | 2616 | 2617 |
| D | 0520 | | 2619 | 2620 | 2621 | 2622 | 2624 | 2625 | 2626 | 2627 |
| D | 0530 | | 2629 | 2630 | 2631 | 2632 | 2634 | 2635 | 2636 | 2637 |
| D | 0540 | | 2641 | 2642 | 2643 | 2644 | 2646 | 2647 | 2648 | 2649 |
| D | 0550 | | 2651 | 2652 | 2653 | 2654 | 2656 | 2657 | 2658 | 2659 |
| D | 0560 | | 2661 | 2662 | 2663 | 2664 | 2666 | 2667 | 2668 | 2669 |
| D | 0570 | | 2671 | 2672 | 2673 | 2674 | 2676 | 2677 | 2678 | 2679 |

; T10KI.MCR[10,1141]

11:45 11-AUG-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 309

;
;

| ; Dcode | Loc'n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------|-------|------|------|------|------|------|------|------|------|
| D 0600 | | 3174 | 3175 | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 |
| D 0610 | | 3183 | 3184 | 3185 | 3186 | 3187 | 3188 | 3189 | 3190 |
| D 0620 | | 3192 | 3193 | 3194 | 3195 | 3196 | 3197 | 3198 | 3199 |
| D 0630 | | 3201 | 3202 | 3203 | 3204 | 3205 | 3206 | 3207 | 3208 |
| D 0640 | | 3210 | 3211 | 3212 | 3213 | 3214 | 3215 | 3216 | 3217 |
| D 0650 | | 3219 | 3220 | 3221 | 3222 | 3223 | 3224 | 3225 | 3226 |
| D 0660 | | 3227 | 3228 | 3229 | 3230 | 3231 | 3232 | 3233 | 3234 |
| D 0670 | | 3236 | 3237 | 3238 | 3239 | 3240 | 3241 | 3242 | 3243 |
| | | | | | | | | | |
| D 0700 | | 6995 | 6996 | 7203 | 7736 | 7817 | 7818 | 7737 | 7738 |
| D 0710 | | 7524 | 7525 | 7535 | 7536 | 7549 | 7550 | 7740 | 7741 |
| D 0720 | | 7526 | 7527 | 7537 | 7538 | 7551 | 7552 | 7743 | 7744 |
| D 0730 | | 7746 | 7747 | 7748 | 7749 | 7750 | 7751 | 7752 | 7753 |
| D 0740 | | 7755 | 7756 | 7757 | 7758 | 7759 | 7760 | 7761 | 7762 |
| D 0750 | | 7764 | 7765 | 7766 | 7767 | 7768 | 7769 | 7770 | 7771 |
| D 0760 | | 7773 | 7774 | 7775 | 7776 | 7777 | 7778 | 7779 | 7780 |
| D 0770 | | 7782 | 7783 | 7784 | 7785 | 7786 | 7787 | 7788 | 7789 |

; Location / Line Number Index

| Ucode | Loc'n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| U 0000 | | 2148: | 7911: | 2149 | 2157: | 7852: | 7854: | 5208= | 5211= |
| U 0010 | | 2181= | 7912: | 2182= | 2150 | 3284= | 3287= | 3290= | 3293= |
| U 0020 | | 3630= | 2151 | 3976= | 3979= | 3633= | 3635= | 2152 | 3982= |
| U 0030 | | 3558= | 4120= | 3562= | 4489= | 3566= | 4121= | 3570= | 4490= |
| U 0040 | | 2353: | 2357: | 2361: | 2369: | 2379: | 2382: | 2389: | 2393: |
| U 0050 | | 2406: | 2412: | 2433: | 2155 | 4411= | 4413= | 4447= | 4448= |
| U 0060 | | 3828= | 2159 | 5041= | 5442= | 3831= | 3835= | 5048= | 7580= |
| U 0070 | | 4751= | 2160 | 4755= | 5443= | 4759= | 4415= | 4762= | 7581= |
| U 0100 | | 2162 | 2240= | 2243= | 2246= | 7839= | 2247= | 2165 | 2251= |
| U 0110 | | 2167 | 2258= | 2261= | 2264= | 7841= | 2265= | 7843= | 2270= |
| U 0120 | | 4175= | 4177= | 4268= | 4272= | 4178= | 2169 | 4276= | 4280= |
| U 0130 | | 4194= | 4196= | 5688= | 5690= | 4197= | 2170 | 2172 | 7046: |
| U 0140 | | 2190= | 2193= | 4286= | 4290= | 5367= | 5368= | 4294= | 4298= |
| U 0150 | | 3550= | 2201= | 3551= | 5598= | 5370= | 5372= | 5374= | 5375= |
| U 0160 | | 4331= | 4334= | 2173 | 5322= | 4337= | 4338= | 4340= | 5324= |
| U 0170 | | 5924= | 4148= | 5926= | 5599= | 5927= | 4149= | 5928= | 5376= |
| U 0200 | | 5781= | 2291= | 5783= | 2297= | 5785= | 2303= | 5787= | 2308= |
| U 0210 | | 5789= | 2316= | 2174 | 2320= | 2175 | 2330= | 5790= | 2334= |
| U 0220 | | 3708= | 3710= | 3712= | 2176 | 7913= | 2177 | 7914= | 7616= |
| U 0230 | | 6892= | 6844= | 6893= | 6845= | 6894= | 2178 | 6895= | 7617= |
| U 0240 | | 4690= | 4691= | 2179 | 5669= | 2183 | 4692= | 6533= | 6534= |
| U 0250 | | 3325= | 3328= | 3331= | 3334= | 3337= | 3340= | 3343= | 3346= |
| U 0260 | | 2186= | 2187= | 5522= | 5670= | 4719= | 4722= | 5523= | 4723= |
| U 0270 | | 3428= | 3431= | 3434= | 3437= | 3440= | 3443= | 3446= | 3449= |
| U 0300 | | 3585= | 3589= | 5527= | 3591= | 7348= | 7349= | 5528= | 3592= |
| U 0310 | | 4013= | 2184 | 4015= | 4018= | 4433= | 7351= | 4434= | 3593= |
| U 0320 | | 3605= | 3606= | 5831= | 2217 | 2219 | 3607= | 5833= | 5507= |
| U 0330 | | 4356= | 4358= | 6692= | 4361= | 2278 | 3609= | 6693= | 5509= |
| U 0340 | | 4787= | 4790= | 4791= | 2340 | 4792= | 4793= | 2345 | 4794= |
| U 0350 | | 4729= | 4730= | 4731= | 4732= | 4443= | 4473= | 4444= | 4474= |
| U 0360 | | 4883= | 4884= | 4885= | 2363 | 4886= | 4887= | 2366 | 4888= |
| U 0370 | | 4513= | 2371 | 2230= | 2231= | 4514= | 4515= | 4516= | 4517= |
| U 0400 | | 4035: | 2383 | 2398= | 2400= | 2415= | 2418= | 2385 | 5525= |
| U 0410 | | 4809= | 4705= | 4810= | 4707= | 2434= | 2435= | 2420 | 5526= |
| U 0420 | | 5480= | 5481= | 5482= | 5483= | 5484= | 5485= | 5486= | 5487= |
| U 0430 | | 5488= | 2421 | 2424 | 5581= | 2456= | 2457= | 5489= | 5582= |
| U 0440 | | 5500= | 6320= | 5501= | 6322= | 5502= | 2425 | 5503= | 6323= |
| U 0450 | | 5504= | 2427 | 2430 | 5615= | 2538= | 2541= | 5505= | 5616= |
| U 0460 | | 6177= | 6179= | 5835= | 5839= | 6181= | 6183= | 5842= | 2591 |
| U 0470 | | 7471= | 2721 | 5605= | 5607= | 7473= | 7475= | 5608= | 5609= |
| U 0500 | | 5992= | 5994= | 2990= | 2994= | 5996= | 5998= | 2727 | 6000= |
| U 0510 | | 6378= | 2786 | 2791 | 5728= | 3020= | 3021= | 6379= | 5729= |
| U 0520 | | 5742= | 5743= | 5745= | 5746= | 5748= | 5749= | 5751= | 5752= |
| U 0530 | | 5754= | 2906 | 2972 | 6817= | 3039= | 3040= | 5755= | 6819= |
| U 0540 | | 6073= | 6075= | 6077= | 4184= | 3070= | 3071= | 6080= | 4185= |
| U 0550 | | 5012= | 5014= | 2985 | 5018= | 3074= | 3075= | 5935= | 5936= |
| U 0560 | | 6211= | 6214= | 6370= | 4204= | 2988 | 6215= | 6371= | 4206= |
| U 0570 | | 5213= | 5215= | 3004 | 5216= | 5971= | 5972= | 5974= | 5977= |

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| U 0600 | 6217= | 6220= | 3087= | 3088= | 3006 | 6221= | 6577= | 6578= |
| U 0610 | 5233= | 5235= | 3007 | 5236= | 3094= | 3096= | 6169= | 6172= |
| U 0620 | 6228= | 6229= | 6233= | 4227= | 3100= | 3102= | 6234= | 4229= |
| U 0630 | 5793= | 3009 | 5795= | 5797= | 3107= | 3109= | 6376= | 6377= |
| U 0640 | 6357= | 6359= | 3113= | 3115= | 3142= | 3143= | 3012 | 6360= |
| U 0650 | 6021= | 6023= | 3023 | 6025= | 6461= | 6462= | 6463= | 6464= |
| U 0660 | 4970= | 6476= | 4971= | 6479= | 4980= | 6481= | 4981= | 6483= |
| U 0670 | 5201= | 6485= | 5202= | 6488= | 5284= | 6491= | 5285= | 6493= |
| U 0700 | 6554= | 3041 | 6558= | 6560= | 6561= | 6564= | 3042 | 6566= |
| U 0710 | 6501= | 6503= | 6505= | 6508= | 6509= | 6511= | 6730= | 6731= |
| U 0720 | 5426= | 6768= | 5427= | 6770= | 5428= | 6774= | 5429= | 6778= |
| U 0730 | 5638= | 6782= | 5643= | 6786= | 5642= | 6790= | 5644= | 6794= |
| U 0740 | 6599= | 6601= | 3146= | 3148= | 3388= | 3389= | 6603= | 3068 |
| U 0750 | 6039= | 6041= | 6043= | 6044= | 6807= | 6810= | 6812= | 6814= |
| U 0760 | 6624= | 6625= | 3452= | 3453= | 3456= | 3457= | 6626= | 3076 |
| U 0770 | 7419= | 7420= | 7421= | 7422= | 7423= | 7424= | 7425= | 7426= |
| U 1000 | 6083= | 6084= | 3077 | 6085= | 6116= | 6118= | 6119= | 3091 |
| U 1010 | 3577= | 3578= | 5655= | 7561= | 3582= | 3583= | 5656= | 7563= |
| U 1020 | 6125= | 6127= | 6128= | 3111 | 3626= | 3627= | 3116 | 7589= |
| U 1030 | 6192= | 6194= | 6195= | 3140 | 3639= | 3640= | 3298 | 7590= |
| U 1040 | 6726= | 6727= | 6728= | 3299 | 7280= | 7283= | 3404 | 7285= |
| U 1050 | 7302= | 7304= | 7305= | 3405 | 5716= | 7658= | 5717= | 7660= |
| U 1060 | 8216= | 8218= | 5662= | 8222= | 3553 | 8224= | 5667= | 8226= |
| U 1070 | 8228= | 8230= | 8232= | 8234= | 7686= | 7689= | 7692= | 7694= |
| U 1100 | 8765= | 8767= | 8769= | 8771= | 8773= | 8775= | 8777= | 8779= |
| U 1110 | 8781= | 8782= | 8783= | 8785= | 8737= | 8738= | 8739= | 3574 |
| U 1120 | 3656= | 3660= | 3696= | 3700= | 3732= | 3736= | 3757= | 3760= |
| U 1130 | 3778= | 3783= | 3786= | 3788= | 3792= | 3793= | 3945= | 3955= |
| U 1140 | 3990= | 3992= | 3995= | 3999= | 4089= | 4090= | 4099= | 4101= |
| U 1150 | 4104= | 4105= | 4122= | 4123= | 4128= | 4129= | 4131= | 4132= |
| U 1160 | 4152= | 4153= | 4154= | 4156= | 4209= | 4210= | 4217= | 4218= |
| U 1170 | 4319= | 4321= | 4346= | 4349= | 4367= | 4371= | 4373= | 4375= |
| U 1200 | 4389= | 4391= | 4393= | 4394= | 4403= | 4404= | 4419= | 4420= |
| U 1210 | 4422= | 4423= | 4426= | 4427= | 4428= | 4429= | 4437= | 4438= |
| U 1220 | 4453= | 4454= | 4463= | 4464= | 4481= | 4482= | 4532= | 4533= |
| U 1230 | 4534= | 4535= | 4541= | 4542= | 4551= | 4552= | 4555= | 4556= |
| U 1240 | 4580= | 4583= | 4597= | 4601= | 4613= | 4614= | 4618= | 4619= |
| U 1250 | 4623= | 4624= | 4639= | 4641= | 4645= | 4646= | 4941= | 4944= |
| U 1260 | 4960= | 4962= | 4965= | 4966= | 4996= | 4997= | 5031= | 5033= |
| U 1270 | 5060= | 5063= | 5096= | 5098= | 5124= | 5131= | 5135= | 5138= |
| U 1300 | 5140= | 5142= | 5147= | 5152= | 5164= | 5165= | 5249= | 5250= |
| U 1310 | 5289= | 5290= | 5293= | 5294= | 5298= | 5299= | 5318= | 5319= |
| U 1320 | 5348= | 5349= | 5351= | 5352= | 5357= | 5358= | 5360= | 5361= |
| U 1330 | 5364= | 5365= | 5389= | 5390= | 5393= | 5394= | 5396= | 5397= |
| U 1340 | 5404= | 5405= | 5412= | 5413= | 5434= | 5435= | 5438= | 5439= |
| U 1350 | 5445= | 5446= | 5447= | 5449= | 5492= | 5493= | 5496= | 5498= |
| U 1360 | 5511= | 5512= | 5513= | 5514= | 5551= | 5553= | 5556= | 5558= |
| U 1370 | 5562= | 5563= | 5570= | 5571= | 5612= | 5613= | 5630= | 5631= |

| Ucode | Loc'n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| U 1400 | | 2227: | 2228: | 2568: | 2571: | 2574: | 2578: | 2588: | 2688: |
| U 1410 | | 2690: | 2715: | 2717: | 2720: | 2723: | 2726: | 2729: | 2735: |
| U 1420 | | 2737: | 2739: | 2742: | 2749: | 2744: | 2746: | 2751: | 2753: |
| U 1430 | | 2756: | 3403: | 2758: | 2760: | 2770: | 2770: | 2771: | 3419: |
| U 1440 | | 3472: | 2803: | 2813: | 2823: | 2838: | 2855: | 2865: | 2875: |
| U 1450 | | 2885: | 2895: | 2905: | 2916: | 2926: | 2936: | 2946: | 4084: |
| U 1460 | | 7542: | 7546: | 3016: | 7543: | 3066: | 3067: | 3085: | 5911: |
| U 1470 | | 3138: | 3139: | 3268: | 3270: | 3273: | 3276: | 3369: | 3387: |
| U 1500 | | | | | | 2484: | 2489: | 2495: | |
| U 1510 | | | | | | 2516: | 2521: | 2527: | 2533: |
| U 1520 | | 3524: | 3525: | 3527: | 3528: | 3529: | 3532: | 3533: | 3534: |
| U 1530 | | 3536: | 3537: | 3539: | 3540: | 3541: | 3542: | 3543: | 3544: |
| U 1540 | | 3600: | 3650: | 3600: | 3677: | 3690: | 3719: | 3752: | 3510: |
| U 1550 | | 3805: | 3770: | 3812: | 8193: | 3825: | 3839: | 3940: | 4030: |
| U 1560 | | 4057: | 4071: | | | | 2780: | 4167: | 2785: |
| U 1570 | | 5317: | 4144: | | | 5345: | | 5279: | 5282: |
| U 1600 | | 4315: | 4324: | | | | | | |
| U 1610 | | 4688: | 3486: | 2966: | 2969: | 7531: | 4096: | 5388: | 7532: |
| U 1620 | | 4696: | 5409: | 2980: | 2982: | 4700: | 4701: | 5425: | 4385: |
| U 1630 | | 4710: | 5627: | 3001: | 3003: | 4712: | 5543: | 5686: | 5546: |
| U 1640 | | 5112: | 4117: | 5114: | 5117: | 7557: | | | 7559: |
| U 1650 | | 7793: | 7799: | 7802: | 7804: | 7806: | 7808: | 7810: | 7812: |
| U 1660 | | 3930: | 3918: | 3920: | 3922: | 3924: | 3932: | 3926: | 3928: |
| U 1670 | | | | | | | | | |
| U 1700 | | 7036: | 6999: | 7000: | 7001: | 7049: | 7083: | 7002: | 7003: |
| U 1710 | | 7005: | 7006: | 7007: | 7008: | 7331: | 7328: | 7009: | 7010: |
| U 1720 | | 7013: | 7186: | 7376: | 7102: | 7146: | 7177: | 7014: | 7015: |
| U 1730 | | 7018: | 7019: | 7020: | 7021: | 7022: | 7023: | 7024: | 7025: |
| U 1740 | | 5879: | 5881: | 5883: | 5885: | 5887: | | 5889: | 5891: |
| U 1750 | | 5893: | 5895: | | | 7824: | 7832: | | |
| U 1760 | | 7208: | 7210: | 7212: | 7214: | 7291: | 7323: | 7216: | 7217: |
| U 1770 | | 7220: | 7224: | 7228: | 7232: | 7275: | 7315: | 7236: | 7239: |
| U 2000 | | 5652= | 5653= | 5673= | 5674= | 5676= | 5678= | 5693= | 5694= |
| U 2010 | | 5703= | 5704= | 5712= | 5713= | 5714= | 5715= | 5731= | 5732= |
| U 2020 | | 5720= | 5982= | 5721= | 5983= | 5764= | 5765= | 5769= | 5770= |
| U 2030 | | 5774= | 5776= | 5801= | 5802= | 5809= | 5810= | 5819= | 5820= |
| U 2040 | | 5913= | 5956= | 5938= | 5940= | 5915= | 5957= | 5960= | 5963= |
| U 2050 | | 5965= | 5966= | 6008= | 6010= | 6014= | 6015= | 6029= | 6031= |
| U 2060 | | 6034= | 6036= | 6047= | 6049= | 6058= | 6161= | 6059= | 6162= |
| U 2070 | | 6094= | 6096= | 6098= | 6099= | 6095= | 6097= | 6102= | 6104= |
| U 2100 | | 6112= | 6113= | 6139= | 6140= | 6146= | 6149= | 6164= | 6167= |
| U 2110 | | 6185= | 6189= | 6199= | 6200= | 6203= | 6205= | 6239= | 6240= |
| U 2120 | | 6252= | 6253= | 6256= | 6257= | 6260= | 6261= | 6266= | 6268= |
| U 2130 | | 6269= | 6271= | 6279= | 6280= | 6290= | 6291= | 6300= | 6302= |
| U 2140 | | 6295= | 6304= | 6297= | 6306= | 6308= | 6309= | 6314= | 6317= |
| U 2150 | | 6327= | 6328= | 6332= | 6333= | 6338= | 6339= | 6341= | 6345= |
| U 2160 | | 6383= | 6384= | 6388= | 6389= | 6395= | 6396= | 6397= | 6398= |
| U 2170 | | 6399= | 6400= | 6417= | 6418= | 6421= | 6551= | 6423= | 6552= |

| ; T1OKI.MCR[10,1141] | | 11:45 11-AUG-1984 | | MICRO 31(254) | | KS10 MICROCODE V124, 27-JUL-84 Page 313 | | | |
|----------------------|--|-------------------|-------|---------------|-------|---|-------|-------|-------|
| ; Ucode Loc'n | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| U 2200 | | 6424= | 6425= | 6431= | 6432= | 6437= | 6439= | 6458= | 6459= |
| U 2210 | | 6453= | 3579 | 6466= | 6467= | 6454= | 3612 | 6472= | 6473= |
| U 2220 | | 6499= | 6500= | 6524= | 6527= | 6541= | 6542= | 6571= | 6572= |
| U 2230 | | 6592= | 6594= | 6616= | 6618= | 6620= | 6621= | 6638= | 6640= |
| U 2240 | | 6661= | 6665= | 6667= | 6669= | 6696= | 6697= | 6750= | 6751= |
| U 2250 | | 6755= | 6757= | 6801= | 6803= | 6823= | 6825= | 6840= | 6841= |
| U 2260 | | 6850= | 6854= | 6869= | 6872= | 6874= | 6886= | 6875= | 6887= |
| U 2270 | | 6926= | 6927= | 6934= | 6935= | 6942= | 6943= | 6945= | 6946= |
| U 2300 | | 6972= | 6980= | 6989= | 6990= | 7057= | 7058= | 7059= | 7060= |
| U 2310 | | 7062= | 7063= | 7064= | 7065= | 7067= | 7068= | 7111= | 7114= |
| U 2320 | | 7121= | 7126= | 7133= | 7136= | 7148= | 7151= | 7154= | 7155= |
| U 2330 | | 7179= | 7180= | 7187= | 7190= | 7196= | 7197= | 7253= | 7254= |
| U 2340 | | 7256= | 7930= | 7258= | 7931= | 7264= | 7265= | 7296= | 7297= |
| U 2350 | | 7332= | 7333= | 7334= | 7335= | 7338= | 7339= | 7340= | 7341= |
| U 2360 | | 7342= | 7343= | 7344= | 7345= | 7346= | 7347= | 7379= | 7381= |
| U 2370 | | 7387= | 7388= | 7392= | 7394= | 7400= | 7402= | 7441= | 7444= |
| U 2400 | | 7445= | 7446= | 7457= | 7458= | 7468= | 7469= | 7485= | 7487= |
| U 2410 | | 7489= | 7491= | 7493= | 7495= | 7499= | 7500= | 7514= | 7515= |
| U 2420 | | 7518= | 7519= | 7576= | 7577= | 7585= | 7588= | 7595= | 7597= |
| U 2430 | | 7601= | 7603= | 7614= | 7615= | 7621= | 7622= | 7627= | 7629= |
| U 2440 | | 7633= | 7635= | 7646= | 7647= | 7650= | 7652= | 7655= | 7656= |
| U 2450 | | 7665= | 7666= | 7672= | 7673= | 7700= | 7701= | 7706= | 7707= |
| U 2460 | | 7847= | 7848= | 7859= | 7860= | 7863= | 7867= | 7920= | 7922= |
| U 2470 | | 7933= | 7935= | 7934= | 7936= | 7937= | 7940= | 7938= | 7941= |
| U 2500 | | 7943= | 7946= | 7944= | 7947= | 7949= | 7952= | 7950= | 7953= |
| U 2510 | | 7954= | 3615 | 7955= | 3617 | 8200= | 8201= | 8251= | 8254= |
| U 2520 | | 8265= | 8267= | 8497= | 8499= | 8503= | 8506= | 8513= | 8515= |
| U 2530 | | 8531= | 8533= | 8536= | 8537= | 8654= | 8659= | 8664= | 8667= |
| U 2540 | | 8674= | 8677= | 8687= | 8689= | 8693= | 8696= | 8699= | 8700= |
| U 2550 | | 8704= | 8706= | 8710= | 8712= | 8716= | 8717= | 8726= | 8729= |
| U 2560 | | 8749= | 8751= | 8753= | 8754= | 8757= | 8758= | 8760= | 3619 |
| U 2570 | | 8791= | 8792= | 8797= | 8808= | 8817= | 8818= | 3621 | 3641 |
| U 2600 | | 3663 | 3685 | 3703 | 3722 | 3727 | 3741 | 3746 | 3756 |
| U 2610 | | 3773 | 3808 | 3814 | 3816 | 3820 | 3826 | 3840 | 3842 |
| U 2620 | | 3845 | 3942 | 3984 | 3985 | 3987 | 4001 | 4006 | 4023 |
| U 2630 | | 4026 | 4038 | 4040 | 4044 | 4092 | 4103 | 4118 | 4126 |
| U 2640 | | 4145 | 4146 | 4150 | 4168 | 4170 | 4187 | 4188 | 4199 |
| U 2650 | | 4208 | 4211 | 4212 | 4213 | 4214 | 4216 | 4222 | 4223 |
| U 2660 | | 4224 | 4231 | 4247 | 4250 | 4262 | 4317 | 4325 | 4326 |
| U 2670 | | 4328 | 4351 | 4353 | 4387 | 4396 | 4397 | 4400 | 4405 |
| U 2700 | | 4406 | 4408 | 4432 | 4435 | 4436 | 4439 | 4445 | 4451 |
| U 2710 | | 4452 | 4456 | 4458 | 4459 | 4461 | 4466 | 4467 | 4469 |
| U 2720 | | 4470 | 4476 | 4479 | 4484 | 4492 | 4531 | 4537 | 4538 |
| U 2730 | | 4544 | 4545 | 4546 | 4547 | 4548 | 4553 | 4557 | 4558 |
| U 2740 | | 4561 | 4591 | 4609 | 4611 | 4616 | 4621 | 4626 | 4627 |
| U 2750 | | 4628 | 4635 | 4637 | 4644 | 4715 | 4716 | 4735 | 4736 |
| U 2760 | | 4744 | 4746 | 4767 | 4771 | 4804 | 4813 | 4816 | 4817 |
| U 2770 | | 4818 | 4821 | 4822 | 4823 | 4890 | 4891 | 4894 | 4897 |

; Ucode Loc'n 0 1 2 3 4 5 6 7

Location / Line Number Index

| | | | | | | | | |
|--------|------|------|------|------|------|------|-------|-------|
| U 3000 | 4898 | 4899 | 4900 | 4902 | 4905 | 4908 | 4909 | 4910 |
| U 3010 | 4912 | 4937 | 4947 | 4948 | 4951 | 4953 | 4955 | 4974 |
| U 3020 | 4975 | 4976 | 4977 | 4983 | 4991 | 5021 | 5035 | 5038 |
| U 3030 | 5039 | 5051 | 5053 | 5056 | 5065 | 5092 | 5099 | 5100 |
| U 3040 | 5102 | 5104 | 5105 | 5119 | 5134 | 5157 | 5159 | 5162 |
| U 3050 | 5170 | 5174 | 5176 | 5178 | 5206 | 5219 | 5221 | 5222 |
| U 3060 | 5223 | 5225 | 5227 | 5228 | 5229 | 5231 | 5240 | 5242 |
| U 3070 | 5243 | 5246 | 5247 | 5248 | 5253 | 5286 | 5287 | 5288 |
| U 3100 | 5296 | 5320 | 5325 | 5328 | 5353 | 5354 | 5355 | 5378 |
| U 3110 | 5401 | 5410 | 5411 | 5431 | 5515 | 5544 | 5549 | 5564 |
| U 3120 | 5568 | 5569 | 5572 | 5577 | 5579 | 5584 | 5586 | 5587 |
| U 3130 | 5600 | 5602 | 5617 | 5632 | 5633 | 5637 | 5645 | 5651 |
| U 3140 | 5657 | 5671 | 5672 | 5675 | 5695 | 5699 | 5702 | 5705 |
| U 3150 | 5706 | 5709 | 5724 | 5758 | 5761 | 5771 | 5799 | 5804 |
| U 3160 | 5806 | 5812 | 5917 | 5919 | 5921 | 5922 | 5930 | 5968 |
| U 3170 | 5979 | 5980 | 5985 | 6006 | 6012 | 6013 | 6016 | 6028 |
| U 3200 | 6045 | 6052 | 6055 | 6056 | 6057 | 6060 | 6061 | 6063 |
| U 3210 | 6069 | 6070 | 6087 | 6088 | 6114 | 6121 | 6122 | 6123 |
| U 3220 | 6131 | 6132 | 6133 | 6134 | 6135 | 6136 | 6141 | 6142 |
| U 3230 | 6155 | 6157 | 6159 | 6173 | 6198 | 6201 | 6224 | 6242 |
| U 3240 | 6245 | 6247 | 6255 | 6258 | 6263 | 6265 | 6272 | 6281 |
| U 3250 | 6292 | 6293 | 6312 | 6334 | 6336 | 6346 | 6347 | 6349 |
| U 3260 | 6350 | 6351 | 6352 | 6353 | 6356 | 6362 | 6367 | 6374 |
| U 3270 | 6380 | 6381 | 6382 | 6385 | 6386 | 6387 | 6402 | 6406 |
| U 3300 | 6426 | 6428 | 6433 | 6434 | 6455 | 6456 | 6470 | 6519 |
| U 3310 | 6520 | 6522 | 6544 | 6545 | 6581 | 6587 | 6588 | 6589 |
| U 3320 | 6590 | 6608 | 6629 | 6631 | 6641 | 6644 | 6649 | 6650 |
| U 3330 | 6651 | 6652 | 6653 | 6655 | 6658 | 6670 | 6673 | 6675 |
| U 3340 | 6677 | 6678 | 6695 | 6736 | 6739 | 6744 | 6746 | 6821 |
| U 3350 | 6843 | 6846 | 6848 | 6867 | 6885 | 6888 | 6890 | 6897 |
| U 3360 | 6898 | 6912 | 6914 | 6919 | 6929 | 6930 | 6938 | 6940 |
| U 3370 | 6948 | 6950 | 6952 | 6953 | 6954 | 6956 | 6959 | 6963 |
| U 3400 | 6965 | 6983 | 6986 | 7027 | 7028 | 7030 | 7031 | 7051 |
| U 3410 | 7052 | 7053 | 7055 | 7056 | 7061 | 7070 | 7072 | 7075 |
| U 3420 | 7077 | 7080 | 7085 | 7088 | 7091 | 7092 | 7094 | 7096 |
| U 3430 | 7105 | 7117 | 7129 | 7138 | 7139 | 7143 | 7152 | 7153 |
| U 3440 | 7156 | 7169 | 7183 | 7193 | 7194 | 7198 | 7221 | 7222 |
| U 3450 | 7225 | 7226 | 7229 | 7230 | 7233 | 7234 | 7237 | 7238 |
| U 3460 | 7246 | 7248 | 7249 | 7250 | 7259 | 7261 | 7266 | 7267 |
| U 3470 | 7270 | 7277 | 7288 | 7292 | 7293 | 7295 | 7299 | 7307 |
| U 3500 | 7308 | 7309 | 7316 | 7317 | 7319 | 7336 | 7337 | 7359 |
| U 3510 | 7360 | 7361 | 7363 | 7369 | 7370 | 7377 | 7382 | 7395 |
| U 3520 | 7404 | 7407 | 7427 | 7428 | 7429 | 7430 | 7431 | 7432 |
| U 3530 | 7433 | 7436 | 7437 | 7438 | 7448 | 7450 | 7453 | 7455 |
| U 3540 | 7456 | 7460 | 7483 | 7497 | 7501 | 7504 | 7517 | 7566 |
| U 3550 | 7619 | 7648 | 7663 | 7664 | 7668 | 7675 | 6907: | 6908: |
| U 3560 | 7682 | 7697 | 7713 | 7715 | 7717 | 7719 | 7721 | 7723 |
| U 3570 | 7725 | 7727 | 7729 | 7731 | 7827 | 7834 | 7845 | 7918 |

| | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|
| U 3600 | 7924 | 7926 | 7929 | 7932 | 7939 | 7942 | 7945 | 7948 |
| U 3610 | 7951 | 7956 | 7957 | 7958 | 7960 | 8194 | 8196 | 8197 |
| U 3620 | 8198 | 8199 | 8208 | 8209 | 8210 | 8212 | 8238 | 8239 |
| U 3630 | 8240 | 8244 | 8247 | 8248 | 8259 | 8261 | 8280 | 8492 |
| U 3640 | 8507 | 8510 | 8511 | 8517 | 8519 | 8524 | 8526 | 8528 |
| U 3650 | 8529 | 8530 | 8538 | 8539 | 8540 | 8543 | 8547 | 8650 |
| U 3660 | 8679 | 8681 | 8697 | 8721 | 8724 | 8731 | 8733 | 8742 |
| U 3670 | 8745 | 8755 | 8794 | 8811 | 8815 | 8831 | 8833 | |

U 3700 - 3737 Unused
U 3740 6093: 6450: 6154: 6287: 5953:
U 3750 6451: 5954:
U 3760
U 3770 8207:

No errors detected
End of microcode assembly
318 pages of listing
Used 47.06 seconds, 116 pages of core
Symbol table: 30P
Text strings: 9P
Loc'n assignment: 18P
Cross reference: 51P

