

; 1	T10KL.MIC[10,1141]	13:19 26-JULY-1984
; 14	KS10.MIC[10,1141]	07:10 27-JULY-1984
; 69	REVISION HISTORY	
; 188	HOW TO READ THE MICROCODE	
; 393	CONDITIONAL ASSEMBLY DEFINITIONS	
; 439	2901 REGISTER USAGE	
; 475	MICROCODE FIELDS -- LISTING FORMAT	
; 523	MICROCODE FIELDS -- DATAPATH CHIP	
; 675	MICROCODE FIELDS -- RAM FILE ADDRESS AND D-BUS	
; 709	MICROCODE FIELDS -- PARITY GENERATION & HALF WORD CONTROL	
; 732	MICROCODE FIELDS -- SPEC	
; 835	MICROCODE FIELDS -- DISPATCH	
; 879	MICROCODE FIELDS -- SKIP	
; 930	MICROCODE FIELDS -- TIME CONTROL	
; 950	MICROCODE FIELDS -- RANDOM CONTROL BITS	
; 972	MICROCODE FIELDS -- NUMBER FIELD	
; 1310	DISPATCH ROM DEFINITIONS	
; 1356	HOW TO READ MACROS	
; 1515	MACROS -- DATA PATH CHIP -- GENERAL	
; 1665	MACROS -- DATA PATH CHIP -- Q	
; 1700	MACROS -- DATA PATH CHIP -- MISC.	
; 1721	MACROS -- STORE IN AC	
; 1753	MACROS -- MICROCODE WORK SPACE	
; 1780	MACROS -- MEMORY CONTROL	
; 1830	MACROS -- VMA	
; 1847	MACROS -- TIME CONTROL	
; 1860	MACROS -- SCAD, SC, FE LOGIC	
; 1943	MACROS -- DATA PATH FIELD CONTROL	
; 1959	MACROS -- SHIFT PATH CONTROL	
; 1972	MACROS -- SPECIAL FUNCTIONS	
; 2003	MACROS -- PC FLAGS	
; 2032	MACROS -- PAGE FAIL FLAGS	
; 2040	MACROS -- SINGLE SKIPS	
; 2065	MACROS -- SPECIAL DISPATCH MACROS	
; 2099	DISPATCH ROM MACROS	
; 2140	SIMPLE.MIC[10,1141]	15:31 27-JULY-1984
; 2142	POWER UP SEQUENCE	
; 2224	THE INSTRUCTION LOOP -- START NEXT INSTRUCTION	
; 2348	THE INSTRUCTION LOOP -- FETCH ARGUMENTS	
; 2460	THE INSTRUCTION LOOP -- STORE ANSWERS	
; 2544	MOVE GROUP	
; 2581	EXCH	
; 2596	HALFWORD GROUP	
; 2763	DMOVE, DMOVN, DMOVEM, DMOVNM	
; 2794	BOOLEAN GROUP	
; 2949	ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO	
; 3045	ROTATES AND LOGICAL SHIFTS -- LSHC	
; 3080	ROTATES AND LOGICAL SHIFTS -- ASHC	
; 3119	ROTATES AND LOGICAL SHIFTS -- ROTC	
; 3151	TEST GROUP	
; 3303	COMPARE -- CAI, CAM	
; 3372	ARITHMETIC SKIPS -- AOS, SOS, SKIP	
; 3422	CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ	
; 3513	AC DECODE JUMPS -- JRST, JFCL	
; 3603	EXTENDED ADDRESSING INSTRUCTIONS	

; 3644 XCT
; 3666 STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ
; 3763 STACK INSTRUCTIONS -- ADJSP
; 3796 SUBROUTINE CALL/RETURN -- JSR, JSP, JSA, JRA
; 3848 ILLEGAL INSTRUCTIONS AND UOO'S
; 4047 ARITHMETIC -- ADD, SUB
; 4076 ARITHMETIC -- DADD, DSUB
; 4109 ARITHMETIC -- MUL, IMUL
; 4160 ARITHMETIC -- DMUL
; 4301 ARITHMETIC -- DIV, IDIV
; 4378 ARITHMETIC -- DDIV
; 4499 ARITHMETIC -- DIVIDE SUBROUTINE
; 4564 ARITHMETIC -- DOUBLE DIVIDE SUBROUTINE
; 4604 ARITHMETIC -- SUBROUTINES FOR ARITHMETIC
; 4650 BYTE GROUP -- IBP, ILDB, LDB, IDPB, DPB
; 4727 BYTE GROUP -- INCREMENT BYTE POINTER SUBROUTINE
; 4740 BYTE GROUP -- BYTE EFFECTIVE ADDRESS EVALUATOR
; 4774 BYTE GROUP -- LOAD BYTE SUBROUTINE
; 4827 BYTE GROUP -- DEPOSIT BYTE IN MEMORY
; 4915 BYTE GROUP -- ADJUST BYTE POINTER
; 5074 BLT
; 5182 UBABLT - BLT BYTES TO/FROM UNIBUS FORMAT
; 5256 FLT.MIC[10,1141] 01:46 20-MAR-1981
; 5257 FLOATING POINT -- FAD, FSB
; 5302 FLOATING POINT -- FMP
; 5331 FLOATING POINT -- FDV
; 5381 FLOATING POINT -- FLTR, FSC
; 5416 FLOATING POINT -- FIX AND FIXR
; 5453 FLOATING POINT -- SINGLE PRECISION NORMALIZE
; 5520 FLOATING POINT -- ROUND ANSWER
; 5531 FLOATING POINT -- DFAD, DFSB
; 5620 FLOATING POINT -- DFMP
; 5681 FLOATING POINT -- DFDV
; 5735 FLOATING POINT -- DOUBLE PRECISION NORMALIZE
; 5845 EXTEND.MIC[10,1141] 11:35 26-JULY-1984
; 5846 EXTEND -- DISPATCH ROM ENTRIES
; 5901 EXTEND -- INSTRUCTION SET DECODING
; 5943 EXTEND -- MOVE STRING -- SETUP
; 5988 EXTEND -- MOVE STRING -- OFFSET/TRANSLATE
; 6019 EXTEND -- MOVE STRING -- MOVSRJ
; 6067 EXTEND -- MOVE STRING -- SIMPLE MOVE LOOP
; 6091 EXTEND -- COMPARE STRING
; 6152 EXTEND -- DECIMAL TO BINARY CONVERSION
; 6284 EXTEND -- BINARY TO DECIMAL CONVERSION
; 6442 EXTEND -- EDIT -- MAIN LOOP
; 6496 EXTEND -- EDIT -- DECODE OPERATE GROUP
; 6515 EXTEND -- EDIT -- STOP EDIT
; 6530 EXTEND -- EDIT -- START SIGNIFICANCE
; 6537 EXTEND -- EDIT -- EXCHANGE MARK AND DESTINATION
; 6548 EXTEND -- EDIT -- PROCESS SOURCE BYTE
; 6611 EXTEND -- EDIT -- MESSAGE BYTE
; 6634 EXTEND -- EDIT -- SKIP
; 6648 EXTEND -- EDIT -- ADVANCE PATTERN POINTER
; 6681 EXTEND SUBROUTINES -- FILL OUT DESTINATION
; 6705 EXTEND SUBROUTINES -- GET MODIFIED SOURCE BYTE

; 6742 EXTEND SUBROUTINES -- TRANSLATE
; 6828 EXTEND SUBROUTINES -- GET UNMODIFIED SOURCE BYTE
; 6857 EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING
; 6878 EXTEND SUBROUTINES -- UPDATE DEST STRING POINTERS
; 6922 EXTEND -- PAGE FAIL CLEANUP
; 6961 INOUT.MIC[10,1141] 09:17 27-JULY-1984
; 6962 TRAPS
; 6993 IO -- INTERNAL DEVICES
; 7099 IO -- INTERNAL DEVICES -- EBR & UBR
; 7201 IO -- INTERNAL DEVICES -- KL PAGING REGISTERS
; 7242 IO -- INTERNAL DEVICES -- TIMER CONTROL
; 7273 IO -- INTERNAL DEVICES -- WRTIME & RDTIME
; 7312 IO -- INTERNAL DEVICES -- WRINT & RDINT
; 7326 IO -- INTERNAL DEVICES -- RDPI & WRPI
; 7366 IO -- INTERNAL DEVICES -- SUBROUTINES
; 7507 PRIORITY INTERRUPTS -- DISMISS SUBROUTINE
; 7522 EXTERNAL IO INSTRUCTIONS
; 7710 SMALL SUBROUTINES
; 7734 UNDEFINED IO INSTRUCTIONS
; 7815 UMOVE AND UMOVEM
; 7870 WRITE HALT STATUS BLOCK
; 7962 PAGEF.MIC[10,1141] 12:32 26-JULY-1984
; 7964 PAGE FAIL REFIL LOGIC
; Cross Reference Index
; DCODE Location / Line Number Index
; UCODE Location / Line Number Index

; 1 .NOBIN
; 2 ;KS10 MICROCODE PARAMETER FILE
; 3
; 4 ;PARAMETER FILE DEFINITIONS FOR T1OKL
; 5 ;TOPS-10 KL PAGING MICROCODE (V7.03)
; 6
; 7 .SET/INHCST=1 ;ALLOW INHIBIT OF CST UPDATE IF CSB = 0
; 8 .SET/UBABLT=1 ;SUPPORT UBABLT INSTRUCTIONS
; 9 .SET/KIPAGE=0 ;REMOVE KI PAGING (REQUIRED FOR UBABLT)
; 10 .SET/NONSTD=1 ;FOR NOW, CALL THIS VERSION NON-STANDARD
; 11
; 12

; 13 .BIN
; 14

```
15 .NOBIN
16 .TITLE "KS10 MICROCODE V124, 27-JUL-84"
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
```

COPYRIGHT (C) 1976, 1977, 1978, 1979, 1980, 1981, 1982,
1984
DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE
ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED
ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT
NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF,
MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO
ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND
TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO
AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS
A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION:

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
RELIABILITY OF ITS SOFTWARE IN EQUIPMENT WHICH IS
NOT SUPPLIED BY DEC.

DESIGNED AND WRITTEN BY:
DONALD A. LEWINE
DIGITAL EQUIPMENT CORP.
MARLBORO, MASS.
MR1-2/E47 X6430

MAINTAINED AND ENHANCED BY:
DONALD D. DOSSA
DIGITAL EQUIPMENT CORP.
MARLBORO, MASS.
MR1-2/E18 DTN 231-4138

SEAN KEENAN
DIGITAL EQUIPMENT CORP.
MARLBORO, MASS.
MR1-2/E18 DTN 231-4463

TIM LITT
PK01/C-2 223-6334

Produced on Advanced Information Services Electronic Laser Printer, PK01/ES, DTN: 223-7881

; 69 .TOC "REVISION HISTORY"
; 70
; 71 ;REV WHY
; 72 ;1 START KS10 MICROCODE BASED ON SM10 MICROCODE VERSION 510
; 73 ;2 UPDATE TO KS10 VERSION 512
; 74 ;3 FIX SOME DEFAULTS
; 75 ;4 CHANGE HARDWARE TO MATCH ECO #215
; 76 ;5 START TO UPDATE IO MICROCODE
; 77 ;6 MORE WORK ON IO
; 78 ;7 MAKE INTERRUPT THE 8080 BE A PULSE.
; 79 ;10 ADD NEW RDIO AND WRIO
; 80 ;11 FIX PROBLEMS IN MUUD CODE & CORRECT T-FIELDS
; 81 ;12 FIX PROBLEMS IN DDIV
; 82 ;13 FIX UP PROBLEMS IN PI
; 83 ;14 TURN ON WRITE FOR FL-EXIT
; 84 ;15 FIX UP MAP INSTRUCTION
; 85 ;16 MORE WORK ON KI-STYLE MAP
; 86 ;17 INVERT HOLD RIGHT AND HOLD LEFT BITS
; 87 ;20 FIXUP WRIO & RDIO EFFECTIVE ADDRESS CALC.
; 88 ;21 FIX EDIT 15
; 89 ;22 HAVE LSH USE FAST SHIFT HARDWARE
; 90 ;23 FIX T-FIELD VALUES FOR PRODUCTION HARDWARE
; 91 ;24 REMOVE WRITE TEST FROM IO READS & WRITES
; 92 ;25 REWRITE MUL & MULI TO BE FASTER AND SMALLER. ALSO MAKE ADJBP
; 93 ; USE NEW MULSUB
; 94 ;26 MAKE BYTES USE FAST SHIFT ECO.
; 95 ;27 MAKE SURE VMA FETCH IS CORRECT
; 96 ;30 MORE OF 25 (FORGOT FMP)
; 97 ;31 FIX SOME PROBLEMS WITH TRAPS
; 98 ;32 SPEED UP EFFECTIVE ADDRESS CALCULATION
; 99 ;33 MORE OF 32
; 100 ;34 SPEED UP ASH & ROT
; 101 ;35 FIX UP RDTIM SO THAT TIME DOES NOT GO BACKWARDS
; 102 ;36 MORE OF 35
; 103 ;37 FIX UP PROBLEMS WITH INTERRUPTS AND DOUBLE F.P.
; 104 ;40 IMPROVE LISTING FORMAT
; 105 ;41 SPEEDUP KL-MODE PAGE REFILL
; 106 ;42 FIX UP DDIV
; 107 ;43 STILL MORE DDIV STUFF
; 108 ;44 CORRECT PROBLEMS IN D.P. PARITY STUFF
; 109 ;45 CORRECT THE BLT CLEAR-CORE CASE TO INTERRUPT CORRECTLY
; 110 ;46 MORE OF 45
; 111 ;47 DO NOT ALLOW SOFTWARE INTERRUPTS IF THE PI LEVEL IS NOT
; 112 ; ACTIVE.
; 113 ;50 MAKE FDV WORK THE SAME AS THE KL10
; 114 ;51 FIX INTERRUPT IN CVTBDX. MAKE ABORT WORK LIKE SPEC.
; 115 ;52 FIX BUG IN HALT LOOP
; 116 ;53 FIX IOEA TO WORK IF NO @ OR INDEXING
; 117 ;54 EDIT 47 BROKE JEN
; 118 ;55 FIX FLAGS IN MULTIPLY. ALSO CODE BUMS
; 119 ;56 MORE CODE BUMS
; 120 ;57 CORRECT OVERFLOW TRAPS WHICH DO MUUOS TO NOT STORE
; 121 ; THE TRAP FLAGS.
; 122

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

Produced on Advanced Information Services Electronic Laser Printer, PRC01566, DTN: 223-7981

; T10KL.MCR[10,1141] 15:34 27-JULY-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 7-1
; KS10.MIC[10,1141] 07:10 27-JULY-1984 REVISION HISTORY

; 179 ;122 DFAD FOLLOWED BY A FSC OF -5 CAUSES THE FSC TO GET WRONG
; 180 ; ANSWER. HAD TO CLEAR FLAG WORD AT EXIT OF DFAD TO FIX PROBLEM
; 181 ;123 MORE CODE FOR EDIT 121. ADDED ANOTHER DISPATCH ADDRESS FOR
; 182 ; PAGE FAIL CODE AT PFD:
; 183 ;124 ADD ASSEMBLY OPTIONS FOR NOCST AND INHIBIT CST UPDATE.
; 184 ; ADD BLTUB/BLTBU TO GET UBA STYLE BYTES SWAPPED TO/FROM ILDB FORM.
; 185 ; ADD ASSEMBLY OPTIONS FOR KI/KL PAGE. NEED THE SPACE FOR BLTUB/BU.
; 186 ;
; 187

; 188 .TOC "HOW TO READ THE MICROCODE"

; 189
; 190
; 191
; 192
; 193
; 194
; 195
; 196
; 197
; 198
; 199
; 200
; 201
; 202
; 203
; 204
; 205
; 206
; 207
; 208
; 209
; 210
; 211
; 212
; 213
; 214
; 215
; 216
; 217
; 218
; 219
; 220
; 221
; 222
; 223
; 224
; 225
; 226
; 227
; 228
; 229
; 230
; 231
; 232
; 233
; 234
; 235
; 236
; 237
; 238

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.

; 239 ;
; 240 ;
; 241 ;
; 242 ;
; 243 ;
; 244 ;
; 245 ;
; 246 ;
; 247 ;
; 248 ;
; 249 ;
; 250 ;
; 251 ;
; 252 ;
; 253 ;
; 254 ;
; 255 ;
; 256 ;
; 257 ;
; 258 ;
; 259 ;
; 260 ;
; 261 ;
; 262 ;
; 263 ;
; 264 ;
; 265 ;
; 266 ;
; 267 ;
; 268 ;
; 269 ;
; 270 ;
; 271 ;
; 272 ;
; 273 ;
; 274 ;
; 275 ;
; 276 ;
; 277 ;
; 278 ;
; 279 ;
; 280 ;
; 281 ;
; 282 ;
; 283 ;
; 284 ;
; 285 ;
; 286 ;
; 287 ;
; 288 ;

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

; 289 ; THE FOLLOWING LINE. EXAMPLE:
; 290 ; READ [AR], ;LOOK AT EFFECTIVE ADDRESS
; 291 ; SKIP DP18, ;SEE IF RIGHT OR LEFT SHIFT
; 292 ; SC_SHIFT-1, ;PUT NUMBER OF PLACE TO
; 293 ; ;SHIFT IN SC
; 294 ; LOAD FE, ; AND IN FE
; 295 ; INST DISP ;GO DO THE SHIFT
; 296 ;
; 297 ;
; 298 ;
; 299 ;
; 300 ;

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

; 345 ;
; 346 ;
; 347 ;
; 348 ;
; 349 ;
; 350 ;
; 351 ;
; 352 ;
; 353 ;
; 354 ;
; 355 ;
; 356 ;
; 357 ;
; 358 ;
; 359 ;
; 360 ;
; 361 ;
; 362 ;
; 363 ;
; 364 ;
; 365 ;
; 366 ;
; 367 ;
; 368 ;
; 369 ;
; 370 ;
; 371 ;
; 372 ;
; 373 ;
; 374 ;
; 375 ;
; 376 ;
; 377 ;
; 378 ;
; 379 ;
; 380 ;
; 381 ;
; 382 ;
; 383 ;
; 384 ;
; 385 ;
; 386 ;
; 387 ;
; 388 ;
; 389 ;
; 390 ;
; 391 ;
; 392 ;

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 0'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 0'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 0'S IN THE CONSTRAINT STRING, THE CONSTRAINT IMPLIES A BLOCK OF <2**N> MICROWORDS, WHERE N IS THE NUMBER OF 0'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 "0" 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 "0", "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 0'S IN THIS CONSTRAINT, THE ASSEMBLER FINDS ONLY ONE LOCATION MEETING THE CONSTRAINT.

=0*****

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.

```
; 393 .TOC "CONDITIONAL ASSEMBLY DEFINITIONS"
; 394
; 395 .DEFAULT/SIM=0 ;O=RUN ON REAL HARDWARE
; 396 ;1=RUN UNDER SIMULATOR
; 397
; 398 .DEFAULT/FULL=1 ;O=INCLUDE ONLY BASIC INSTRUCTIONS
; 399 ;1=INCLUDE FULL INSTRUCTION SET
; 400
; 401 .DEFAULT/INHCST=0 ;O=NO CODE TO INHIBIT CST UPDATE IF CSB=0
; 402 ;1=DON'T UPDATE CST IF CSB=0
; 403
; 404 .DEFAULT/NOCST=0 ;O=INCLUDE SUPPORT FOR WRITING THE CST
; 405 ;1=COMPLETELY DESUPPORT CST (FOR TOPS-10)
; 406
; 407 .DEFAULT/UBABLT=0 ;O=NO UBABLT SUPPORT
; 408 ;1=SUPPORT UBA STYLE BLT INSTRUCTIONS.
; 409
; 410 .DEFAULT/KIPAGE=1 ;O=DON'T SUPPORT KI PAGING
; 411 ;1=DO
; 412
; 413 .DEFAULT/KLPAGE=1 ;O=DON'T SUPPORT KL PAGING
; 414 ;1=DO
; 415
; 416 .IF/UBABLT ;THESE INSTRUCTIONS
; 417 .IF/KLPAGE
; 418 .CHANGE/KIPAGE=0
; 419 .ENDIF/KLPAGE
; 420 .IF/KIPAGE ;ALLOW ONLY ONE PAGING MODE
; 421 .CHANGE/KLPAGE=0
; 422 .ENDIF/KIPAGE
; 423 .ENDIF/UBABLT ;OTHERWISE, MAY HAVE EITHER OR BOTH
; 424
; 425 .IF/NOCST
; 426 .CHANGE/INHCST=0
; 427 .ENDIF/NOCST
; 428
; 429 .DEFAULT/NONSTD=0 ;O=STANDARD MICROCODE
; 430 ;1=NON-STANDARD MICROCODE
; 431
; 432 .WIDTH/108 ;ONLY FIELDS BETWEEN BITS 0 AND 107 EVER
; 433 ; GET LOADED INTO THE CRAM. OTHER FIELDS
; 434 ; ARE USED FOR DEFAULTING PROCESS.
; 435
; 436 .REGION/O,1377/2000,3777/1400,1777
; 437 ;TRY AND KEEP STUFF OUT OF DROM SPACE
; 438
```

```

; 439 .TOC "2901 REGISTER USAGE"
; 440
; 441 ;
; 442 ;0: |=====|
; 443 ; | MAG (ONES IN BITS 1-36, REST ZERO) |
; 444 ;1: |-----|
; 445 ; | PC (ADDRESS OF CURRENT INSTRUCTION + 1) |
; 446 ;2: |-----|
; 447 ; | HR (CURRENT INSTRUCTION) |
; 448 ;3: |-----|
; 449 ; | AR (TEMP -- MEM OP AT INST START) |
; 450 ;4: |-----|
; 451 ; | ARX (TEMP -- LOW ORDER HALF OF DOUBLE PREC) |
; 452 ;5: |-----|
; 453 ; | BR (TEMP) |
; 454 ;6: |-----|
; 455 ; | BRX (TEMP -- LOW ORDER HALF OF DOUBLE PREC BR!BRX) |
; 456 ;7: |-----|
; 457 ; | ONE (THE CONSTANT 1) |
; 458 ;10: |-----|
; 459 ; | EBR (EXEC BASE REGISTER) |
; 460 ;11: |-----|
; 461 ; | UBR (USER BASE REGISTER) |
; 462 ;12: |-----|
; 463 ; | MASK (ONES IN BITS 0-35, ZERO IN -1, -2, 36 AND 37) |
; 464 ;13: |-----|
; 465 ; | FLG (FLAG BITS) ! PAGE FAIL CODE |
; 466 ;14: |-----|
; 467 ; | PI (PI SYSTEM STATUS REGISTER [RDPI]) |
; 468 ;15: |-----|
; 469 ; | XWD1 (1 IN EACH HALF WORD) |
; 470 ;16: |-----|
; 471 ; | TO (TEMP) |
; 472 ;17: |-----|
; 473 ; | T1 (TEMP) |
; 474 ; |=====|
  
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES5, DTN: 233-7881

```

; 475      .TOC      "MICROCODE FIELDS -- LISTING FORMAT"
; 476
; 477
; 478
; 479
; 480      U 1561, 1500,2551,0303,0274,4463,7701,4200,0001,0001
; 481      [--] [--] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!] [!]
; 482
; 483      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 484      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 485      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 486      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 487      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 488      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 489      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 490      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 491      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 492      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 493      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 494      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 495      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 496      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 497      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 498      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 499      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 500      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 501      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 502      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 503      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 504      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 505      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 506      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 507      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 508      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 509      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 510      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 511      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 512      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 513      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 514      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 515      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 516      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 517      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 518      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 519      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 520      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 521      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !
; 522      !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !          !

```

; 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

Produced on Advanced Information Services Electronic Laser Printer. PK011556, DTN: 223-7981

```
; 523 .TOC "MICROCODE FIELDS -- DATAPATH CHIP"
; 524
; 525 J/= <0:11>+ ;CRA1
; 526 ;NEXT MICROCODE ADDRESS
; 527
; 528 ;ALU FUNCTIONS
; 529
; 530 ;NOTE: THE AD FIELD IS A 2 DIGIT FIELD. THE LEFT DIGIT IS THE
; 531 ; 2901 ALU FUNCTION. THE RIGHT DIGIT IS THE 2901 SRC CODE FOR
; 532 ; THE LEFT HALF. NORMALY THE RIGHT HALF SRC CODE IS THE SAME AS
; 533 ; THE LEFT HALF.
; 534 AD/= <12:17>D,44 ;DPE1 & DPE2
; 535 A+Q=00
; 536 A+B=01
; 537 O+Q=02
; 538 O+B=03
; 539 O+A=04
; 540 D+A=05
; 541 D+Q=06
; 542 O+D=07
; 543 Q-A-.25=10
; 544 B-A-.25=11
; 545 Q-.25=12
; 546 B-.25=13
; 547 A-.25=14
; 548 A-D-.25=15
; 549 Q-D-.25=16
; 550 -D-.25=17
; 551 A-Q-.25=20
; 552 A-B-.25=21
; 553 -Q-.25=22
; 554 -B-.25=23
; 555 -A-.25=24
; 556 D-A-.25=25
; 557 D-Q-.25=26
; 558 D-.25=27
; 559 A.OR.Q=30
; 560 A.OR.B=31
; 561 Q=32
; 562 B=33
; 563 A=34
; 564 D.OR.A=35
; 565 D.OR.Q=36
; 566 D=37
; 567 A.AND.Q=40
; 568 A.AND.B=41
; 569
```


; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 570 ;MORE ALU FUNCTIONS
; 571
; 572     ZERO=42
; 573 ;   ZERO=43
; 574 ;   ZERO=44
; 575 ;   D.AND.A=45
; 576 ;   D.AND.Q=46
; 577 ;   ZERO=47
; 578 ;   .NOT.A.AND.Q=50
; 579 ;   .NOT.A.AND.B=51
; 580 ;   Q=52
; 581 ;   B=53
; 582 ;   A=54
; 583 ;   .NOT.D.AND.A=55
; 584 ;   .NOT.D.AND.Q=56
; 585 ;   ZERO=57
; 586 ;   A.XOR.Q=60
; 587 ;   A.XOR.B=61
; 588 ;   Q=62
; 589 ;   B=63
; 590 ;   A=64
; 591 ;   D.XOR.A=65
; 592 ;   D.XOR.Q=66
; 593 ;   D=67
; 594 ;   A.EQV.Q=70
; 595 ;   A.EQV.B=71
; 596 ;   .NOT.Q=72
; 597 ;   .NOT.B=73
; 598 ;   .NOT.A=74
; 599 ;   D.EQV.A=75
; 600 ;   D.EQV.Q=76
; 601 ;   .NOT.D=77
; 602
; 603 ;THIS FIELD IS THE RIGHTMOST 3 BITS OF THE
; 604 ; AD FIELD. IT IS USED ONLY TO DEFAULT THE RSRC
; 605 ; FIELD.
; 606 LSRC/=<15:17>           ;DPE1
; 607
; 608 ;THIS IS THE SOURCE FOR THE RIGHT HALF OF THE
; 609 ; DATA PATH. IT LETS US MAKE THE RIGHT AND LEFT
; 610 ; HALF WORDS DO SLIGHTLY DIFFERENT THINGS.
; 611 RSRC/=<18:20>F,LSRC    ;DPE2
; 612     AQ=0               ;A Q
; 613     AB=1               ;A B
; 614     OQ=2               ;O Q
; 615     OB=3               ;O B
; 616     OA=4               ;O A
; 617     DA=5               ;D A
; 618     DQ=6               ;D Q
; 619     DO=7               ;D O
; 620
```

```
; 621 ;DESTINATION CONTROL
; 622 ;SEE DPE1 AND DPE2 (2'S WEIGHT IS INVERTED ON DPE5)
; 623 DEST/=<21:23>D,3 ;DPE1 & DPE2
; 624 A=0 ;A REG IS CHIP OUTPUT, AD IS WRITTEN
; 625 ; INTO REG FILE
; 626 AD=1 ;REG FILE GETS AD
; 627 Q_AD=2 ;REG FILE IS NOT LOADED
; 628 PASS=3 ;AD OUTPUT IS CHIP OUTPUT
; 629 ; Q AND REG FILE LEFT ALONE
; 630 Q_Q*2=4 ;ALSO REG FILE GETS AD*2
; 631 AD*2=5 ;AND Q IS LEFT ALONE
; 632 Q_Q*.5=6 ;ALSO REG FILE GETS AD*.5
; 633 AD*.5=7 ;AND Q IS LEFT ALONE
; 634
; 635 ; <24:25> ;UNUSED
; 636
; 637 A/=<26:29> ;DPE1 & DPE2
; 638 MAG=0
; 639 PC=1
; 640 HR=2
; 641 AR=3
; 642 ARX=4
; 643 BR=5
; 644 BRX=6
; 645 ONE=7
; 646 EBR=10
; 647 UBR=11
; 648 MASK=12
; 649 FLG=13
; 650 PI=14
; 651 XWD1=15
; 652 TO=16
; 653 T1=17
; 654
; 655 ; <30:31> ;UNUSED
; 656
; 657 B/=<32:35>D,0 ;DPE1 & DPE2
; 658 MAG=0
; 659 PC=1
; 660 HR=2
; 661 AR=3
; 662 ARX=4
; 663 BR=5
; 664 BRX=6
; 665 ONE=7
; 666 EBR=10
; 667 UBR=11
; 668 MASK=12
; 669 FLG=13
; 670 PI=14
; 671 XWD1=15
; 672 TO=16
; 673 T1=17
; 674
```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-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

```
; 675 .TOC "MICROCODE FIELDS -- RAM FILE ADDRESS AND D-BUS"  
; 676  
; 677 RAMADR/=<36:38>D,4 ;DPE6  
; 678 AC#=0 ;AC NUMBER  
; 679 AC*#=1 ;AC .FN. #  
; 680 XR#=2 ;INDEX REGISTER  
; 681 VMA=4 ;VIRTUAL MEMORY REFERENCE  
; 682 RAM=6 ;VMA SUPPLIES 10-BIT RAM ADDRESS  
; 683 #=7 ;ABSOLUTE RAM FILE REFERENCE  
; 684  
; 685 ; <39:39>  
; 686  
; 687 ;LEFT HALF ON DPE3 AND RIGHT HALF ON DPE4  
; 688 DBUS/=<40:41>D,1 ;DPE3 & DPE4  
; 689 PC FLAGS=0 ;PC FLAGS IN LEFT HALF  
; 690 PI NEW=0 ;NEW PI LEVEL IN BITS 19-21  
; 691 ; VMA=0 ;VMA IN BITS 27-35  
; 692 DP=1 ;DATA PATH  
; 693 RAM=2 ;CACHE, AC'S AND WORKSPACE  
; 694 DBM=3 ;DBM MIXER  
; 695  
; 696 ;LEFT HALF ON DPM1 AND RIGHT HALF ON DPM2  
; 697 DBM/=<42:44>D,7 ;DPM1 & DPM2  
; 698 SCAD DIAG=0 ;(LH) SCAD DIAGNOSTIC  
; 699 PF DISP=0 ;PAGE FAIL DISP IN BITS 18-21  
; 700 APR FLAGS=0 ;APR FLAGS IN BITS 22-35  
; 701 BYTES=1 ;5 COPIES OF SCAD 1-7  
; 702 EXP=2 ;LH=EXPONENT, RH=TIME FRACTION  
; 703 DP=3 ;DATA PATH  
; 704 DP SWAP=4 ;DATA PATH SWAPPED  
; 705 VMA=5 ;VMA FLAGS, VMA  
; 706 MEM=6 ;MEMORY BUFFER  
; 707 #=7 ;NUMBER FIELD IN BOTH HALVES  
; 708
```

```
; 709 .TOC "MICROCODE FIELDS -- PARITY GENERATION & HALF WORD CONTROL"  
; 710  
; 711 AD PARITY OK/=<108>D,0 ;**NOT STORED IN CRAM**  
; 712 ;THIS BIT IS A 1 IF THE ALU IS DOING  
; 713 ; SOMETHING WHICH DOES NOT INVALIDATE  
; 714 ; PARITY. IT DOES NOT APPEAR IN THE  
; 715 ; REAL MACHINE. WE JUST USE IT TO SET  
; 716 ; THE DEFAULT FOR GENR & GENL  
; 717  
; 718 CLKL/=<45:45>D,1 ;DPE5  
; 719 ;CLOCK THE LEFT HALF OF THE MACHINE  
; 720 GENL/=<46:46>F,AD PARITY OK ;DPE4 FROM CRM2 PARITY EN LEFT H  
; 721 ;STORE PARITY FOR 2901 LEFT  
; 722 CHKL/=<47:47> ;DPE4 FROM CRM2 PARITY CHK LEFT H  
; 723 ;CHECK LEFT HALF DBUS PARITY  
; 724  
; 725 CLKR/=<48:48>D,1 ;DPE5  
; 726 ;CLOCK THE RIGHT HALF OF THE MACHINE  
; 727 GENR/=<49:49>F,AD PARITY OK ;DPE4 FROM CRM2 PARITY EN RIGHT H  
; 728 ;STORE PARITY FOR 2901 RIGHT  
; 729 CHKR/=<50:50> ;DPE4 FROM CRM2 PARITY CHK RIGHT H  
; 730 ;CHECK RIGHT HALF DBUS PARITY  
; 731
```

: 732 .TOC "MICROCODE FIELDS -- SPEC"

: 733
: 734
: 735
: 736
: 737
: 738
: 739
: 740
: 741
: 742
: 743
: 744
: 745
: 746
: 747
: 748
: 749
: 750
: 751
: 752
: 753
: 754
: 755
: 756
: 757
: 758
: 759
: 760
: 761
: 762
: 763
: 764
: 765
: 766
: 767
: 768
: 769
: 770
: 771
: 772
: 773
: 774
: 775
: 776
: 777
: 778
: 779
: 780
: 781
: 782
: 783
: 784

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 !	! ACTION WHEN DBM		!
! P !	! SELECTS DP		!
! E !	! GET DP BITS ! GET SCAD 1-7		!
! C !	!		!
! 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		!

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

; 788
; 789 SPEC/= <51:56>D,0 ;DPE1 & DPE5 & DPM1 & DPMA
; 790 #=10 ;DECODE # BITS
; 791 CLRCLK=11 ;CLEAR 1MS NICOND FLAG
; 792 CLR IO LATCH=12 ;CLEAR IO LATCH
; 793 CLR IO BUSY=13 ;CLEAR IO BUSY
; 794 LDPAGE=14 ;WRITE PAGE TABLE
; 795 NICOND=15 ;DOING NICOND DISPATCH
; 796 LDPXCT=16 ;LOAD PXCT FLAGS
; 797 WAIT=17 ;MEM WAIT
; 798 PREV=20 ;FORCE PREVIOUS CONTEXT
; 799 LOADXR=21 ;LOAD XR #, USES PXCT FIELD TO SELECT
; 800 ; CORRECT AC BLOCK
; 801 APR FLAGS=23 ;LOAD APR FLAGS
; 802 CLRCSH=24 ;CLEAR CACHE
; 803 APR EN=25 ;SET APR ENABLES
; 804 MEMCLR=27 ;CLEAR PAGE FAULT CONDITION
; 805 SWEEP=34 ;SET SWEEP
; 806 PXCT OFF=36 ;TURN OFF THE EFFECT OF PXCT
; 807 INHCRY18=40 ;INHIBIT CARRY INTO LEFT HALF
; 808 LOADIR=41 ;LOAD THE IR
; 809 LDPI=43 ;LOAD PI SYSTEM
; 810 ASHOV=44 ;TEST RESULT OF ASH
; 811 EXPTST=45 ;TEST RESULT OF FLOATING POINT
; 812 FLAGS=46 ;CHANGE PC FLAGS
; 813 LDACBLK=47 ;LOAD AC BLOCK NUMBERS
; 814 LDINST=64 ;LOAD INSTRUCTION

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

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

; 823
; 824 ;THE SPEC FIELD IS REDEFINED WHEN USED TO CONTROL SHIFT PATHS
; 825 SHSTYLE/= <54:56> ;DPE1 (SPEC SEL)
; 826 NORM=0 ;2 40-BIT REGISTERS
; 827 ZERO=1 ;SHIFT ZERO INTO 36 BITS (ASH TOP 2901)
; 828 ONES=2 ;SHIFT IN ONES
; 829 ROT=3 ;ROTATE
; 830 ASHC=4 ;ASHC
; 831 LSHC=5 ;LSHC
; 832 DIV=6 ;SPECIAL DIVIDE
; 833 ROTC=7 ;ROTATE DOUBLE
; 834

```
835 .TOC "MICROCODE FIELDS -- DISPATCH"
836 :
837 : ! D ! CRA1 ! CRA1 ! DPEA
838 : ! I ! DISP ! DISP ! DISP
839 : ! S ! 10 ! 20 ! 40
840 : ! P ! ! ! !
841 :
842 : ! O ! DIAG ADR ! DIAG ADR ! O
843 :
844 : ! 1 ! RETURN ! RETURN ! DP 18-21
845 :
846 : ! 2 ! MULTIPLY ! J ! J
847 :
848 : ! 3 ! PAGE FAIL ! AREAD ! AREAD !
849 :
850 : ! 4 ! NICOND ! NOT USABLE ! NORM
851 :
852 : ! 5 ! BYTE ! NOT USABLE ! DP 32-35
853 :
854 : ! 6 ! EA MODE ! NOT USABLE ! DROM A
855 :
856 : ! 7 ! SCAD ! NOT USABLE ! DROM B
857 :
```

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

- 861
- 862 DISP/= <57:62>D,70 ;CRA1 & DPEA
- 863 CONSOLE=00 ;CONSOLE DISPATCH
- 864 DROM=12 ;DROM
- 865 AREAD=13 ;AREAD
- 866 DP LEFT=31 ;DP 18-21
- 867 NORM=34 ;NORMALIZE
- 868 DP=35 ;DP 32-35
- 869 ADISP=36 ;DROM A FIELD
- 870 BDISP=37 ;DROM B FIELD
- 871 RETURN=41 ;RETURN
- 872 MUL=62 ;MULTIPLY
- 873 PAGE FAIL=63 ;PAGE FAIL
- 874 NICOND=64 ;NEXT INSTRUCTION DISPATCH
- 875 BYTE=65 ;BYTE SIZE AND POSITION
- 876 EAMODE=66 ;EFFECTIVE ADDRESS MODE
- 877 SCADO=67 ;J!2 IF SCAD BIT 0 = 1
- 878

```

; 879 .TOC "MICROCODE FIELDS -- SKIP"
; 880 ;
; 881 ; ! S ! CRA2 ! DPEA ! DPEA !
; 882 ; ! K ! SKIP ! SKIP ! SKIP !
; 883 ; ! I ! 10 ! 20 ! 40 !
; 884 ; ! P ! ! ! ! !
; 885 ; !-----!
; 886 ; ! O ! 0 ! 0 ! 0 !
; 887 ; !-----!
; 888 ; ! 1 ! TRAP CYCLE ! CRY O2 ! CARRY OUT !
; 889 ; !-----!
; 890 ; ! 2 ! AD=0 ! ADL SIGN ! ADL=0 !
; 891 ; !-----!
; 892 ; ! 3 ! SC SIGN ! ADR SIGN ! ADR=0 !
; 893 ; !-----!
; 894 ; ! 4 ! EXECUTE ! USER IOT ! -USER !
; 895 ; !-----!
; 896 ; ! 5 ! -BUS IO BUSY ! JFCL SKIP ! FPD FLAG !
; 897 ; !-----!
; 898 ; ! 6 ! -CONTINUE ! CRY O1 ! AC # IS ZERO !
; 899 ; !-----!
; 900 ; ! 7 ! -1 MSEC ! TXXX ! INTERRUPT REQ !
; 901 ; !-----!

```

```

; 903 SKIP/=63:68>D,70 ;CRA2 & DPEA
; 904 IOLGL=04 ;(.NOT.USER)!(USER IOT)!(CONSOLE EXECUTE MODE)
; 905 LLE=12 ;AD LEFT .LE. 0
; 906 CRYO=31 ;AD CRY -2
; 907 ADLEQO=32 ;ADDER LEFT = 0
; 908 ADREQO=33 ;ADDER RIGHT = 0
; 909 KERNEL=34 ;.NOT. USER
; 910 FPD=35 ;FIRST PART DONE
; 911 ACO=36 ;AC NUMBER IS ZERO
; 912 INT=37 ;INTERRUPT REQUEST
; 913 LE=42 ;(AD SIGN)!(AD.EQ.O)
; 914 CRY2=51 ;AD CRY O2
; 915 DPO=52 ;AD SIGN
; 916 DP18=53 ;AD BIT 18
; 917 IOT=54 ;USER IOT
; 918 JFCL=55 ;JFCL SKIP
; 919 CRY1=56 ;AD CRY 1
; 920 TXXX=57 ;TEST INSTRUCTION SHOULD SKIP
; 921 TRAP CYCLE=61 ;THIS INSTRUCTION IS THE RESULT OF A
; 922 ; TRAP 1, 2, OR 3
; 923 ADEQO=62 ;AD.EQ.O
; 924 SC=63 ;SC SIGN BIT
; 925 EXECUTE=64 ;CONSOLE EXECUTE MODE
; 926 -IO BUSY=65 ;.NOT. I/O LATCH
; 927 -CONTINUE=66 ;.NOT. CONTINUE
; 928 -1 MS=67 ;.NOT. 1 MS. TIMER
; 929

```

Produced on Advanced Information Services Electronic Laser Printer, PK01E66, DTN: 223-7881

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 930 .TOC "MICROCODE FIELDS -- TIME CONTROL"  
; 931  
; 932 DT/=<109:111>D,0 ;**NOT STORED IN CRAM**  
; 933 ;DEFAULT TIME FIELD (USED IN MACROS)  
; 934 ; CAN BE OVERRIDDEN IN MACRO CALL  
; 935 2T=0  
; 936 3T=1  
; 937 4T=2  
; 938 5T=3  
; 939  
; 940  
; 941 T/=<69:71>F,DT ;CSL5 (E601)  
; 942 ;CLOCK TICKS MINUS TWO REQUIRED TO  
; 943 ; DO A MICRO INSTRUCTION  
; 944 2T=0 ;TWO TICKS  
; 945 3T=1 ;THREE TICKS  
; 946 4T=2 ;FOUR TICKS  
; 947 5T=3 ;FIVE TICKS  
; 948  
; 949
```

```
; 950 .TOC "MICROCODE FIELDS -- RANDOM CONTROL BITS"  
; 951  
; 952 CRY38/=<72> ;DPE5  
; 953 ;INJECT A CARRY INTO THE 2901 ADDER  
; 954 LOADSC/=<73> ;DPM4  
; 955 ;LOAD THE STEP COUNTER FROM THE SCAD  
; 956 LOADFE/=<74> ;DPM4  
; 957 ;LOAD THE FE REGISTER FROM THE SCAD  
; 958 FMWRITE/=<75> ;DPE5 (E302)  
; 959 ;WRITE THE RAM FILE.  
; 960 MEM/=<76> ;DPM5 (E612) & DPE5 (E205)  
; 961 ;START (OR COMPLETE) A MEMORY OR I/O CYCLE UNDER  
; 962 ; CONTROL OF THE NUMBER FIELD.  
; 963 DIVIDE/=<77> ;DPE5  
; 964 ;THIS MICROINSTRUCTION IS DOING A DIVIDE  
; 965 MULTI PREC/=<78> ;DPE5  
; 966 ;MULTIPRECISION STEP IN DIVIDE, DFAD, DFSB  
; 967 MULTI SHIFT/=<79> ;CSL5 (HAS NOTHING TO DO WITH DPE5 MULTI SHIFT)  
; 968 ;FAST SHIFT  
; 969 CALL/=<80> ;CRA2 (STACK IS ON CRA3)  
; 970 ;THIS IS A CALL  
; 971
```

```
; 972 .TOC "MICROCODE FIELDS -- NUMBER FIELD"  
; 973  
; 974 ;HERE IS THE GENERAL FIELD  
; 975 #/= <90:107> ;MANY PLACES  
; 976  
; 977 ;# REDEFINED WHEN USED AS SCAD CONTROL:  
; 978 SCAD/= <90:92> ;DPM3  
; 979 A*2=0  
; 980 A.OR.B=1  
; 981 A-B-1=2  
; 982 A-B=3  
; 983 A+B=4  
; 984 A.AND.B=5  
; 985 A-1=6  
; 986 A=7  
; 987 SCADA/= <93:95> ;DPM3  
; 988 SC=0  
; 989 S#=1  
; 990 PTR44=2 ;44 AND BIT 6 (SEE DPM3)  
; 991 BYTE1=3  
; 992 BYTE2=4  
; 993 BYTE3=5  
; 994 BYTE4=6  
; 995 BYTE5=7  
; 996 SCADB/= <96:97> ;DPM3  
; 997 FE=0  
; 998 EXP=1  
; 999 SHIFT=2  
; 1000 SIZE=3  
; 1001 S#/= <98:107> ;DPM3  
; 1002  
; 1003 ;# REDEFINED WHEN USED AS STATE REGISTER CONTROL:  
; 1004 STATE/= <90:107> ;NOT USED BY HARDWARE  
; 1005 SIMPLE=0 ;SIMPLE INSTRUCTIONS  
; 1006 BLT=1 ;BLT IN PROGRESS  
; 1007 MAP=400002 ;MAP IN PROGRESS  
; 1008 SRC=3 ;MOVE STRING SOURCE IN PROGRESS  
; 1009 DST=4 ;MOVE STRING FILL IN PROGRESS  
; 1010 SRC+DST=5 ;MOVE STRING DEST IN PROGRESS  
; 1011 DSTF=6 ;FILLING DEST  
; 1012 CVTDB=7 ;CONVERT DEC TO BIN  
; 1013 COMP-DST=10 ;COMPARE DEST  
; 1014 EDIT-SRC=11 ;EDIT SOURCE  
; 1015 EDIT-DST=12 ;EDIT DEST  
; 1016 EDIT-S+D=13 ;BOTH SRC AND DST POINTERS  
; 1017
```

```

; 1018 ;# REDEFINED WHEN USED AS WORSPACE ADDRESS
; 1019
; 1020 WORK/= <98:107> ;DPE6
; 1021 BADWO=160 ;AC BLK 7 WORD 0 (BAD DATA FROM MEMORY)
; 1022 BADW1=161 ;AC BLK 7 WORD 1 (BAD DATA FROM MEMORY)
; 1023 MUL=200 ;TEMP FOR MULTIPLY
; 1024 DIV=201 ;TEMP FOR DIVIDE
; 1025 SV.VMA=210 ;SAVE VMA
; 1026 SV.AR=211 ;SAVE AR
; 1027 SV.ARX=212 ;SAVE ARX
; 1028 SV.BR=213 ;SAVE BR
; 1029 SV.BRX=214 ;SAVE BRX
; 1030 SBR=215 ;SPT BASE REGISTER
; 1031 CBR=216 ;CST BASE ADDRESS
; 1032 CSTM=217 ;CST MASK
; 1033 PUR=220 ;PROCESS USE REGISTER
; 1034 ADJP=221 ;"P" FOR ADJBP
; 1035 ADJS=222 ;"S" FOR ADJBP
; 1036 ADJPTR=223 ;BYTE POINTER FOR ADJBP
; 1037 ADJQ1=224 ;TEMP FOR ADJBP
; 1038 ADJR2=225 ;TEMP FOR ADJBP
; 1039 ADJBPW=226 ;(BYTES/WORD) FOR ADJBP
; 1040 HSBADR=227 ;ADDRESS OF HALT STATUS BLOCK
; 1041 APR=230 ;APR ENABLES
; 1042 ;THE FOLLOWING WORDS ARE USED BY EXTEND INSTRUCTION
; 1043 EO=240 ;ORIGINAL EFFECTIVE ADDRESS
; 1044 E1=241 ;EFFECTIVE ADDRESS OF WORD AT EO
; 1045 SLEN=242 ;SOURCE LENGTH
; 1046 MSK=243 ;BYTE MASK
; 1047 FILL=244 ;FILL BYTE
; 1048 CMS=245 ;SRC BYTE IN STRING COMPARE
; 1049 FSIG=246 ;PLACE TO SAVE ARX WHILE STORING
; 1050 ; THE FLOAT CHAR
; 1051 BDH=247 ;BINARY BEING CONVERTED TO
; 1052 BDL=250 ; DECIMAL
; 1053
; 1054 ;TIMER STUFF
; 1055 TIME0=300 ;HIGH ORDER 36 BITS OF TIME
; 1056 TIME1=301 ;LOW ORDER 36 BITS OF TIME
; 1057 PERIOD=302 ;INTERRUPT PERIOD
; 1058 TTG=303 ;TIME TO GO TO NEXT INTERRUPT
; 1059

```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1060 ;DDIV STUFF
; 1061     ACO=314
; 1062     AC1=315
; 1063     AC2=316
; 1064     AC3=317
; 1065     DDIV SGN=320
; 1066     DVSOR H=321
; 1067     DVSOR L=322
; 1068 ;POWERS OF TEN
; 1069     DECL0=344     ;LOW WORD
; 1070     DECHI=373     ;HIGH WORD
; 1071
; 1072     YSAVE=422     ;Y OF LAST INDIRECT POINTER
; 1073     PTA.E=423     ;ADDRESS OF EXEC PAGE MAP (NOT PROCESS TABLE)
; 1074     PTA.U=424     ;ADDRESS OF USER PAGE MAP
; 1075     TRAPPC=425    ;SAVED PC FROM TRAP CYCLE
; 1076     SV.AR1=426    ;ANOTHER PLACE TO SAVE AR
; 1077
```

```
; 1078 ;# REDEFINED WHEN USED AS PC FLAG CONTROL (ALL ON DPE9)
; 1079
; 1080 SETOV/= <90> ;DPE9
; 1081 ;SET ARITHMETIC OVERFLOW
; 1082 SETFOV/= <91> ;SET FLOATING OVERFLOW
; 1083 SETNDV/= <92> ;SET NO DIVIDE
; 1084
; 1085 ;-----
; 1086
; 1087 CLRFPD/= <93> ;CLEAR FIRST PART DONE
; 1088 SETFPD/= <94> ;SET FIRST PART DONE
; 1089 HOLD USER/= <95> ;WHEN THIS BIT IS SET IT:
; 1090 ; 1. PREVENTS SETTING USER IOT IN USER MODE
; 1091 ; 2. PREVENTS CLEARING USER IN USER MODE
; 1092
; 1093 ;-----
; 1094
; 1095 ; <96> ;SPARE
; 1096 TRAP2/= <97> ;SET TRAP 2
; 1097 TRAP1/= <98> ;SET TRAP 1
; 1098
; 1099 ;-----
; 1100
; 1101 LD PCU/= <99> ;LOAD PCU FROM USER
; 1102 ; <100> ;SPARE
; 1103 ; <101> ;SPARE
; 1104
; 1105 ;-----
; 1106
; 1107 ; <102> ;SPARE
; 1108 ; <103> ;SPARE
; 1109 JFCLFLG/= <104> ;DO A JFCL INSTRUCTION
; 1110
; 1111 ;-----
; 1112
; 1113 LD' FLAGS/= <105> ;LOAD FLAGS FROM DP
; 1114 ; <106>
; 1115 ADFLGS/= <107> ;UPDATE CARRY FLAGS
; 1116
```

```
; 1117 ;# REDEFINED WHEN USED AS MEMORY CYCLE CONTROL
; 1118
; 1119 FORCE USER/=<90> ;FORCE USER MODE REFERENCE
; 1120 FORCE EXEC/=<91> ;FORCE EXEC MODE REFERENCE
; 1121 ; (DOES NOT WORK UNDER PXCT)
; 1122 FETCH/=<92> ;THIS IS AN INSTRUCTION FETCH
; 1123
; 1124 ;-----
; 1125
; 1126 READ CYCLE/=<93> ;SELECT A READ CYCLE
; 1127 WRITE TEST/=<94> ;PAGE FAILE IF NOT WRITTEN
; 1128 WRITE CYCLE/=<95> ;SELECT A MEMORY WRITE CYCLE
; 1129
; 1130 ;-----
; 1131
; 1132 ; <96> ;SPARE BIT
; 1133 DONT CACHE/=<97> ;DO NOT LOOK IN CACHE
; 1134 PHYSICAL/=<98> ;DO NOT INVOKE PAGING HARDWARE
; 1135
; 1136 ;-----
; 1137
; 1138 PXCT/=<99:101> ;WHICH PXCT BITS TO LOOK AT
; 1139 CURRENT=0
; 1140 E1=1
; 1141 D1=3
; 1142 BIS-SRC-EA=4
; 1143 E2=5
; 1144 BIS-DST-EA=6
; 1145 D2=7
; 1146
; 1147 ;-----
; 1148
; 1149 AREAD/=<102> ;LET DROM SELECT SYSLE TYPE AND VMA LOAD
; 1150 DP FUNC/=<103> ;IGNORE # BITS 0-11 AND USE DP 0-13 INSTEAD
; 1151 ; DP9 MEANS "FORCE PREVIOUS"
; 1152 LDVMA/=<104> ;LOAD THE VMA
; 1153
; 1154 ;-----
; 1155
; 1156 EXT ADR/=<105> ;PUT VMA BITS 14-17 ONTO BUS
; 1157 WAIT/=<106> ;START A MEMORY OR I/O CYCLE
; 1158 BWRITE/=<107> ;START A MEMORY CYCLE IF DROM ASKS FOR IT
; 1159
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

; 1160 ;THESE BITS ARE USED ONLY TO SETUP DP FOR A DP FUNCTION
; 1161
; 1162 ; <99> ;PREVIOUS
; 1163 IO CYCLE/= <100> ;THIS IS AN I/O CYCLE
; 1164 WRU CYCLE/= <101> ;WHO ARE YOU CYCLE
; 1165
; 1166 ;-----
; 1167
; 1168 VECTOR CYCLE/= <102> ;READ INTERRUPT VECTOR
; 1169 IO BYTE/= <103> ;BYTE CYCLE
; 1170 ; <104>
; 1171

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1172 ;# REDEFINED WHEN USED AS PI RIGHT BITS
; 1173 PI.ZER/=<90:92> ;ZEROS
; 1174 PI.IP1/=<93> ;PI 1 IN PROG
; 1175 PI.IP2/=<94>
; 1176 PI.IP3/=<95>
; 1177 PI.IP4/=<96>
; 1178 PI.IP5/=<97>
; 1179 PI.IP6/=<98>
; 1180 PI.IP7/=<99>
; 1181 PI.ON/=<100> ;SYSTEM IS ON
; 1182 PI.CO1/=<101> ;CHAN 1 IS ON
; 1183 PI.CO2/=<102>
; 1184 I.CO3/=<103>
; 1185 I.CO4/=<104>
; 1186 I.CO5/=<105>
; 1187 I.CO6/=<106>
; 1188 I.CO7/=<107>
; 1189
; 1190 ;# REDEFINED WHEN USED AS WRPI DATA
; 1191 PI.MBZ/=<90:93> ;MUST BE ZERO
; 1192 PI.DIR/=<94> ;DROP INTERRUPT REQUESTS
; 1193 PI.CLR/=<95> ;CLEAR SYSTEM
; 1194 PI.REQ/=<96> ;REQUEST INTERRUPT
; 1195 PI.TCN/=<97> ;TURN CHANNEL ON
; 1196 PI.TCF/=<98> ;TURN CHANNEL OFF
; 1197 PI.TSF/=<99> ;TURN SYSTEM OFF
; 1198 PI.TSN/=<100> ;TURN SYSTEM ON
; 1199 PI.SC1/=<101> ;SELECT CHANNEL 1
; 1200 PI.SC2/=<102>
; 1201 PI.SC3/=<103>
; 1202 PI.SC4/=<104>
; 1203 PI.SC5/=<105>
; 1204 PI.SC6/=<106>
; 1205 PI.SC7/=<107>
; 1206
```

```

; 1207 ;# REDEFINED WHEN USED AS AC CONTROL
; 1208
; 1209
; 1210 ;THIS FIELD CONTROLS THE INPUT TO A 74LS181 ON DPE6. THE NUMBER
; 1211 ; FIELD HAS THIS FORMAT IN <98:107>:
; 1212 ;
; 1213 ; |-----|-----|-----|-----|-----|-----|-----|-----|
; 1214 ; |CARRY| S8  | S4 | S2  | S1  | MODE| B8  | B4  | B2  | B1  |
; 1215 ; | IN  |      |     |     |     |    |    |    |    |    |
; 1216 ; |-----|-----|-----|-----|-----|-----|-----|-----|
; 1217 ;
; 1218
; 1219 ACALU/=<98:103>
; 1220     B=25
; 1221     AC+N=62
; 1222 ACN/=<104:107>
; 1223
; 1224     SRCLEN=0      ;AC NAMES FOR STRING INSTRUCTIONS
; 1225     SRCP=1      ;SOURCE LENGTH
; 1226     DLEN=3      ;SOURCE POINTER
; 1227     DSTP=4      ;DEST LENGTH
; 1228     MARK=3      ;DEST POINTER
; 1229     BINO=3      ;POINTER TO MARK
; 1230     BIN1=4      ;HIGH WORD OF BINARY
; 1231     BIN1=4      ;LOW WORD OF BINARY

```

Produced on Advanced Information Services Electronic Laser Printer. PK01155, DTN: 223-7781

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1232 ;# FIELD REDEFINED WHEN USE AS APRID DATA
; 1233 MICROCODE OPTIONS/=<90:98>
; 1234 ;100 - NON-STANDARD MICROCODE
; 1235 ;200 - NO CST AT ALL
; 1236 ;400 - INHIBIT CST UPDATE IS AVAILABLE
; 1237 ;040 - UBABL INSTRUCTIONS ARE PRESENT
; 1238 ;020 - KI PAGING IS PRESENT
; 1239 ;010 - KL PAGING IS PRESENT
; 1240 MICROCODE OPTION(INHCST)/=<90>
; 1241 .IF/INHCST
; 1242 .OPT=1
; 1243 .IFNOT/INHCST
; 1244 .OPT=0
; 1245 .ENDIF/INHCST
; 1246 MICROCODE OPTION(NOCST)/=<91>
; 1247 .IF/NOCST
; 1248 .OPT=1
; 1249 .IFNOT/NOCST
; 1250 .OPT=0
; 1251 .ENDIF/NOCST
; 1252 MICROCODE OPTION(NONSTD)/=<92>
; 1253 .IF/NONSTD
; 1254 .OPT=1
; 1255 .IFNOT/NONSTD
; 1256 .OPT=0
; 1257 .ENDIF/NONSTD
; 1258 MICROCODE OPTION(UBABLT)/=<93>
; 1259 .IF/UBABLT
; 1260 .OPT=1
; 1261 .IFNOT/UBABLT
; 1262 .OPT=0
; 1263 .ENDIF/UBABLT
; 1264 MICROCODE OPTION(KIPAGE)/=<94>
; 1265 .IF/KIPAGE
; 1266 .OPT=1
; 1267 .IFNOT/KIPAGE
; 1268 .OPT=0
; 1269 .ENDIF/KIPAGE
; 1270 MICROCODE OPTION(KLPAGE)/=<95>
; 1271 .IF/KLPAGE
; 1272 .OPT=1
; 1273 .IFNOT/KLPAGE
; 1274 .OPT=0
; 1275 .ENDIF/KLPAGE
; 1276
; 1277 MICROCODE VERSION/=<99:107>
; 1278 UCV=124 125
; 1279
```

MICROCODE RELEASE (MAJOR) /= <99:104>

UCR = 2

MICROCODE RELEASE (MINOR) /= <105:107>

UCR = 0

; MAJOR VERSION # (99, 2)

; MINOR (SOUP) VERSION # (105, 1A)

```
; 1280 ;# FIELD REDEFINED WHEN USED AS A HALT CODE
; 1281
; 1282 HALT/= <90:107>
; 1283 ;CODES 0 TO 77 ARE "NORMAL" HALTS
; 1284 POWER=0 ;POWER UP
; 1285 HALT=1 ;HALT INSTRUCTION
; 1286 CSL=2 ;CONSOLE HALT
; 1287 ;CODES 100 TO 777 ARE SOFTWARE ERRORS
; 1288 IOPF=100 ;I/O PAGE FAIL
; 1289 ILLII=101 ;ILLEGAL INTERRUPT INSTRUCTION
; 1290 ILLINT=102 ;BAD POINTER TO UNIBUS INTERRUPT VECTOR
; 1291 ;CODES 1000 TO 1777 ARE HARDWARE ERRORS
; 1292 BW14=1000 ;ILLEGAL BWRITE FUNCTION (BAD DROM)
; 1293 NICOND 5=1004 ;ILLEGAL NICOND DISPATCH
; 1294 MULERR=1005 ;VALUE COMPUTED FOR 10**21 WAS WRONG
; 1295 .IFNOT/FULL
; 1296 PAGEF=1777 ;PAGE FAIL IN SMALL MICROCODE
; 1297 .ENDIF/FULL
; 1298
; 1299
; 1300
; 1301 ;# FIELD REDEFINED WHEN USED AS FLG BITS
; 1302
; 1303 FLG.W/= <94> ;W BIT FROM PAGE MAP
; 1304 FLG.PI/= <95> ;PI CYCLE
; 1305 FLG.C/= <96> ;CACHE BIT FROM PAGE MAP
; 1306 FLG.SN/= <97> ;SPECIAL NEGATE IN FDV & DFDV
; 1307
; 1308 ;RIGHT HALF OF FLG USED TO RECOVER FROM PAGE FAILS
; 1309
```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 36
DISPATCH ROM DEFINITIONS

```
; 1310 .TOC "DISPATCH ROM DEFINITIONS"
; 1311
; 1312 ;ALL ON DPEA
; 1313
; 1314 .DCODE
; 1315 A/=<2:5> ;OPERAND FETCH MODE
; 1316 READ=0 ;READ
; 1317 WRITE=1 ;WRITE
; 1318 DREAD=2 ;DOUBLE READ
; 1319 DBLAC=3 ;DOUBLE AC
; 1320 SHIFT=4 ;SIMPLE SHIFT
; 1321 DSHIFT=5 ;DOUBLE SHIFT
; 1322 FPI=6 ;FLOATING POINT IMMEDIATE
; 1323 FP=7 ;FLOATING POINT
; 1324 RD-PF=10 ;READ, THEN START PREFETCH
; 1325 DFP=11 ;DOUBLE FLOATING POINT
; 1326 IOT=12 ;CHECK FOR IO LEGAL THEN SAME AS I
; 1327
; 1328 B/=<8:11> ;STORE RESULTS AS
; 1329 SELF=4 ;SELF
; 1330 DBLAC=5 ;DOUBLE AC
; 1331 DBLB=6 ;DOUBLE BOTH
; 1332 AC=15 ;AC
; 1333 MEM=16 ;MEMORY
; 1334 BOTH=17 ;BOTH
; 1335
; 1336 ;B-FIELD WHEN USED IN FLOATING POINT OPERATIONS
; 1337 ROUND/=<8> ;ROUND THE RESULT
; 1338 MODE/=<9> ;SEPARATE ADD/SUB & MUL/DIV ETC.
; 1339 FL-B/=<10:11> ;STORE RESULTS AS
; 1340 AC=1 ;AC
; 1341 MEM=2 ;MEMORY
; 1342 BOTH=3 ;BOTH
; 1343
; 1344 J/=<12:23> ;DISPATCH ADDRESS (MUST BE 1400 TO 1777)
; 1345
; 1346 ACDISP/=<24> ;DISPATCH ON AC FIELD
; 1347 I/=<25> ;IMMEDIATE DISPATCH. DISP/AREAD DOES A DISP/DROM
; 1348 ; IF THIS BIT IS SET.
; 1349 READ/=<26> ;START A READ AT AREAD
; 1350 TEST/=<27> ;START A WRITE TEST AT AREAD
; 1351 COND FUNC/=<28> ;START A MEMORY CYCLE ON BWRITE
; 1352 VMA/=<29>D,1 ;LOAD THE VMA ON AREAD
; 1353 WRITE/=<30> ;START A WRITE ON AREAD
; 1354 .UCODE
; 1355
```

; 1356 .TOC "HOW TO READ MACROS"
; 1357 ;
; 1358 ; 1.0 REGISTER TRANSFER MACROS
; 1359 ;
; 1360 ; MOST MACROS USED IN THE KS10 ARE USED TO OPERATE ON DATA IN (OR FROM/TO) 2901
; 1361 ; REGISTERS. THE NAMES OF THE 2901 REGISTERS ARE MACRO PARAMETERS AND ARE
; 1362 ; ENCLOSED IN []. A TYPICAL MACRO IS:
; 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 ;
; 1374 ; 1.1 SPECIAL SYMBOLS
; 1375 ;
; 1376 ; THERE ARE A BUNCH OF SYMBOLS USED IN THE MACROS WHICH ARE NOT 2901 REGISTERS.
; 1377 ; THEY ARE DEFINED HERE:
; 1378 ;
; 1379 ; 1. AC -- THE AC SELECTED BY THE CURRENT INSTRUCTION. SEE DPEA
; 1380 ;
; 1381 ; 2. AC[] -- AC+N. AC[1] IS AC+1, AC[2] IS AC+2, ETC.
; 1382 ;
; 1383 ; 3. APR -- THE APR FLAGS FROM DPMA
; 1384 ;
; 1385 ; 4. EA -- THE EFFECTIVE ADDRESS. THAT IS, 0 IN THE LEFT HALF AND THE
; 1386 ; CONTENTS OF THE HR IN THE RIGHT HALF.
; 1387 ;
; 1388 ; 5. EXP -- THE F.P. EXPONENT FROM THE SCAD. [AR]_EXP WILL TAKE THE
; 1389 ; EXPONENT OUT OF THE FE AND PUT IT BACK INTO THE NUMBER IN THE AR.
; 1390 ;
; 1391 ; 6. FE -- THE FE REGISTER
; 1392 ;
; 1393 ; 7. FLAGS -- THE PC FLAGS (FROM DPE9) IN THE LEFT HALF.
; 1394 ;
; 1395 ; 8. Q -- THE Q REGISTER
; 1396 ;
; 1397 ; 9. RAM -- THE RAM FILE, RAM ADDRESS IS IN THE VMA.
; 1398 ;
; 1399 ; 10. P -- THE P FIELD OF THE BYTE POINTER. SAME IDEA AS EXP.
; 1400 ;
; 1401 ; 11. TIME -- THE 1MS. TIMER
; 1402 ;
; 1403 ; 12. VMA -- THE VMA. WHEN READ IT INCLUDES THE VMA FLAGS
; 1404 ;
; 1405 ; 13. XR -- INDEX REGISTER
; 1406 ;

- ; 1407 ;
; 1408 ;
; 1409 ;
; 1410 ;
; 1411 ;
; 1412 ;
; 1413 ;
; 1414 ;
; 1415 ;
; 1416 ;
; 1417 ;
; 1418 ;
; 1419 ;
; 1420 ;
; 1421 ;
; 1422 ;
; 1423 ;
; 1424 ;
; 1425 ;
; 1426 ;
; 1427 ;
; 1428 ;
; 1429 ;
; 1430 ;
; 1431 ;
; 1432 ;
; 1433 ;
; 1434 ;
; 1435 ;
; 1436 ;
; 1437 ;
; 1438 ;
; 1439 ;
; 1440 ;
; 1441 ;
; 1442 ;
; 1443 ;
; 1444 ;
; 1445 ;
; 1446 ;
; 1447 ;
; 1448 ;
; 1449 ;
; 1450 ;
; 1451 ;
; 1452 ;
; 1453 ;
; 1454 ;
; 1455 ;
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 TO 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 EXCEPT NO [] ARE REQUIRED. THERE ARE ONLY A FEW SYMBOLS USED.

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

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 39
HOW TO READ MACROS

; 1456 ;
; 1457 ;
; 1458 ;
; 1459 ;
; 1460 ;
; 1461 ;
; 1462 ;
; 1463 ;
; 1464 ;
; 1465 ;
; 1466 ;
; 1467 ;
; 1468 ;
; 1469 ;
; 1470 ;
; 1471 ;
; 1472 ;
; 1473 ;
; 1474 ;
; 1475 ;
; 1476 ;
; 1477 ;
; 1478 ;
; 1479 ;
; 1480 ;
; 1481 ;
; 1482 ;
; 1483 ;
; 1484 ;
; 1485 ;
; 1486 ;
; 1487 ;
; 1488 ;
; 1489 ;
; 1490 ;
; 1491 ;
; 1492 ;
; 1493 ;
; 1494 ;
; 1495 ;
; 1496 ;
; 1497 ;
; 1498 ;
; 1499 ;
; 1500 ;
; 1501 ;
; 1502 ;
; 1503 ;
; 1504 ;
; 1505 ;
; 1506 ;

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.0 ;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

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

; 1507 ;
; 1508 ;
; 1509 ;
; 1510 ;
; 1511 ;
; 1512 ;
; 1513 ;
; 1514 ;

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

```

; 1515 .TOC "MACROS -- DATA PATH CHIP -- GENERAL"
; 1516
; 1517 .NOT.[ ] "AD/.NOT.A,A/@1"
; 1518 [ ]+[ ] "AD/A+B,A/@1,B/@2"
; 1519 [ ]-[ ] "AD/A-B-.25,A/@1,B/@2,ADD .25"
; 1520 [ ]-# "AD/A-D-.25,DBUS/DBM,DBM/#,A/@1,ADD .25"
; 1521 [ ] .AND.# "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1"
; 1522 [ ] .AND.Q "AD/A.AND.Q,A/@1,DEST/PASS"
; 1523 [ ] .AND.[ ] "AD/A.AND.B,A/@2,B/@1,DEST/PASS"
; 1524 [ ] .AND.NOT.[ ] "AD/.NOT.A.AND.B,A/@2,B/@1,DEST/PASS"
; 1525 [ ] .OR.[ ] "AD/A.OR.B,A/@2,B/@1,DEST/PASS"
; 1526 [ ] .XOR.# "AD/D.XOR.A,DBUS/DBM,DBM/#,A/@1"
; 1527 [ ] .XOR.[ ] "AD/A.XOR.B,A/@2,B/@1,DEST/PASS"
; 1528 [ ] #-[ ] "AD/D-A-.25,DEST/AD,A/@2,B/@1,DBUS/DBM,DBM/#,ADD .25"
; 1529 [ ] # "AD/D,DBUS/DBM,DBM/#,DEST/AD,B/@1"
; 1530 [ ] -1 "AD/-A-.25,A/ONE,DEST/AD,B/@1,ADD .25"
; 1531 [ ] -2 "AD/-A-.25,DEST/AD*2,A/ONE,B/@1,ADD .25"
; 1532 [ ] -Q "AD/-Q-.25,DEST/AD,B/@1,ADD .25"
; 1533 [ ] -Q*2 "AD/-Q-.25,DEST/AD*2,B/@1,ADD .25"
; 1534 [ ] -Q*.5 "AD/-Q-.25,DEST/AD*.5,B/@1,ADD .25"
; 1535 [ ] -[ ] "AD/-A-.25,A/@2,DEST/AD,B/@1,ADD .25"
; 1536 [ ] -[ ]-.25 "AD/-A-.25,A/@2,DEST/AD,B/@1"
; 1537 [ ] -[ ]*2 "AD/-A-.25,A/@2,DEST/AD*2,B/@1,ADD .25"
; 1538 [ ] .NOT.AC "AD/.NOT.D,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1539 [ ] .NOT.AC[ ] "AD/.NOT.D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,DT/3T"
; 1540 [ ] .NOT.Q "AD/.NOT.Q,DEST/AD,B/@1"
; 1541 [ ] .NOT.[ ] "AD/.NOT.A,A/@2,DEST/AD,B/@1"
; 1542 [ ] O "AD/ZERO,DEST/AD,B/@1"
; 1543 [ ] O*.5 LONG "AD/ZERO,DEST/Q Q*.5,B/@1"
; 1544 [ ] O XWD [ ] "AD/47,DEST/AD,B/@1,DBM/#,DBUS/DBM,./@2,RSRC/DA,A/MASK"
; 1545 [ ] AC "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,AD PARITY"
; 1546 [ ] -AC "AD/-D-.25,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1547 [ ] -AC[ ] "AD/-D-.25,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,ADD .25,DT/3T"
; 1548 [ ] AC*.5 "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD*.5,B/@1,DT/3T"
; 1549 [ ] AC*.5 LONG "AD/D,DBUS/RAM,RAMADR/AC#,DEST/Q Q*.5,B/@1,DT/3T"
; 1550 [ ] AC*2 "AD/D,DBUS/RAM,RAMADR/AC#,DEST/AD*2,B/@1,DT/3T"
; 1551 [ ] AC+1 "AD/D+A,DBUS/RAM,RAMADR/AC#,A/ONE,DEST/AD,B/@1"
; 1552 [ ] AC+1000001 "AD/D+A,DBUS/RAM,RAMADR/AC#,A/XWD1,DEST/AD,B/@1"
; 1553 [ ] AC+[ ] "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,DT/3T"
; 1554 [ ] AC-1 "AD/D-A-.25,DBUS/RAM,RAMADR/AC#,A/ONE,DEST/AD,B/@1,ADD .25"
; 1555 [ ] AC-[ ] "AD/D-A-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1556 [ ] AC-[ ]-.25 "AD/D-A-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1557 [ ] AC[ ]-[ ] "AD/D-A-.25,A/@3,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,ADD .25,DT/3T"
; 1558 [ ] 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"
; 1559 [ ] AC[ ] .AND.[ ] "AD/D.AND.A,A/@3,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,DT/3T"
; 1560 [ ] AC.AND.MASK "AD/D.AND.A,A/MASK,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,AD PARITY"
; 1561 [ ] AC[ ] "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD,B/@1,AD PARITY,DT/3T"
; 1562 [ ] AC[ ]*2 "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD*2,B/@1,AD PARITY,DT/3T"
; 1563 [ ] AC[ ]*.5 "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@2,DEST/AD*.5,B/@1,AD PARITY,DT/3T"
; 1564 [ ] APR "AD/D,DBUS/DBM,DBM/APR FLAGS,DEST/AD,B/@1,DT/3T"
; 1565 [ ] CURRENT AC [ ] "AD/D,DBUS/RAM,RAMADR/#,ACALU/B,ACN/@2,DEST/AD,B/@1,AD PARITY,DT/3T"
; 1566 [ ] EA FROM [ ] "AD/57,RSRC/OA,A/@2,DEST/AD,B/@1"
; 1567
    
```

```
; 1568 []_EA "AD/57,RSRC/OA,A/HR,DEST/AD,B/@1"
; 1569 []_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"
; 1570 []_FE "AD/D,DEST/AD*.5,B/@1,DBUS/DBM,DBM/DP,SCAD/A+B,SCADA/S#,S#/O,SCADB/FE,BYTE/BYTE5"
; 1571 []_FLAGS "AD/D.AND.A,DBUS/PC_FLAGS,A/MASK,DEST/AD,B/@1,RSRC/OQ"
; 1572 []_P "AD/D,DEST/A,A/@1,B/@1,DBUS/DBM,DBM/DP,BYTE/BYTE1,SCAD/A+B,SCADA/S#,S#/O,SCADB/FE"
; 1573 []_PC WITH FLAGS "AD/D,DBUS/PC_FLAGS,RSRC/OA,A/PC,DEST/AD,B/@1"
; 1574 []_Q "AD/Q,DEST/AD,B/@1"
; 1575 []_Q*.5 "AD/Q,DEST/AD*.5,B/@1"
; 1576 []_Q*2 "AD/Q,DEST/AD*2,B/@1"
; 1577 []_Q*2 LONG "AD/Q,DEST/Q_Q*2,B/@1"
; 1578 []_Q+1 "AD/A+Q,A/ONE,DEST/AD,B/@1"
; 1579 []_RAM "AD/D,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1,AD PARITY"
; 1580 []_TIME "AD/44,RSRC/DA,A/MASK,DBUS/DBM,DBM/EXP,DEST/AD,B/@1"
; 1581 []_VMA "AD/D,DEST/AD,B/@1,DBUS/DBM,DBM/VMA"
; 1582 []_XR "AD/D,DBUS/RAM,RAMADR/XR#,DEST/AD,B/@1"
; 1583 []_[] "AD/A,A/@2,DEST/AD,B/@1"
; 1584 []_[] SWAP "AD/D,DBUS/DBM,DBM/DP SWAP,DEST/A,A/@2,B/@1"
; 1585 []_[] XWD O "AD/45,DEST/AD,B/@1,DBM/#,DBUS/DBM,#/@2,RSRC/DO,A/MASK"
; 1586 []_[]*.5 "AD/A,A/@2,DEST/AD*.5,B/@1"
; 1587 []_[]*.5 LONG "AD/A,A/@2,DEST/Q_Q*.5,B/@1"
; 1588 []_[]*2 "AD/A,A/@2,DEST/AD*2,B/@1"
; 1589 []_[]*2 LONG "AD/A,A/@2,DEST/Q_Q*2,B/@1"
; 1590 []_[]*4 "AD/A+B,A/@2,B/@1,DEST/AD*2"
; 1591 []_[]+# "AD/D+A,DBUS/DBM,DBM/#,A/@2,DEST/AD,B/@1"
; 1592 []_[]+.25 "AD/O+A,A/@2,DEST/AD,B/@1,ADD .25"
; 1593 []_[]+O "AD/O+A,A/@2,DEST/AD,B/@1"
; 1594 []_[]+1 "AD/A+B,A/ONE,B/@1,B/@2,DEST/AD"
; 1595 []_[]+1000001 "AD/D+A,A/@2,DBUS/DBM,DBM/#,#/1,DEST/AD,B/@1"
; 1596 []_[]+AC "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1"
; 1597 []_[]+AC[] "AD/D+A,A/@2,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@3,DEST/AD,B/@1,DT/3T"
; 1598 []_[]+Q "AD/A+Q,A/@2,DEST/AD,B/@1"
; 1599 []_[]+RAM "AD/D+A,A/@2,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1"
; 1600 []_[]+XR "AD/D+A,DBUS/RAM,RAMADR/XR#,A/@2,DEST/AD,B/@1,HOLD LEFT"
; 1601 []_[]+[] "AD/A+B,A/@3,B/@1,B/@2,DEST/AD"
; 1602 []_[]+[]+.25 "AD/A+B,A/@3,B/@1,B/@2,DEST/AD,ADD .25"
; 1603 []_[]-# "AD/A-D-.25,DBUS/DBM,DBM/#,A/@2,DEST/AD,B/@1,ADD .25"
; 1604 []_[]-1 "AD/B-A-.25,B/@1,A/ONE,DEST/AD,ADD .25"
; 1605 []_[]-1000001 "AD/A-D-.25,A/@2,DBUS/DBM,DBM/#,#/1,DEST/AD,B/@1,ADD .25"
; 1606 []_[]-AC "AD/A-D-.25,A/@2,DBUS/RAM,RAMADR/AC#,DEST/AD,B/@1,ADD .25"
; 1607 []_[]-RAM "AD/A-D-.25,A/@2,DBUS/RAM,RAMADR/RAM,DEST/AD,B/@1,ADD .25"
; 1608 []_[]-[] "AD/B-A-.25,B/@1,B/@2,A/@3,DEST/AD,ADD .25"
; 1609 []_[]-[] REV "AD/A-B-.25,B/@1,B/@3,A/@2,DEST/AD,ADD .25"
; 1610 []_[]-AND.# "AD/D.AND.A,DBUS/DBM,DBM/#,DEST/AD,A/@2,B/@1"
; 1611 []_[]-AND.# CLR LH "AD/ZERO,RSRC/DA,DBUS/DBM,DBM/#,DEST/AD,A/@2,B/@1"
; 1612 []_[]-AND.# CLR RH "AD/D.AND.A,RSRC/OQ,DBUS/DBM,DBM/#,DEST/AD,A/@2,B/@1"
; 1613 []_[]-AC[]*.5 "AD/D.AND.A,DEST/AD*.5,A/@3,B/@1,RAMADR/AC*#,DBUS/RAM,ACALU/AC+N,ACN/@2"
; 1614 []_[]-(Q+1)*.5 "AD/A+Q,A/ONE,DEST/AD*.5,B/@1"
; 1615 []_[]-[#-[]]*2 "AD/D-A-.25,DEST/AD*2,A/@2,B/@1,DBUS/DBM,DBM/#,ADD .25"
; 1616 []_[]-[-[]]*.5 "AD/-A-.25,A/@2,DEST/AD*.5,B/@1,ADD .25"
; 1617 []_[]-[-.25]*.5 LONG "AD/-A-.25,A/@2,DEST/Q_Q*.5,B/@1"
; 1618 []_[]-[-.25]*2 LONG "AD/-A-.25,A/@2,DEST/Q_Q*2,B/@1"
; 1619
```

```

; 1620 [ ]_([ ]_AND.#)*.5 "AD/D.AND.A,DBUS/DBM,DBM/#,DEST/AD*.5,A/@2,B/@1"
; 1621 [ ]_([ ]_AND.#)*2 "AD/D.AND.A,DBUS/DBM,DBM/#,DEST/AD*2,A/@2,B/@1"
; 1622 [ ]_([ ]_AND.NOT.#)*.5 "AD/.NOT.D.AND.A,DBUS/DBM,DBM/#,DEST/AD*.5,A/@2,B/@1"
; 1623 [ ]_([ ]_AND.NOT.#)*2 "AD/.NOT.D.AND.A,DBUS/DBM,DBM/#,DEST/AD*2,A/@2,B/@1"
; 1624 [ ]_([ ]_AND.[ ])*.5 "AD/A.AND.B,DEST/AD*.5,A/@3,B/@1,B/@2"
; 1625 [ ]_([ ]_AND.[ ])*2 "AD/A.AND.B,DEST/AD*2,A/@3,B/@1,B/@2"
; 1626 [ ]_([ ]_OR.#)*.5 "AD/D.OR.A,DBUS/DBM,DBM/#,DEST/AD*.5,A/@2,B/@1"
; 1627 [ ]_([ ]_OR.#)*2 "AD/D.OR.A,DBUS/DBM,DBM/#,DEST/AD*2,A/@2,B/@1"
; 1628 [ ]_([ ]_+#)*2 "AD/D+A,DBUS/DBM,DBM/#,DEST/AD*2,A/@2,B/@1"
; 1629 [ ]_([ ]_+1)*2 "AD/A+B,A/ONE,B/@1,B/@2,DEST/AD*2"
; 1630 [ ]_([ ]_+ [ ])*.5 LONG "AD/A+B,A/@3,B/@1,B/@2,DEST/Q_Q*.5"
; 1631 [ ]_([ ]_+ [ ])*2 LONG "AD/A+B,A/@3,B/@1,B/@2,DEST/Q_Q*2"
; 1632 [ ]_([ ]_- [ ])*.5 LONG "AD/B-A-.25,A/@3,B/@1,B/@2,DEST/Q_Q*.5,ADD .25"
; 1633 [ ]_([ ]_- [ ])*2 LONG "AD/B-A-.25,A/@3,B/@1,B/@2,DEST/Q_Q*2,ADD .25"
; 1634 [ ]_([ ]_+ [ ]_+.25)*.5 LONG "AD/A+B,A/@3,B/@1,B/@2,DEST/Q_Q*.5,ADD .25"
; 1635 [ ]_([ ]_AND.AC "AD/D.AND.A,DBUS/RAM,RAMADR/AC#,A/@2,DEST/AD,B/@1"
; 1636 [ ]_([ ]_AND.NOT.# "AD/.NOT.D.AND.A,DBUS/DBM,DBM/#,A/@2,DEST/AD,B/@1"
; 1637 [ ]_([ ]_AND.NOT.[ ] "AD/.NOT.A.AND.B,DEST/AD,B/@1,B/@2,A/@3"
; 1638 [ ]_([ ]_AND.NOT.AC "AD/.NOT.D.AND.A,DBUS/RAM,RAMADR/AC#,A/@2,DEST/AD,B/@1"
; 1639 [ ]_([ ]_AND.Q "AD/A.AND.Q,A/@2,DEST/AD,B/@1"
; 1640 [ ]_([ ]_AND.[ ] "AD/A.AND.B,A/@3,B/@1,B/@2,DEST/AD"
; 1641 [ ]_([ ]_EQV.AC "AD/D.EQV.A,DBUS/RAM,RAMADR/AC#,A/@2,DEST/AD,B/@1"
; 1642 [ ]_([ ]_EQV.Q "AD/A.EQV.Q,A/@2,DEST/AD,B/@1"
; 1643 [ ]_([ ]_OR.# "AD/D.OR.A,DBUS/DBM,DBM/#,A/@2,DEST/AD,B/@1"
; 1644 [ ]_([ ]_OR.AC "AD/D.OR.A,DBUS/RAM,RAMADR/AC#,A/@2,DEST/AD,B/@1"
; 1645 [ ]_([ ]_OR.FLAGS "AD/D.OR.A,DBUS/PC FLAGS,RSRC/OA,A/@1,DEST/AD,B/@1"
; 1646 [ ]_([ ]_OR.[ ] "AD/A.OR.B,A/@3,B/@2,B/@1,DEST/AD"
; 1647 [ ]_([ ]_XOR.# "AD/D.XOR.A,DBUS/DBM,DBM/#,DEST/AD,A/@2,B/@1"
; 1648 [ ]_([ ]_XOR.AC "AD/D.XOR.A,DBUS/RAM,RAMADR/AC#,A/@1,DEST/AD,B/@2"
; 1649 [ ]_([ ]_XOR.[ ] "AD/A.XOR.B,A/@3,B/@1,B/@2,DEST/AD"
; 1650
; 1651 [ ] LEFT_O "AD/57,RSRC/OB,DEST/AD,B/@1"
; 1652 [ ] RIGHT_O "AD/53,RSRC/DO,DEST/AD,B/@1"
; 1653 [ ] LEFT_-1 "AD/54,RSRC/OB,DEST/AD,A/MASK,B/@1"
; 1654 [ ] RIGHT_-1 "AD/53,RSRC/OA,DEST/AD,A/MASK,B/@1"
; 1655
; 1656
; 1657 [ ]_+SIGN "[@1]_[@1].AND.#, #/777, HOLD RIGHT"
; 1658 [ ]_-SIGN "[@1]_[@1].OR.#, #/777000, HOLD RIGHT"
; 1659 ;THE FOLLOWING 2 MACROS ARE USED IN DOUBLE FLOATING STUFF
; 1660 ; THEY ASSUME THAT THE OPERAND HAS BEEN SHIFTED RIGHT 1 PLACE.
; 1661 ; THEY SHIFT 1 MORE PLACE
; 1662 [ ]_+SIGN*.5 "AD/.NOT.D.AND.A,A/@1,B/@1,DEST/AD*.5,DBUS/DBM,DBM/#,#/777400,RSRC/OA"
; 1663 [ ]_-SIGN*.5 "AD/D.OR.A,A/@1,B/@1,DEST/AD*.5,DBUS/DBM,DBM/#,#/777400,RSRC/OA"
; 1664
    
```

Produced on Advanced Information Electronic Laser Printer, PK01E56, DTN: 223-7891

```
; 1665 .TOC "MACROS -- DATA PATH CHIP -- Q"
; 1666
; 1667 Q-[ ] "AD/Q-A-.25,A/@1,ADD .25"
; 1668 Q.AND.NOT.[ ] "AD/.NOT.A.AND.Q,A/@1,DEST/PASS"
; 1669 Q[ ] "AD/A,DEST/Q_AD,A/@1"
; 1670 Q[ ]-[ ] "AD/A-B-.25,A/@1,B/@2,DEST/Q_AD,ADD .25"
; 1671 Q[ ]+[ ] "AD/A+B,A/@1,B/@2,DEST/Q_AD"
; 1672 Q[ ] .AND.[ ] "AD/A.AND.B,A/@1,B/@2,DEST/Q_AD"
; 1673 Q_.NOT.AC[ ] "AD/.NOT.D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,DT/3T"
; 1674 Q_-[ ] "AD/-A-.25,DEST/Q_AD,A/@1,ADD .25"
; 1675 Q_-1 "Q_-[ONE]"
; 1676 Q_-AC[ ] "AD/-D-.25,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,ADD .25,DT/3T"
; 1677 Q_-Q "AD/-Q-.25,ADD .25,DEST/Q_AD"
; 1678 Q_AC "AD/D,DBUS/RAM,RAMADR/AC#,DEST/Q_AD,CHK PARITY"
; 1679 Q_AC[ ] "AD/D,DBUS/RAM,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DEST/Q_AD,CHK PARITY,DT/3T"
; 1680 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"
; 1681 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"
; 1682 Q_.NOT.Q "AD/.NOT.Q,DEST/Q_AD"
; 1683 Q_# "AD/D,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1684 Q_0 "AD/ZERO,DEST/Q_AD"
; 1685 Q_0 XWD [ ] "AD/47,DEST/Q_AD,DBM/#,DBUS/DBM,#/@1,RSRC/DA,A/MASK"
; 1686 Q_Q+.25 "AD/Q+.25,DEST/Q_AD,ADD .25"
; 1687 Q_Q+1 "AD/A+Q,A/ONE,DEST/Q_AD"
; 1688 Q_Q-1 "AD/Q-A-.25,A/ONE,DEST/Q_AD,ADD .25"
; 1689 Q_Q+AC "AD/D+Q,DBUS/RAM,RAMADR/AC#,DEST/Q_AD"
; 1690 Q_Q*.5 "[MAG][MASK]*.5 LONG,SHSTYLE/NORM"
; 1691 Q_Q*2 "[MASK][MAG]*2 LONG,SHSTYLE/NORM"
; 1692 Q_Q.OR.# "AD/D.OR.Q,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1693 Q_Q.AND.# "AD/D.AND.Q,DBUS/DBM,DBM/#,DEST/Q_AD"
; 1694 Q_Q.AND.[ ] "AD/A.AND.Q,A/@1,DEST/Q_AD"
; 1695 Q_Q.AND.NOT.[ ] "AD/.NOT.A.AND.Q,A/@1,DEST/Q_AD"
; 1696 Q_Q+[ ] "AD/A+Q,A/@1,DEST/Q_AD"
; 1697 Q_[ ] .AND.Q "AD/A.AND.Q,A/@1,DEST/Q_AD"
; 1698 Q_[ ] .OR.Q "AD/A.OR.Q,A/@1,DEST/Q_AD"
; 1699
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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.

```
; 1700 .TOC "MACROS -- DATA PATH CHIP -- MISC."  
; 1701  
; 1702 CLEAR []O "AD/D.AND.A,A/@1,DBUS/DBM,DBM/#,#/377777,DEST/AD,B/@1,HOLD RIGHT"  
; 1703 CLEAR ARXO "CLEAR [ARX]O"  
; 1704  
; 1705 ;CYCLE CHIP REGISTERS THRU AD SO WE CAN TEST BITS  
; 1706 READ XR "AD/D,DBUS/RAM,RAMADR/XR#"  
; 1707 READ [] "AD/B,B/@1"  
; 1708 READ Q "AD/Q"  
; 1709  
; 1710 ;TEST BITS IN REGISTERS (SKIP IF ZERO)  
; 1711 TR [] "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1,SKIP ADR.EQ.O,DT/3T"  
; 1712 TL [] "AD/D.AND.A,DBUS/DBM,DBM/#,A/@1,SKIP ADL.EQ.O,DT/3T"  
; 1713  
; 1714  
; 1715 ;CAUSE BITS -2 AND -1 TO MATCH BIT 0.  
; 1716 FIX [] SIGN "AD/D,DEST/A,A/@1,B/@1,DBUS/DP,HOLD RIGHT"  
; 1717  
; 1718 ;GENERATE A MASK IN Q AND ZERO A 2901 REGISTER  
; 1719 GEN MSK [] "AD/ZERO,DEST/Q_Q*2,B/@1,ONES"  
; 1720
```

```
; 1721 .TOC "MACROS -- STORE IN AC"
; 1722
; 1723 FM WRITE "FMWRITE/1"
; 1724
; 1725 AC[] [] VIA AD "AD/B,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,CHK PARITY"
; 1726 AC[] [] VIA AD "AD/B,DEST/PASS,B/@1,RAMADR/AC#,DBUS/DP,FM WRITE,CHK PARITY"
; 1727 AC[] [] "AD/A,DEST/A,B/@2,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1728 AC[] [] TEST "AD/D,DBUS/DP,DEST/A,B/@2,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1729 AC[] []+1 "AD/A+B,DEST/PASS,A/ONE,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1730 AC[] []*2 "AD/A+B,DEST/PASS,A/@2,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1731 AC[] [] "AD/A,DEST/A,B/@1,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1732 AC[] [] TEST "AD/D,DBUS/DP,DEST/A,B/@1,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1733 AC[] []+1 "AD/A+B,DEST/PASS,A/ONE,B/@1,RAMADR/AC#,FM WRITE"
; 1734 AC[] []+Q "AD/A+Q,DEST/PASS,A/@1,B/@1,RAMADR/AC#,FM WRITE"
; 1735 AC[] []+Q "AD/A+Q,DEST/PASS,A/@2,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,FM WRITE"
; 1736 AC[] []-[] "AD/A-B-.25,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1737 AC[] []+[] "AD/A+B,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1738 AC[] []+[] "AD/A+B,DEST/PASS,B/@2,A/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1739 AC[] [] .AND. [] "AD/A.AND.B,DEST/PASS,B/@3,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1740 AC[] [] .Q.AND. [] "AD/A.AND.Q,DEST/PASS,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1741 AC[] [] .EQV.Q "AD/A.EQV.Q,DEST/PASS,A/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1742 AC[] []-[] "AD/-B-.25,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1743 AC[] []-[] "AD/-A-.25,DEST/PASS,A/@1,RAMADR/AC#,DBUS/DP,ADD .25,FM WRITE"
; 1744 AC[] [] .OR. [] "AD/A.OR.B,A/@1,B/@2,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1745 AC[] [] .NOT. [] "AD/.NOT.B,DEST/PASS,B/@2,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1746 AC[] [] .NOT. [] "AD/.NOT.B,DEST/PASS,B/@1,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1747 AC[] []-Q "AD/-Q-.25,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE,ADD .25"
; 1748 AC[] [] Q "AD/Q,RAMADR/AC#,DBUS/DP,FM WRITE"
; 1749 AC[] []_O "AD/ZERO,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1750 AC[] []_1 "AD/B,DEST/PASS,B/ONE,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1751 AC[] []_Q "AD/Q,RAMADR/AC*#,ACALU/AC+N,ACN/@1,DBUS/DP,FM WRITE"
; 1752
```

```
; 1753 .TOC "MACROS -- MICROCODE WORK SPACE"
; 1754
; 1755
; 1756 WORK[.]_Q "AD/Q,DEST/PASS,RAMADR/#,WORK/@1,FM WRITE"
; 1757 Q_WORK[ ] "AD/D,DEST/Q_AD,RAMADR/#,DBUS/RAM,WORK/@1,DT/3T"
; 1758 WORK[ ]_O "AD/ZERO,DEST/PASS,RAMADR/#,WORK/@1,FM WRITE"
; 1759 WORK[ ]_1 "AD/B,DEST/PASS,RAMADR/#,WORK/@1,B/ONE,FM WRITE"
; 1760 WORK[ ]_[] "AD/B,DEST/PASS,RAMADR/#,WORK/@1,B/@2,FM WRITE"
; 1761 WORK[ ]_[] CLR LH "AD/47,RSRC/AB,DEST/PASS,RAMADR/#,WORK/@1,B/@2,A/MASK,FM WRITE"
; 1762 WORK[ ]_[]-1 "AD/A-B-.25,A/@2,B/ONE,DEST/PASS,RAMADR/#,WORK/@1,FM WRITE, ADD .25"
; 1763 WORK[ ]_[] .NOT. [ ] "AD/.NOT.B,DEST/PASS,RAMADR/#,WORK/@1,B/@2,FM WRITE"
; 1764 WORK[ ]_[] .AND. [ ] "AD/A.AND.B,DEST/PASS,RAMADR/#,WORK/@1,A/@2,B/@3,FM WRITE"
; 1765 [ ]_ .AND. NOT. WORK[ ] "AD/.NOT.D.AND.A,A/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1766 [ ]_ .AND. WORK[ ] "AD/D.AND.A,A/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1767 [ ]_ []+WORK[ ] "AD/D+A,A/@2,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@3,DT/3T"
; 1768 [ ]_ [] .AND. WORK[ ] "AD/D.AND.A,A/@2,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@3,DT/3T"
; 1769 [ ]_ [] .AND. NOT. WORK[ ] "AD/.NOT.D.AND.A,A/@2,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@3,DT/3T"
; 1770 [ ]_ [] .OR. WORK[ ] "AD/D.OR.A,A/@2,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@3,DT/3T"
; 1771 [ ]_ WORK[ ] "AD/D,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1772 [ ]_ .NOT. WORK[ ] "AD/.NOT.D,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1773 [ ]_ -WORK[ ] "AD/-D-.25,ADD .25,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1774 [ ]_ WORK[ ]+1 "AD/D+A,A/ONE,DEST/AD,B/@1,DBUS/RAM,RAMADR/#,WORK/@2,DT/3T"
; 1775 Q_Q-WORK[ ] "AD/Q-D-.25,DEST/Q_AD,DBUS/RAM,RAMADR/#,WORK/@1,ADD .25,DT/3T"
; 1776 [ ]_ []-WORK[ ] "AD/A-D-.25,DEST/AD,A/@2,B/@1,DBUS/RAM,RAMADR/#,WORK/@3,ADD .25,DT/3T"
; 1777
; 1778 RAM_[ ] "AD/B,DEST/PASS,RAMADR/RAM,B/@1,FM WRITE"
; 1779
```



```
; 1780 .TOC "MACROS -- MEMORY CONTROL"
; 1781
; 1782 MEM CYCLE "MEM/1"
; 1783
; 1784 ;THE FOLLOWING MACROS CONTROL MEMORY ADDRESS
; 1785 LOAD VMA "MEM CYCLE,LDVMA/1"
; 1786 FORCE EXEC "FORCE EXEC/1"
; 1787 VMA PHYSICAL "PHYSICAL/1,FORCE EXEC/1,FORCE USER/O,EXT ADR/1,LOAD VMA"
; 1788 VMA PHYSICAL WRITE "LOAD VMA,VMA PHYSICAL,WAIT/1,MEM/1,WRITE CYCLE/1,WRITE TEST/O"
; 1789 VMA PHYSICAL READ "LOAD VMA,VMA PHYSICAL,WAIT/1,MEM/1,READ CYCLE/1,WRITE TEST/O"
; 1790 VMA EXTENDED "EXT ADR/1"
; 1791
; 1792 PXCT EA "PXCT/E1"
; 1793 PXCT DATA "PXCT/D1"
; 1794 PXCT BLT DEST "PXCT/D1"
; 1795 PXCT BYTE PTR EA "PXCT/E2"
; 1796 PXCT BYTE DATA "PXCT/D2"
; 1797 PXCT STACK WORD "PXCT/D2"
; 1798 PXCT BLT SRC "PXCT/D2"
; 1799 PXCT EXTEND EA "PXCT/E2"
; 1800
; 1801 ;THE FOLLOWING MACROS GET MEMORY CYCLES STARTED
; 1802 WRITE TEST "WRITE TEST/1,WAIT/1"
; 1803 START READ "MEM CYCLE,READ CYCLE/1,WAIT/1"
; 1804 START WRITE "MEM CYCLE,WRITE TEST,WRITE CYCLE/1,WAIT/1"
; 1805 START NO TEST WRITE "MEM CYCLE,WRITE CYCLE/1,WAIT/1"
; 1806 FETCH "START READ,FETCH/1,PXCT/CURRENT,WAIT/1"
; 1807
; 1808 ;THE FOLLOWING MACROS COMPLETE MEMORY CYCLES
; 1809 MEM WAIT "MEM CYCLE,WAIT/1"
; 1810 MEM READ "MEM WAIT,DBUS/DBM,DBM/MEM"
; 1811 MEM WRITE "MEM WAIT,DT/3T"
; 1812 SPEC MEM READ "SPEC/WAIT,DBUS/DBM,DBM/MEM"
; 1813 SPEC MEM WRITE "SPEC/WAIT,DT/3T"
; 1814
; 1815
; 1816 ;THINGS WHICH WRITE MEMORY
; 1817 MEM_[] "AD/B,DEST/PASS,B/@1,DBUS/DP,RAMADR/VMA,CHK PARITY"
; 1818 MEM_Q "AD/Q,DBUS/DP,RAMADR/VMA"
; 1819
; 1820
; 1821 ;THINGS WHICH READ MEMORY
; 1822 []_IO DATA "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1"
; 1823 []_MEM "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1,CHK PARITY"
; 1824 []_MEM THEN FETCH "AD/D,DBUS/DBM,RAMADR/VMA,DEST/A,A/PC,B/@1,CHK PARITY, FETCH, LOAD VMA"
; 1825 []_MEM*.5 "AD/D,DBUS/DBM,RAMADR/VMA,DEST/AD*.5,B/@1,CHK PARITY"
; 1826 []_MEM.AND.MASK "AD/D.AND.A,A/MASK,DBUS/DBM,RAMADR/VMA,DEST/AD,B/@1,CHK PARITY"
; 1827 []_(MEM.AND.[])*.5 "AD/D.AND.A,A/@2,DBUS/DBM,RAMADR/VMA,DEST/AD*.5,B/@1,CHK PARITY"
; 1828 Q_MEM "AD/D,DBUS/DBM,RAMADR/VMA,DEST/Q_AD,CHK PARITY"
; 1829
```

```
; 1830 .TOC "MACROS -- VMA"
; 1831
; 1832 VMA_[] "AD/A,A/@1,DEST/PASS,LOAD VMA"
; 1833 VMA_[] WITH FLAGS "AD/A,A/@1,DEST/PASS,LOAD VMA,WAIT/1, MEM/1, EXT ADR/1, DP FUNC/1, DT/3T"
; 1834 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"
; 1835 VMA_[]+1 "AD/A+B,A/ONE,B/@1,DEST/AD,HOLD LEFT,LOAD VMA"
; 1836 VMA_[]-1 "AD/B-A-.25,A/ONE,B/@1,ADD .25,HOLD LEFT,LOAD VMA"
; 1837 VMA_[]+XR "AD/D+A,DBUS/RAM,RAMADR/XR#,A/@1,LOAD VMA"
; 1838 VMA_[]+[] "AD/A+B,DEST/PASS,A/@1,E/@2,LOAD VMA"
; 1839
; 1840 NEXT [] PHYSICAL WRITE "AD/A+B,A/ONE,B/@1,DEST/AD,HOLD LEFT,LOAD VMA, VMA PHYSICAL, START WRITE"
; 1841
; 1842 ;MACROS TO LOAD A 2901 REGISTER WITH VMA FLAG BITS
; 1843 []_VMA FLAGS "AD/45,DEST/AD,B/@1,DBM/#,DBUS/DBM,RSRC/DO,A/MASK"
; 1844 []_VMA IO READ "[@1]_VMA FLAGS,READ CYCLE/1,IO CYCLE/1,WRITE TEST/O, PHYSICAL/1, FORCE EXEC/1, FORCE USER/O"
; 1845 []_VMA IO WRITE "[@1]_VMA FLAGS,WRITE CYCLE/1,IO CYCLE/1,WRITE TEST/O, PHYSICAL/1, FORCE EXEC/1, FORCE USER/O"
; 1846
```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1847 .TOC      "MACROS -- TIME CONTROL"  
; 1848  
; 1849 AC        "RAMADR/AC#"  
; 1850 AC[]     "RAMADR/AC*#,ACALU/AC+N,ACN/@1"  
; 1851 XR       "RAMADR/XR#"  
; 1852 VMA      "RAMADR/VMA"  
; 1853 WORK[]   "RAMADR/#, WORK/@1"  
; 1854  
; 1855 2T       "T/2T"  
; 1856 3T       "T/3T"  
; 1857 4T       "T/4T"  
; 1858 5T       "T/5T"  
; 1859
```

```
; 1860 .TOC "MACROS -- SCAD, SC, FE LOGIC"
; 1861
; 1862 LOAD SC "LOADSC/1"
; 1863 LOAD FE "LOADFE/1"
; 1864 STEP SC "SCAD/A-1, SCADA/SC, LOAD SC, SKIP/SC"
; 1865 SHIFT "SCAD/A+B, SCADA/S#, SCADB/FE, S#/1, LOAD FE, MULTI SHIFT/1"
; 1866
; 1867 SC_SC-1 "SCAD/A-1, SCADA/SC, LOAD SC"
; 1868 SC_SHIFT "SCAD/A+B, SCADA/S#, S#/O, SCADB/SHIFT, LOAD SC"
; 1869 SC_SHIFT-1 "SCAD/A+B, SCADA/S#, S#/1777, SCADB/SHIFT, LOAD SC"
; 1870 SC_SHIFT-2 "SCAD/A+B, SCADA/S#, S#/1776, SCADB/SHIFT, LOAD SC"
; 1871 SC_-SHIFT "SCAD/A-B, SCADA/S#, S#/O000, SCADB/SHIFT, LOAD SC"
; 1872 SC_-SHIFT-1 "SCAD/A-B, SCADA/S#, SCADB/SHIFT, S#/1777, LOAD SC"
; 1873 SC_-SHIFT-2 "SCAD/A-B, SCADA/S#, SCADB/SHIFT, S#/1776, LOAD SC"
; 1874 SC_SC-EXP "SCAD/A-B, SCADA/SC, SCADB/EXP, LOAD SC"
; 1875 SC_SC-EXP-1 "SCAD/A-B-1, SCADA/SC, SCADB/EXP, LOAD SC"
; 1876 SC_SC-FE-1 "SCAD/A-B-1, SCADA/SC, SCADB/FE, LOAD SC"
; 1877 SC_SC-FE "SCAD/A-B, SCADA/SC, SCADB/FE, LOAD SC"
; 1878 SC_EXP "SCAD/A+B, SCADA/S#, S#/O, SCADB/EXP, LOAD SC"
; 1879 SC_S#-FE "SCAD/A-B, SCADA/S#, SCADB/FE, LOAD SC"
; 1880 SC_FE+S# "SCAD/A+B, SCADA/S#, SCADB/FE, LOAD SC"
; 1881 SC_FE "SCAD/A.OR.B, SCADA/S#, S#/O, SCADB/FE, LOAD SC"
; 1882 SC_S# "SCAD/A, SCADA/S#, LOAD SC"
; 1883
; 1884
; 1885 SC_36. "SC_S#, S#/36."
; 1886 SC_35. "SC_S#, S#/35."
; 1887 SC_34. "SC_S#, S#/34."
; 1888 SC_28. "SC_S#, S#/28."
; 1889 SC_27. "SC_S#, S#/27."
; 1890 SC_26. "SC_S#, S#/26."
; 1891 SC_24. "SC_S#, S#/24."
; 1892 SC_22. "SC_S#, S#/22."
; 1893 SC_20. "SC_S#, S#/20."
; 1894 SC_19. "SC_S#, S#/19."
; 1895 SC_14. "SC_S#, S#/14."
; 1896 SC_11. "SC_S#, S#/11."
; 1897 SC_9. "SC_S#, S#/9."
; 1898 SC_8. "SC_S#, S#/8."
; 1899 SC_7. "SC_S#, S#/7."
; 1900 SC_6. "SC_S#, S#/6."
; 1901 SC_5. "SC_S#, S#/5."
; 1902 SC_4. "SC_S#, S#/4."
; 1903 SC_3. "SC_S#, S#/3."
; 1904 SC_2. "SC_S#, S#/2."
; 1905 SC_1. "SC_S#, S#/1."
; 1906 SC_0. "SC_S#, S#/O."
; 1907 SC_-1. "SC_S#, S#/1777."
; 1908 SC_-2. "SC_S#, S#/1776."
; 1909
```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1910 FE_-FE "SCAD/A-B,SCADA/S#,S#/O,SCADB/FE,LOAD FE"
; 1911 FE_-FE-1 "SCAD/A-B,SCADA/S#,S#/1777,SCADB/FE,LOAD FE"
; 1912 FE_FE-19 "SCAD/A+B,SCADB/FE,SCADA/S#,S#/1550,LOAD FE"
; 1913 FE_-FE+S# "SCAD/A-B,SCADA/S#,SCADB/FE,LOAD FE"
; 1914 FE_FE+SC "SCAD/A+B,SCADA/SC,SCADB/FE,LOAD FE"
; 1915 FE_FE.AND.S# "SCAD/A.AND.B,SCADA/S#,SCADB/FE,LOAD FE"
; 1916 FE_P "SCAD/A,SCADA/BYTE1,LOAD FE"
; 1917 FE_S "SCAD/A+B,SCADA/S#,S#/O,SCADB/SIZE,LOAD FE"
; 1918 FE_S+2 "SCAD/A+B,SCADA/S#,S#/20,SCADB/SIZE,LOAD FE"
; 1919 FE_-S-20 "SCAD/A-B,SCADA/S#,S#/1760,SCADB/SIZE,LOAD FE"
; 1920 FE_-S-10 "SCAD/A-B,SCADA/S#,S#/1770,SCADB/SIZE,LOAD FE"
; 1921 FE_S# "SCAD/A,SCADA/S#,LOAD FE"
; 1922 FE_S#-FE "SCAD/A-B,SCADA/S#,SCADB/FE,LOAD FE"
; 1923 FE_-2 "FE_S#,S#/1776"
; 1924 FE_-12 "FE_S#,S#/1764"
; 1925 FE_O "FE_S#,S#/O"
; 1926 FE_-1 "FE_S#,S#/1777"
; 1927 FE_FE+1 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/1,LOAD FE"
; 1928 FE_FE+2 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/2,LOAD FE"
; 1929 FE_FE+10 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/10,LOAD FE"
; 1930 FE_FE-1 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/1777,LOAD FE"
; 1931 FE_FE+4 "SCAD/A+B,SCADA/S#,SCADB/FE,S#/4,LOAD FE"
; 1932 FE_EXP "SCAD/A+B,SCADA/S#,S#/O,SCADB/EXP,LOAD FE"
; 1933 FE_SC+EXP "SCAD/A+B,SCADA/SC,SCADB/EXP,LOAD FE"
; 1934 FE_SC-EXP "SCAD/A-B,SCADA/SC,SCADB/EXP,LOAD FE"
; 1935 FE_FE+P "SCAD/A+B,SCADA/BYTE1,SCADB/FE,LOAD FE"
; 1936 FE_FE-200 "SCAD/A+B,SCADA/S#,S#/1600,SCADB/FE,LOAD FE"
; 1937 FE_-FE+200 "SCAD/A-B,SCADA/S#,S#/200,SCADB/FE,LOAD FE"
; 1938 FE_FE+S# "SCAD/A+B,SCADA/S#,SCADB/FE,LOAD FE"
; 1939
; 1940
; 1941 GEN 17-FE "SCAD/A-B,SCADA/S#,S#/210,SCADB/FE"
; 1942
```

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

; 1943 .TOC "MACROS -- DATA PATH FIELD CONTROL"
; 1944
; 1945 HOLD LEFT "CLKL/O,GENL/O"
; 1946 ADL PARITY "GENL/1"
; 1947 CHK PARITY L "CHKL/1"
; 1948
; 1949 HOLD RIGHT "CLKR/O,GENR/O"
; 1950 ADR PARITY "GENR/1"
; 1951 CHK PARITY R "CHKR/1"
; 1952
; 1953 AD PARITY "AD PARITY OK/1"
; 1954 CHK PARITY "CHKL/1,CHKR/1"
; 1955 BAD PARITY "CHKL/O,CHKR/O"
; 1956
; 1957 INH CRY18 "SPEC/INHCRY18"
; 1958

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 1959 .TOC      "MACROS -- SHIFT PATH CONTROL"  
; 1960  
; 1961 ASH      "SHSTYLE/NORM" ;ASH SHIFT  
; 1962 LSH      "SHSTYLE/NORM" ;LSH SHIFT (SAME HARDWARE AS ASH BUT  
; 1963          ; BITS -2 AND -1 ARE PRESET TO ZERO)  
; 1964 ROT      "SHSTYLE/ROT"  
; 1965 LSHC     "SHSTYLE/LSHC"  
; 1966 ASHC     "SHSTYLE/ASHC"  
; 1967 ROTC     "SHSTYLE/ROTC"  
; 1968 ONES     "SHSTYLE/ONES" ;SHIFT IN 1 BITS  
; 1969 DIV      "SHSTYLE/DIV"  ;SPECIAL PATH FOR DIVIDE (LIKE ROTC BUT  
; 1970          ; COMPLEMENT BIT AS IT GOES AROUND)  
; 1971
```

```
; 1972 .TOC "MACROS -- SPECIAL FUNCTIONS"
; 1973
; 1974 LOAD IR "SPEC/LOADIR" ;LOAD INSTRUCTION REG FROM
; 1975 ; DBUS0-DBUS8, LOAD AC# FROM
; 1976 ; DBUS9-DBUS12
; 1977 ; UPDATE LAST-INST-PUBLIC PC FLAG
; 1978 LOAD INST "SPEC/LDINST"
; 1979 LOAD INST EA "SPEC/LOADXR,PXCT/CURRENT"
; 1980 LOAD BYTE EA "SPEC/LOADXR,PXCT/E2"
; 1981 LOAD IND EA "SPEC/LOADXR,PXCT/E1"
; 1982 LOAD SRC EA "SPEC/LOADXR,PXCT/BIS-SRC-EA"
; 1983 LOAD DST EA "SPEC/LOADXR,PXCT/BIS-DST-EA"
; 1984 ADD .25 "CRY38/1" ;GENERATE CARRY IN TO BIT 37
; 1985 CALL [] "CALL/1,J/@1" ;CALL A SUBROUTINE
; 1986 LOAD PXCT "SPEC/LDPXCT" ;LOAD PXCT FLAGS IF EXEC MODE
; 1987 TURN OFF PXCT "SPEC/PXCT OFF"
; 1988 LOAD PAGE TABLE "SPEC/LDPAGE"
; 1989 LOAD AC BLOCKS "SPEC/LDACBLK"
; 1990 SWEEP "SPEC/SWEEP,PHYSICAL/1"
; 1991 CLRCSH "SPEC/CLRCSH,PHYSICAL/1"
; 1992 LOAD PI "SPEC/LDPI"
; 1993 SET HALT "SPEC/##/74"
; 1994 CLEAR CONTINUE "SPEC/##/40"
; 1995 CLEAR EXECUTE "SPEC/##/20"
; 1996 CLEAR RUN "SPEC/##/10"
; 1997 UNHALT "SPEC/##/62"
; 1998 SET APR ENABLES "SPEC/APR EN"
; 1999 ABORT MEM CYCLE "DBUS/DBM,RAMADR/VMA,DBM/MEM,AD/ZERO,SPEC/MEMCLR,LOAD VMA"
; 2000 CLR IO BUSY "SPEC/CLR IO BUSY"
; 2001 CLR IO LATCH "SPEC/CLR IO LATCH"
; 2002
```


; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 2003 .TOC "MACROS -- PC FLAGS"
; 2004
; 2005 CHANGE FLAGS "SPEC/FLAGS"
; 2006
; 2007 SET AROV "CHANGE FLAGS, HOLD USER/1, SETOV/1, TRAP1/1"
; 2008 SET FOV "CHANGE FLAGS, HOLD USER/1, SETFOV/1, TRAP1/1"
; 2009 SET NO DIVIDE "CHANGE FLAGS, HOLD USER/1, SETOV/1, SETNDV/1, TRAP1/1"
; 2010 SET FL NO DIVIDE "SET NO DIVIDE, SETFOV/1"
; 2011
; 2012 ASH AROV "SPEC/ASHOV"
; 2013 SET FPD "CHANGE FLAGS, HOLD USER/1, SETFPD/1"
; 2014 CLR FPD "CHANGE FLAGS, HOLD USER/1, CLRFPD/1"
; 2015
; 2016 SET PDL OV "CHANGE FLAGS, HOLD USER/1, TRAP2/1"
; 2017 SET TRAP1 "CHANGE FLAGS, HOLD USER/1, TRAP1/1"
; 2018
; 2019 LOAD PCU "CHANGE FLAGS, LD PCU/1"
; 2020 UPDATE USER "CHANGE FLAGS, HOLD USER/1"
; 2021 LEAVE USER "CHANGE FLAGS, HOLD USER/O"
; 2022
; 2023 JFCL FLAGS "CHANGE FLAGS, HOLD USER/1, JFCLFLG/1"
; 2024
; 2025 LOAD FLAGS "CHANGE FLAGS, LD FLAGS/1"
; 2026 EXP TEST "SPEC/EXPTST"
; 2027 AD FLAGS "CHANGE FLAGS, ADFLGS/1, HOLD USER/1"
; 2028
; 2029 NO DIVIDE "SET NO DIVIDE, J/NIDISP"
; 2030 FL NO DIVIDE "SET FL NO DIVIDE, J/NIDISP"
; 2031
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

; 2032 .TOC "MACROS -- PAGE FAIL FLAGS"
; 2033
; 2034 STATE_[] "[FLG]_#,STATE/@1,HOLD LEFT"
; 2035 END STATE "[FLG]_O, HOLD LEFT"
; 2036
; 2037 END BLT "END STATE"
; 2038 END MAP "END STATE"
; 2039

; T10KL.MCR[10,1141]
; KS10.MIC[10,1141]

15:34 27-JULY-1984
07:10 27-JULY-1984

MICRO 31(254)

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

```
; 2040 .TOC      "MACROS -- SINGLE SKIPS"
; 2041
; 2042 SKIP IF ACO      "SKIP/ACO"      ;SKIPS IF:
; 2043 SKIP DPO        "SKIP/DPO"      ;THE AC NUMBER IS ZERO
; 2044 SKIP DP18       "SKIP/DP18"     ;DP BIT 0=1
; 2045 SKIP AD.EQ.O    "SKIP/ADEQO,DT/3T" ;DP BIT 18=1
; 2046 SKIP AD.LE.O    "SKIP/LE,DT/3T" ;ADDER OUTPUT IS ZERO
; 2047                ;ADDER OUTPUT IS LESS THAN OR EQUAL
; 2048 SKIP ADL.LE.O   "SKIP/LLE,DT/3T" ; TO ZERO.
; 2049                ;ADDER LEFT IS LESS THAN OR EQUAL
; 2050 SKIP FPD        "SKIP/FPD"      ; TO ZERO.
; 2051 SKIP KERNEL    "SKIP/KERNEL"   ;FIRST-PART-DONE PC FLAG IS SET
; 2052 SKIP IO LEGAL  "SKIP/IOLGL"    ;USER=0
; 2053 SKIP CRYO      "SKIP/CRYO"     ;USER=0 OR USER IOT=1
; 2054 SKIP CRY1      "SKIP/CRY1"     ;ADDER BIT CRYO=1 (NOT PC FLAG BIT)
; 2055 SKIP CRY2      "SKIP/CRY2,DT/3T" ;ADDER BIT CRY1=1 (NOT PC FLAG BIT)
; 2056 SKIP JFCL      "SKIP/JFCL"     ;ADDER BIT CRY2=1
; 2057 SKIP ADL.EQ.O  "SKIP/ADLEQO"   ; IF JFCL SHOULD JUMP
; 2058 SKIP ADR.EQ.O  "SKIP/ADREQO"   ;ALU BITS -2 TO 17 = 0
; 2059 SKIP IRPT     "SKIP/INT"       ;ALU BITS 18-35 = 0
; 2060 SKIP -1MS     "SKIP/-1 MS"     ;INTERRUPT IS PENDING
; 2061 SKIP AC REF   "SKIP/ACREF"     ;DON'T SKIP IF 1MS TIMER HAS EXPIRED.
; 2062 SKIP EXECUTE "SKIP/EXECUTE"   ;VMA IS 0-17
; 2063 TXXX TEST    "SKIP/TXXX"      ;CONSOLE EXECUTE
; 2064                ;TEST INSTRUCTION SHOULD SKIP
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 59
; KS10.MIC[10,1141] 07:10 27-JULY-1984 MACROS -- SPECIAL DISPATCH MACROS

; 2065 .TOC "MACROS -- SPECIAL DISPATCH MACROS"
; 2066
; 2067 NEXT INST "DISP/NICOND,SPEC/NICOND,J/NICOND"
; 2068 NEXT INST FETCH "DISP/NICOND,SPEC/NICOND,J/NICOND-FETCH"
; 2069 EA MODE DISP "DISP/EAMODE,RAMADR/XR#"
; 2070 AREAD "DISP/AREAD,WAIT/1,AREAD/1,MEM/1,J/O"
; 2071 B DISP "DISP/BDISP"
; 2072 BWRITE DISP "B DISP,MEM/1,BWRITE/1,WRITE CYCLE/1,J/BWRITE"
; 2073 INST DISP "DISP/DROM,J/O"
; 2074 EXIT "BWRITE DISP,SPEC/O, WRITE TEST/1"
; 2075 AD FLAGS EXIT "BWRITE DISP, WRITE TEST/O, AD FLAGS"
; 2076 FL-EXIT "WRITE CYCLE/1,WRITE TEST/1,MEM/1,BWRITE/1,B DISP,J/FL-BWRITE"
; 2077 TEST DISP "B DISP,J/TEST-TABLE"
; 2078 SKIP-COMP DISP "B DISP,J/SKIP-COMP-TABLE"
; 2079 JUMP DISP "B DISP,J/JUMP-TABLE"
; 2080 DONE "VMA [PC],LOAD VMA, FETCH, NEXT INST FETCH"
; 2081 JUMPA "[PC]_[AR],HOLD LEFT,LOAD VMA,FETCH,NEXT INST FETCH"
; 2082 UUD "[HR]_[HR].AND.#,#/777740,HOLD RIGHT,J/UUOGO"
; 2083 LUUD "[AR]_O XWD [40], J/LUUD01"
; 2084 PAGE FAIL TRAP "TL [FLG], FLG.PI/1, J/PFT"
; 2085 TAKE INTERRUPT "[FLG]_[FLG].OR.#,FLG.PI/1,HOLD RIGHT,J/PI"
; 2086 INTERRUPT TRAP "WORK[SV.AR]_[AR], J/ITRAP"
; 2087 MUL DISP "DISP/MUL"
; 2088 DIV DISP "DISP/DIV"
; 2089 BYTE DISP "DISP/BYTE, DT/3T"
; 2090 SCAD DISP "DISP/SCADO" ;SKIP (2'S WEIGHT) IS SCAD IS MINUS
; 2091 RETURN [] "DISP/RETURN,J/@1"
; 2092 PI DISP "DISP/PI"
; 2093 NORM DISP "DISP/NORM,DT/3T"
; 2094 DISMISS "TR [PI], #/077400, CALL [JEN1],DT/3T"
; 2095 CALL LOAD PI "[TO]_[PI] SWAP, CALL [LDPI2]"
; 2096 HALT [] "AD/47,DEST/AD,B/T1,DBM/#,DBUS/DBM,HALT/@1,RSRC/DA,A/MASK, J/HALTED"
; 2097 CLEANUP DISP "READ [FLG], DBUS/DP, DISP/DP, 3T, J/CLEANUP"
; 2098

```
; 2099 .TOC "DISPATCH ROM MACROS"
; 2100 .DCODE
; 2101
; 2102 ;"A FIELD" MACROS SAY HOW TO FETCH ARGUMENTS
; 2103
; 2104 I "I/1"
; 2105 I-PF "I/1,VMA/O,READ/1"
; 2106 R "A/READ,READ/1"
; 2107 R-PF "A/RD-PF,READ/1"
; 2108 W "A/WRITE,TEST/1"
; 2109 RW "A/READ,TEST/1,READ/1"
; 2110 IW "I/1,TEST/1" ;IMMED WHICH STORE IN E. (SETZM, ETC.)
; 2111 IR "I/1,READ/1" ;START READ A GO TO EXECUTE CODE
; 2112 DBL R "A/DREAD,READ/1" ;AR!ARX _ E!E+1
; 2113 DBL AC "A/DBLAC"
; 2114 SH "A/SHIFT,VMA/O,READ/1"
; 2115 SHC "A/DSHIFT,VMA/O,READ/1"
; 2116 FL-R "A/FP,READ/1" ;FLOATING POINT READ
; 2117 FL-RW "A/FP,READ/1,TEST/1"
; 2118 FL-I "A/FPI,READ/O" ;FLOATING POINT IMMEDIATE
; 2119 DBL FL-R "A/DFP,READ/1"
; 2120 IOT "A/IOT" ;CHECK FOR IO LEGAL
; 2121
; 2122 ;"B FIELD" MACROS SAY HOW TO STORE RESULTS
; 2123
; 2124 AC "B/AC"
; 2125 M "B/MEM,TEST/1,COND FUNC/1"
; 2126 B "B/BOTH,TEST/1,COND FUNC/1"
; 2127 S "B/SELF,TEST/1,COND FUNC/1"
; 2128 DAC "B/DBLAC"
; 2129 DBL B "B/DBLB,TEST/1,COND FUNC/1"
; 2130 FL-AC "FL-B/AC" ;FLOATING POINT
; 2131 FL-MEM "FL-B/MEM,TEST/1,COND FUNC/1" ;FLOATING POINT TO MEMORY
; 2132 FL-BOTH "FL-B/BOTH,TEST/1,COND FUNC/1" ;FLOATING POINT TO BOTH
; 2133 ROUND "ROUND/1" ;FLOATING POINT ROUNDED
; 2134
; 2135
; 2136 ;CONTROL BITS
; 2137 W TEST "TEST/1"
; 2138 AC DISP "ACDISP/1"
; 2139 .UCODE
; 2140
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
; 2141
; 2142 .TOC "POWER UP SEQUENCE"
; 2143
; 2144 .UCODE
; 2145
; 2146 ;HERE IS WHERE WE FIRE THE MACHINE UP DURING POWER ON
; 2147
; 2148
; 2149
; 2150 O: [MASK]_#, #/377777 ;BUILD A MASK WITH
; [MASK]_[MASK]*2 ; A ONE IN 36-BITS AND O
; 2151 [MASK]_[MASK].OR.#,#/1 ; IN BITS -2,-1,36,37
; 2152 [MAG]_[MASK]*.5 ;MAKE CONSTANT
; 2153 [XWD1]_#, #/1 ;CONSTANT WITH 1 IN EACH
; 2154 ; HALF WORD
; 2155 [ONE]_O XWD [1], ;THE CONSTANT 1
; 2156 CALL/1 ;RESET STACK (CAN NEVER RETURN
; 2157 ; TO WHERE MR LEFT US)
; 2158 3: [AR]_O XWD [376000] ;ADDRESS OF HALT STATUS
; 2159 ; BLOCK
; 2160 WORK[HSBADR]_[AR] ;SAVE FOR HALT LOOP
; 2161 [UBR]_O, ABORT MEM CYCLE ;CLEAR THE UBR AND RESET
; 2162 ; MEMORY CONTROL LOGIC
; 2163 [EBR]_O, LOAD AC BLOCKS ;CLEAR THE EBR AND FORCE
; 2164 ; PREVIOUS AND CURRENT AC
; 2165 ; BLOCKS TO ZERO
; 2166 [FLG]_O, SET APR ENABLES ;CLEAR THE STATUS FLAGS AND
; 2167 ; DISABLE ALL APR CONDITIONS
; 2168 WORK[APR]_[FLG] ;ZERO REMEMBERED ENABLES
; 2169
; 2170 WORK[TIME0]_[FLG] ;CLEAR TIME BASE
; 2171 WORK[TIME1]_[FLG] ; ..
; 2172 .IF/FULL
; 2173 AC[BINO]_O ;COMPUTE A TABLE OF POWERS OF
; 2174 AC[BIN1]_1 ; TEN
; 2175 [AR]_O, SC 19. ;WE WANT TO GET 22 NUMBERS
; 2176 WORK[DECLO]_1 ;STARTING WITH 1
; 2177 WORK[DECHI]_O ; ..
; 2178 [HR]_#, WORK/DECLO ;ADDRESS OF LOW WORD
; 2179 [BRX]_#, WORK/DECHI ;ADDRESS OF HIGH WORD
; 2180 TENLP: [BRX]_[BRX]+1, LOAD VMA ;ADDRESS THE HIGH WORD
; 2181 =O* [ARX]_AC[BIN1], ;LOW WORD TO ARX
; 2182 CALL [DBSLOW] ;MULTIPLY BY TEN
; 2183 RAM_[BR] ;SAVE HIGH WORD
; 2184 [HR]_[HR]+1, LOAD VMA ;WHERE TO STORE LOW WORD
; 2185 RAM_[ARX], STEP SC ;STORE LOW WORD AND SEE IF
; 2186 ; WE ARE DONE
; 2187 =O J/TENLP ;NOT YET--KEEP GOING
; 2188 [BR].XOR.#, 3T, SKIP ADL.EQ.O, #/330656
; 2189 ;DID WE GET THE RIGHT ANSWER
; 2190 ; IN THE TOP 18 BITS?
; 2191 =O**O HALT [MULERR] ;NO--CPU IS BROKEN
; 2192 .ENDIF/FULL
; 2193
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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, 3622,4221,0014,4174,4007,0700,0010,0000,0000 ; 2194 =0**1 [PI]_O, CALL [LOADPI] ;CLEAR PI STATE
; 2195 =1**1 ;CLEAR REGISTERS SO NO
; 2196 ;PARITY ERROR HAPPEN
;;2197 .IFNOT/FULL
; 2198 [ARX]_O ;WRITTEN WHILE COMPUTING POWERS
; 2199 [BR]_O ;OF 10
; 2200 [BRX]_O
; 2201 .ENDIF/FULL
U 0151, 0324,4751,1217,4374,4007,0700,0000,0000,0120 ; 2202 [T1]_O XWD [120] ;RH OF 120 CONTAINS START ADDRESS
; 2203 ; FOR SIMULATOR. FOR THE REAL
; 2204 ; MACHINE IT IS JUST DATA WITH
; 2205 ; GOOD PARITY.
; 2206 =
; 2207 ;THE CODE UNDER .IF/SIM MUST USE THE SAME ADDRESS AS THE CODE
; 2208 ; UNDER .IFNOT/SIM SO THAT MICROCODE ADDRESSES DO NOT CHANGE BETWEEN
; 2209 ; VERSIONS
; 2210 .IF/SIM
; 2211 VMA_[T1], START READ ;READ THE WORD
; 2212 MEM READ, [PC]_MEM, HOLD LEFT, J/START
; 2213 ;GO FIRE UP SIMULATOR AT THE
; 2214 ; PROGRAMS STARTING ADDRESS
; 2215 .IFNOT/SIM
; 2216 [PC]_O, ;CLEAR LH OF PC
; 2217 LEAVE USER, ;ENTER EXEC MODE
; 2218 LOAD FLAGS ;CLEAR TRAP FLAGS
; 2219 [T1]_#, HALT/POWER, ;LOAD T1 WITH POWER UP CODE
; 2220 J/PWRON ;ENTER HALT LOOP. DO NOT STORE
; 2221 ; HALT STATUS BLOCK
; 2222 .ENDIF/SIM
; 2223
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 2224 .TOC "THE INSTRUCTION LOOP -- START NEXT INSTRUCTION"
; 2225
; 2226 ;ALL INSTRUCTIONS EXCEPT JUMP'S AND UOO'S END UP HERE
; 2227 1400:
; 2228 DONE: DONE
; 2229 1401: VMA_[PC]+1, NEXT INST FETCH, FETCH
; 2230 =0
; 2231 SKIP: VMA_[PC]+1, NEXT INST FETCH, FETCH
; 2232 DONE
; 2233
; 2234
; 2235 ;16-WAY DISPATCH BASED ON NEXT INSTRUCTION
; 2236 =0000
; 2237 NICOND:
; 2238 =0001 [AR]_O XWD [423], ;TRAP TYPE 3
; 2239 ; GET ADDRESS OF TRAP INST
; 2240 TURN OFF PXCT, ;CLEAR PXCT
; 2241 J/TRAP ;PROCESS TRAP (INOUT.MIC)
; 2242 =0010 [AR]_O XWD [422], ;TRAP TYPE 2
; 2243 TURN OFF PXCT, ;CLEAR PXCT
; 2244 J/TRAP ;GO TRAP
; 2245 =0011 [AR]_O XWD [421], ;TRAP TYPE 1
; 2246 TURN OFF PXCT, ;TURN OF PXCT
; 2247 J/TRAP ;GO TRAP
; 2248 =0101 HALT [CSL] ;"HA" COMMAND TO 8080
; 2249 =0111
; 2250 START: VMA_[PC], ;LOAD VMA
; 2251 FETCH, ;INDICATE INSTRUCTION FETCH
; 2252 J/XCTGO ;GO GET INSTRUCTION
; 2253 ;THE NEXT SET OF CASES ARE USED WHEN THERE IS A FETCH
; 2254 ; IN PROGRESS
; 2255 =1000
; 2256 NICOND-FETCH:
; 2257 =1001 [AR]_O XWD [423], ;TRAP TYPE 3
; 2258 TURN OFF PXCT,
; 2259 J/TRAP
; 2260 =1010 [AR]_O XWD [422], ;TRAP TYPE 2
; 2261 TURN OFF PXCT,
; 2262 J/TRAP
; 2263 =1011 [AR]_O XWD [421], ;TRAP TYPE 1
; 2264 TURN OFF PXCT,
; 2265 J/TRAP
; 2266 =1101 HALT [CSL] ;"HA" COMMAND TO 8080
; 2267 =1111
; 2268 XCTGO: MEM READ, ;WAIT FOR MEMORY
; 2269 [HR]_MEM, ;PUT DATA IN HR
; 2270 LOAD INST, ;LOAD IR & AC #
; 2271 J/INCP C ;GO BUMP PC
; 2272 =
; 2273
```

U 1400, 0110,3443,0100,4174,4156,4700,0200,0014,0012
U 1401, 0110,0111,0701,4170,4156,4700,0200,0014,0012
U 0260, 0110,0111,0701,4170,4156,4700,0200,0014,0012
U 0261, 0110,3443,0100,4174,4156,4700,0200,0014,0012
U 0101, 3510,4751,1203,4374,4367,0700,0000,0000,0423
U 0102, 3510,4751,1203,4374,4367,0700,0000,0000,0422
U 0103, 3510,4751,1203,4374,4367,0700,0000,0000,0421
U 0105, 0104,4751,1217,4374,4007,0700,0000,0000,0002
U 0107, 0117,3443,0100,4174,4007,0700,0200,0014,0012
U 0111, 3510,4751,1203,4374,4367,0700,0000,0000,0423
U 0112, 3510,4751,1203,4374,4367,0700,0000,0000,0422
U 0113, 3510,4751,1203,4374,4367,0700,0000,0000,0421
U 0115, 0104,4751,1217,4374,4007,0700,0000,0000,0002
U 0117, 0343,3771,0002,4365,5617,0700,0200,0000,0002

; T10KL.MCR[10,1141] 15:34 27-JULY-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 0343, 0201,0111,0701,2170,4366,6700,0200,0010,0010

```
; 2274 ;HERE WE POINT PC TO NEXT INSTRUCTION WHILE WE WAIT FOR
; 2275 ; EFFECTIVE ADDRESS LOGIC TO SETTLE
; 2276 INCPC: VMA_[PC]+1, ;ADDRESS OF NEXT INSTRUCTION
; 2277 FETCH/1, ;INSTRUCTION FETCH
; 2278 TURN OFF PXCT, ;CLEAR EFFECT OF PXCT
; 2279 EA MODE DISP ;DISPACTH OF INDEXING AND @
; 2280
; 2281 ;MAIN EFFECTIVE ADDRESS CALCULATION
; 2282 =0001
; 2283 EACALC:
; 2284 ;
; 2285 ; THE FIRST 4 CASES ARE USED ONLY FOR JRST
; 2286 ;
; 2287
```

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

```
; 2288 ;CASE 0 -- JRST 0,FOO(XR)
; 2289 [PC]_[HR]+XR, ;UPDATE PC
; 2290 HOLD LEFT, ;ONLY RH
; 2291 LOAD VMA, FETCH, ;START GETTING IT
; 2292 NEXT INST FETCH ;START NEXT INST
; 2293
```

U 0203, 0110.3441,0201,4170,4156,4700,0200,0014,0012

```
; 2294 ;CASE 2 -- JRST 0,FOO
; 2295 [PC]_[HR], ;NEW PC
; 2296 HOLD LEFT, ;ONLY RH
; 2297 LOAD VMA, FETCH, ;START GETTING IT
; 2298 NEXT INST FETCH ;START NEXT INST
; 2299
```

U 0205, 0346,0551,0202,2270,4007,0700,0200,0004,0012

```
; 2300 ;CASE 4 -- JRST 0,@FOO(XR)
; 2301 [HR]_[HR]+XR, ;ADD IN INDEX
; 2302 START READ, ;START TO FETCH @ WORD
; 2303 LOAD VMA, ;PUT ADDRESS IN VMA
; 2304 J/FETIND ;GO DO MEM WAIT (FORGET ABOUT JRST)
; 2305
```

U 0207, 0346,3443,0200,4174,4007,0700,0200,0004,0012

```
; 2306 ;CASE 6 -- JRST 0,@FOO
; 2307 VMA_[HR], ;LOAD UP ADDRESS
; 2308 START READ, ;START TO FETCH @ WORD
; 2309 J/FETIND ;GO DO MEM WAIT (FORGET ABOUT JRST)
; 2310
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 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

U 0213, 0000,5741,0203,4174,4001,3700,0200,0000,0342

U 0215, 0346,0551,0202,2270,4007,0700,0200,0004,0112

U 0217, 0346,3443,0200,4174,4007,0700,0200,0004,0112

U 0346, 0363,3771,0002,4361,5217,0700,0200,0000,0102

U 0363, 0201,3443,0100,2174,4006,6700,0200,0010,0010

```
; 2311 ;  
; 2312 ;THESE 4 ARE FOR THE NON-JRST CASE  
; 2313 ;  
; 2314 ;  
; 2315 ;CASE 10 -- JUST INDEXING  
; 2316 INDEX: [HR]_[HR]+XR, ;ADD IN INDEX REGISTER  
; 2317 HOLD LEFT ;JUST DO RIGHT HALF  
; 2318 ;  
; 2319 ;CASE 12 -- NO INDEXING OR INDIRECT  
; 2320 NOMOD: [AR]_EA, ;PUT O,,E IN AR  
; 2321 PXCT DATA, AREAD ;DO ONE OR MORE OF THE FOLLOWING  
; 2322 ; ACCORDING TO THE DROM:  
; 2323 ;1. LOAD VMA  
; 2324 ;2. START READ OR WRITE  
; 2325 ;3. DISPATCH TO 40-57  
; 2326 ; OR DIRECTLY TO EXECUTE CODE  
; 2327 ;  
; 2328 ;CASE 14 -- BOTH INDEXING AND INDIRECT  
; 2329 BOTH: [HR]_[HR]+XR, ;ADD IN INDEX REGISTER  
; 2330 LOAD VMA, PXCT EA, ;PUT ADDRESS IN VMA  
; 2331 START READ, J/FETIND ;START CYCLE AND GO WAIT FOR DATA  
; 2332 ;  
; 2333 ;CASE 16 -- JUST INDIRECT  
; 2334 INDRCT: VMA_[HR], ;LOAD ADDRESS OF @ WORD  
; 2335 START READ, PXCT EA ;START CYCLE  
; 2336 ;  
; 2337 ;  
; 2338 ;HERE TO FETCH INDIRECT WORD  
; 2339 FETIND: MEM READ, [HR]_MEM, ;GET DATA WORD  
; 2340 HOLD LEFT, ;JUST RIGHT HALF  
; 2341 LOAD IND EA ;RELOAD @ AND INDEX FLOPS  
; 2342 ;  
; 2343 XCT2: VMA_[PC], ;PUT PC BACK IN VMA  
; 2344 FETCH/1, ;TURN ON FETCH FLAG  
; 2345 EA MODE DISP, ;REDO CALCULATION FOR  
; 2346 J/EACALC ; NEW WORD  
; 2347 ;
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
; 2348 .TOC "THE INSTRUCTION LOOP -- FETCH ARGUMENTS"
; 2349 ;HERE ON AREAD DISP TO HANDLE VARIOUS CASES OF ARGUMENT FETCH
; 2350
; 2351 ;CASE 0 -- READ (E)
; 2352 40: MEM READ, ;WAIT FOR DATA
; 2353 [AR] MEM, ;PUT WORD IN AR
; 2354 INST DISP ;GO TO EXECUTE CODE
; 2355
; 2356 ;CASE 1 -- WRITE (E)
; 2357 41: [AR]_AC, ;PUT AC IN AR
; 2358 INST DISP ;GO TO EXECUTE CODE
; 2359
; 2360 ;CASE 2 -- DOUBLE READ
; 2361 42: MEM READ, ;WAIT FOR DATA
; 2362 [AR] MEM ;PUT HI WORD IN AR
; 2363 VMA [HR]+1, PXCT DATA, ;POINT TO E+1
; 2364 START READ ;START MEMORY CYCLE
; 2365 MEM READ, ;WAIT FOR DATA
; 2366 [ARX] MEM, ;LOW WORD IN ARX
; 2367 INST DISP ;GO TO EXECUTE CODE
; 2368
; 2369 ;CASE 3 -- DOUBLE AC
; 2370 43: [AR]_AC ;GET HIGH AC
; 2371 [ARX]_AC[1], ;PUT C(AC+1) IN ARX
; 2372 INST DISP ;GO TO EXECUTE CODE
; 2373
; 2374 ;CASE 4 -- SHIFT
; 2375 44:
; 2376 SHIFT: READ [AR], ;LOOK AT EFFECTIVE ADDRESS
; 2377 SKIP DP18, ;SEE IF LEFT OR RIGHT
; 2378 SC_SHIFT-1, ;PUT NUMBER OF PLACES TO SHIFT IN
; 2379 LOAD FE, ; SC AND FE
; 2380 INST DISP ;GO DO THE SHIFT
; 2381
; 2382 ;CASE 5 -- SHIFT COMBINED
; 2383 45: Q_AC[1] ;PUT LOW WORD IN Q
; 2384 [BR]_AC*.5 LONG ;PUT AC IN BR & SHIFT BR!Q RIGHT
; 2385 [BR]_[BR]*.5 LONG, ;SHIFT BR!Q 1 MORE PLACE RIGHT
; 2386 J/SHIFT ;GO DO SHIFT SETUP
; 2387
```

U 0040, 0000,3771,0003,4365,5001,2700,0200,0000,0002

U 0041, 0000,3771,0003,0276,6001,2700,0000,0000,0000

U 0042, 0366,3771,0003,4365,5007,0700,0200,0000,0002

U 0366, 0371,0111,0702,4170,4007,0700,0200,0004,0312

U 0371, 0000,3771,0004,4365,5001,2700,0200,0000,0002

U 0043, 0401,3771,0003,0276,6007,0700,0000,0000,0000

U 0401, 0000,3771,0004,1276,6001,2701,0000,0000,1441

U 0044, 0000,3333,0003,4174,4001,2530,3000,0041,5777

U 0045, 0406,3772,0000,1275,5007,0701,0000,0000,1441

U 0406, 0431,3776,0005,0274,4007,0701,0000,0000,0000

U 0431, 0044,3446,0505,4174,4007,0700,0000,0000,0000

; T10KL.MCR[10,1141] 15:34 27-JULY-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, 0304,3770,0303,4344,4007,0700,0000,0000,0000 ; 2388 ;CASE 6 -- FLOATING POINT IMMEDIATE
; 2389 46: [AR]_[AR] SWAP, ;FLIP BITS TO LEFT HALF
; 2390 J/FPRO ;JOIN COMMON F.P. CODE
; 2391
; 2392 ;CASE 7 -- FLOATING POINT
U 0047, 0304,3771,0003,4365,5007,0700,0200,0000,0002 ; 2393 47: MEM READ, ;WAIT FOR MEMORY (SPEC/MEM WAIT)
; 2394 [AR]_MEM ;DATA INTO AR
; 2395 =0
; 2396 FPRO: READ [AR], ;LOOK AT NUMBER
; 2397 SC_EXP, FE_EXP, ;PUT EXPONENT IN SC & FE
; 2398 SKIP DPO, ;SEE IF NEGATIVE
U 0304, 0434,3333,0003,4174,4007,0520,3010,0041,2000 ; 2399 CALL [ARSIGN] ;EXTEND AR SIGN
; 2400 FPR1: [ARX]_O, ;ZERO ARX
U 0305, 0000,4221,0004,4174,4001,2700,0000,0000,0000 ; 2401 INST DISP ;GO TO EXECUTE CODE
; 2402
; 2403 ;CASE 10 -- READ THEN PREFETCH
U 0050, 0000,3770,0103,4365,5001,2700,0200,0014,0012 ; 2404 50: MEM READ, ;WAIT FOR DATA
; 2405 [AR]_MEM THEN FETCH, ;PUT DATA IN AR AND START A READ
; 2406 ; VMA HAS PC+1.
; 2407 INST DISP ;GO DO IT
; 2408
; 2409 ;CASE 11 -- DOUBLE FLOATING READ
U 0051, 0372,3771,0005,4365,5177,0521,3000,0041,2000 ; 2410 51: SPEC MEM READ, ;WAIT FOR DATA
; 2411 [BR]_MEM, ;HOLD IN BR
; 2412 SC_EXP, FE_EXP, ;SAVE EXPONENT
; 2413 SKIP DPO, 3T ;SEE IF MINUS
; 2414 =0 [AR]_[AR]+1, ;POINT TO E+1
; 2415 LOAD VMA, PXCT DATA, ;PUT IN VMA
U 0372, 0445,0111,0703,4174,4007,0700,0200,0004,0312 ; 2416 START READ, J/DFPR1 ;GO GET POSITIVE DATA
; 2417 [AR]_[AR]+1, ;POINT TO E+1
; 2418 LOAD VMA, PXCT DATA, ;PUT IN VMA
U 0373, 0432,0111,0703,4174,4007,0700,0200,0004,0312 ; 2419 START READ ;GO GET NEGATIVE DATA
; 2420 [BR]_SIGN, ;SMEAR MINUS SIGN
U 0432, 0451,3551,0505,4374,0007,0700,0000,0077,7000 ; 2421 J/DFPR2 ;CONTINUE BELOW
U 0445, 0451,4551,0505,4374,0007,0700,0000,0000,0777 ; 2422 DFPR1: [BR]_+SIGN ;SMEAR PLUS SIGN
; 2423 DFPR2: MEM READ, 3T, ;WAIT FOR MEMORY
; 2424 [ARX]_(MEM.AND.[MAG])* .5,
; 2425 ASH ;SET SHIFT PATHS
; 2426 [AR]_[BR]*.5 ;SHIFT AR
; 2427 [AR]_[AR]*.5, ;COMPLETE SHIFTING
U 0467, 0471,3447,0303,4174,4007,0700,2000,0011,0000 ; 2428 SC_FE ;PAGE FAIL MAY HAVE ZAPPED
; 2429 ; THE SC.
; 2430 VMA_[PC], FETCH, ;GET NEXT INST
U 0471, 0000,3443,0100,4174,4001,2700,0200,0014,0012 ; 2431 INST DISP ;DO THIS ONE
; 2432
```

; T10KL.MIC[10,1141] 15:34 27-JULY-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, 0402,4443,0000,4174,4007,0040,0000,0000,0000
U 0402, 2732,4551,0202,4374,0007,0700,0000,0077,7740
U 0403, 0000,4443,0000,4174,4001,2700,0000,0000,0000

```
; 2433 ;CASE 12 -- TEST FOR IO LEGAL
; 2434 52: SKIP IO LEGAL ;IS IO LEGAL?
; 2435 =0 UUO ;NO
; 2436 INST DISP ;YES--DO THE INSTRUCTION
; 2437
; 2438
; 2439 ;CASE 13 -- RESERVED
; 2440 ;53:
; 2441
; 2442 ;CASE 14 -- RESERVED
; 2443 ;54:
; 2444
; 2445 ;CASE 15 -- RESERVED
; 2446 ;55:
; 2447
; 2448 ;CASE 16 -- RESERVED
; 2449 ;56:
; 2450
; 2451 ;CASE 17 -- RESERVED
; 2452 ;57:
; 2453
; 2454 ;EXTEND AR SIGN.
; 2455 ;CALL WITH SKIP ON ARO, RETURNS 1 ALWAYS
; 2456 =0
; 2457 ARSIGN: [AR]_+SIGN, RETURN [1] ;EXTEND + SIGN
; 2458 [AR]_-SIGN, RETURN [1] ;EXTEND - SIGN
; 2459
```

U 0434, 0001,4551,0303,4374,0004,1700,0000,0000,0777
U 0435, 0001,3551,0303,4374,0004,1700,0000,0077,7000

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 2460 .TOC "THE INSTRUCTION LOOP -- STORE ANSWERS"  
; 2461  
; 2462 ;NOTE: INSTRUCTIONS WHICH STORE IN BOTH AC AND MEMORY  
; 2463 ; (E.G. ADDB, AOS) MUST STORE IN MEMORY FIRST  
; 2464 ; SO THAT IF A PAGE FAIL HAPPENS THE AC IS  
; 2465 ; STILL INTACT.  
; 2466  
; 2467 1500:  
; 2468 BWRITE: ;BASE ADDRESS OF BWRITE DISPATCH  
; 2469  
; 2470 ;CASE 0 -- RESERVED  
; 2471 ;1500:  
; 2472  
; 2473 ;CASE 1 -- RESERVED  
; 2474 ;1501:  
; 2475  
; 2476 ;CASE 2 -- RESERVED  
; 2477 ;1502:  
; 2478  
; 2479 ;CASE 3 -- RESERVED  
; 2480 ;1503:  
; 2481  
; 2482 ;CASE 4 -- STORE SELF  
; 2483 1504:  
; 2484 STSELF: SKIP IF ACO, ;IS AC # ZERO?  
; 2485 J/STBTH1 ;GO TO STORE BOTH CASE  
; 2486  
; 2487 ;CASE 5 -- STORE DOUBLE AC  
; 2488 1505:  
; 2489 DAC: AC[1][ARX], ;STORE AC 1  
; 2490 J/STAC ;GO STORE AC  
; 2491  
; 2492 ;CASE 6 -- STORE DOUBLE BOTH (KA10 STYLE MEM_AR ONLY)  
; 2493 1506:  
; 2494 STDBTH: MEM WRITE, ;WAIT FOR MEMORY  
; 2495 MEM [AR], ;STORE AR  
; 2496 J/DAC ;NOW STORE AC & AC+1  
; 2497  
; 2498 ;CASE 7 -- RESERVED  
; 2499 ;1507:  
; 2500
```

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

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 2501 ;CASE 10 -- RESERVED
; 2502 ;1510:
; 2503
; 2504 ;CASE 11 -- RESERVED
; 2505 ;1511:
; 2506
; 2507 ;CASE 12 -- RESERVED
; 2508 ;1512:
; 2509
; 2510 ;CASE 13 -- RESERVED
; 2511 ;1513:
; 2512
; 2513 ;CASE 14 -- RESERVED
; 2514 1514:
; 2515 FL-BWRITE: ;THE NEXT 4 CASES ARE ALSO
; 2516 ;USED IN FLOATING POINT
; 2517 HALT [BW14]
; 2518
; 2519 ;CASE 15 -- STORE AC
; 2520 1515:
; 2521 STAC: AC [AR], ;STORE AC
; 2522 NEXT INST ;DO NEXT INSTRUCTION
; 2523
; 2524 ;CASE 16 -- STORE IN MEMORY
; 2525 1516:
; 2526 STMEM: MEM WRITE, ;WAIT FOR MEMORY
; 2527 MEM [AR], ;STORE AR
; 2528 J/DONE ;START FETCH OF NEXT
; 2529
; 2530 ;CASE 17 -- STORE BOTH
; 2531 1517:
; 2532 STBOTH: MEM WRITE, ;WAIT FOR MEMORY
; 2533 MEM [AR], ;STORE AR
; 2534 J/STAC ;NOW STORE AC
; 2535
; 2536 =0
; 2537 STBTH1: MEM WRITE, ;WAIT FOR MEMORY
; 2538 MEM [AR], ;STORE AR
; 2539 J/STAC ;NOW STORE AC
; 2540 STORE: MEM WRITE, ;WAIT FOR MEMORY
; 2541 MEM [AR], ;STORE AC
; 2542 J/DONE ;START NEXT INST
; 2543
```

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

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```

; 2544 .TOC "MOVE GROUP"
; 2545
; 2546 .DCODE
D 0200, 1015, 1515, 1100 ; 2547 200: R-PF, AC, J/STAC ;MOVE
D 0201, 0015, 1515, 3000 ; 2548 I-PF, AC, J/STAC ;MOVE I
D 0202, 0116, 1404, 0700 ; 2549 W, M, J/MOVE ;MOVEM
D 0203, 0004, 1504, 1700 ; 2550 RW, S, J/STSELF ;MOVES
; 2551
D 0204, 1015, 1402, 1100 ; 2552 204: R-PF, AC, J/MOVS ;MOVS
D 0205, 0015, 1402, 3000 ; 2553 I-PF, AC, J/MOVS ;MOVSI
D 0206, 0116, 1402, 0700 ; 2554 W, M, J/MOVS ;MOVSM
D 0207, 0004, 1402, 1700 ; 2555 RW, S, J/MOVS ;MOVSS
; 2556
D 0210, 1015, 1405, 1100 ; 2557 210: R-PF, AC, J/MOVN ;MOVN
D 0211, 0015, 1405, 3000 ; 2558 I-PF, AC, J/MOVN ;MOVNI
D 0212, 0116, 1405, 0700 ; 2559 W, M, J/MOVN ;MOVNM
D 0213, 0004, 1405, 1700 ; 2560 RW, S, J/MOVN ;MOVNS
; 2561
D 0214, 1015, 1403, 1100 ; 2562 214: R-PF, AC, J/MOVM ;MOVM
D 0215, 0015, 1515, 3000 ; 2563 I-PF, AC, J/STAC ;MOVMI
D 0216, 0116, 1403, 0700 ; 2564 W, M, J/MOVM ;MOVMM
D 0217, 0004, 1403, 1700 ; 2565 RW, S, J/MOVM ;MOVMS
; 2566 .UCODE
; 2567
U 1402, 1500, 3770, 0303, 4344, 4003, 7700, 0200, 0003, 0001 ; 2568 1402:
; 2569 MOVSI: [AR]_[AR] SWAP,EXIT
; 2570
; 2571 1403:
U 1403, 1404, 3333, 0003, 4174, 4007, 0520, 0000, 0000, 0000 ; 2572 MOVSI: READ [AR], SKIP DPO, J/MOVE
; 2573
; 2574 1404:
U 1404, 1500, 4443, 0000, 4174, 4003, 7700, 0200, 0003, 0001 ; 2575 MOVSI: EXIT
; 2576 1405:
; 2577 MOVSI: [AR]_[AR], ;NEGATE NUMBER
; 2578 AD FLAGS, 3T, ;UPDATE FLAGS
U 1405, 1404, 2441, 0303, 4174, 4467, 0701, 4000, 0001, 0001 ; 2579 J/MOVE ;STORE ANSWER
; 2580
```


; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

D 0250, 0015, 1406, 1500

U 1406, 0506, 3771, 0005, 0276, 6007, 0700, 0200, 0003, 0002

U 0506, 1515, 3333, 0005, 4175, 5007, 0701, 0200, 0000, 0002

```
; 2581 .TOC "EXCH"
; 2582
; 2583 .DCODE
; 2584 250: R,W TEST, AC, J/EXCH
; 2585 .UCODE
; 2586
; 2587 1406:
; 2588 EXCH: [BR] AC, ;COPY AC TO THE BR
; 2589 START WRITE ;START A WRITE CYCLE
; 2590 MEM WRITE, ;COMPLETE WRITE CYCLE
; 2591 MEM_[BR], ;STORE BR (AC) IN MEMORY
; 2592 J/STAC ;STORE THE AR IN AC. NOTE: AR
; 2593 ; WAS LOADED WITH MEMORY OPERAND
; 2594 ; AS PART OF INSTRUCTION DISPATCH
; 2595
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 73
HALFWORD GROUP

```

; 2596 .TOC "HALFWORD GROUP"
; 2597 ; DESTINATION LEFT HALF
; 2598
; 2599 .DCODE
D 0500, 1015, 1410, 1100 ; 2600 500: R-PF, AC, J/HLL
D 0501, 0015, 1410, 3000 ; 2601 I-PF, AC, J/HLL
D 0502, 0016, 1407, 1700 ; 2602 RW, M, J/HRR ;HLLM = HRR EXCEPT FOR STORE
D 0503, 0004, 1404, 1700 ; 2603 RW, S, J/MOVE ;HLLS = MOVES
; 2604
D 0504, 1015, 1411, 1100 ; 2605 R-PF, AC, J/HRL
D 0505, 0015, 1411, 3000 ; 2606 I-PF, AC, J/HRL
D 0506, 0016, 1413, 1700 ; 2607 RW, M, J/HRLM
D 0507, 0004, 1414, 1700 ; 2608 RW, S, J/HRLS
; 2609
D 0510, 1015, 1432, 1100 ; 2610 510: R-PF, AC, J/HLLZ
D 0511, 0015, 1432, 3000 ; 2611 I-PF, AC, J/HLLZ
D 0512, 0116, 1432, 0700 ; 2612 W, M, J/HLLZ
D 0513, 0004, 1432, 1700 ; 2613 RW, S, J/HLLZ
; 2614
D 0514, 1015, 1424, 1100 ; 2615 R-PF, AC, J/HRLZ
D 0515, 0015, 1424, 3000 ; 2616 I-PF, AC, J/HRLZ
D 0516, 0116, 1424, 0700 ; 2617 W, M, J/HRLZ
D 0517, 0004, 1424, 1700 ; 2618 RW, S, J/HRLZ
; 2619
D 0520, 1015, 1433, 1100 ; 2620 520: R-PF, AC, J/HLLO
D 0521, 0015, 1433, 3000 ; 2621 I-PF, AC, J/HLLO
D 0522, 0116, 1433, 0700 ; 2622 W, M, J/HLLO
D 0523, 0004, 1433, 1700 ; 2623 RW, S, J/HLLO
; 2624
D 0524, 1015, 1425, 1100 ; 2625 R-PF, AC, J/HRLO
D 0525, 0015, 1425, 3000 ; 2626 I-PF, AC, J/HRLO
D 0526, 0116, 1425, 0700 ; 2627 W, M, J/HRLO
D 0527, 0004, 1425, 1700 ; 2628 RW, S, J/HRLO
; 2629
D 0530, 1015, 1430, 1100 ; 2630 530: R-PF, AC, J/HLLE
D 0531, 0015, 1430, 3000 ; 2631 I-PF, AC, J/HLLE
D 0532, 0116, 1430, 0700 ; 2632 W, M, J/HLLE
D 0533, 0004, 1430, 1700 ; 2633 RW, S, J/HLLE
; 2634
D 0534, 1015, 1422, 1100 ; 2635 R-PF, AC, J/HRLE
D 0535, 0015, 1422, 3000 ; 2636 I-PF, AC, J/HRLE
D 0536, 0116, 1422, 0700 ; 2637 W, M, J/HRLE
D 0537, 0004, 1422, 1700 ; 2638 RW, S, J/HRLE
; 2639
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-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

; 2640 ; DESTINATION RIGHT HALF
; 2641
; 2642 540: R-PF, AC, J/HRR
; 2643 I-PF, AC, J/HRR
; 2644 RW, M, J/HLL
; 2645 RW, S, J/MOVE
; 2646
; 2647 R-PF, AC, J/HLR
; 2648 I-PF, AC, J/HLR
; 2649 RW, M, J/HLRM
; 2650 RW, S, J/HLRS
; 2651
; 2652 550: R-PF, AC, J/HRRZ
; 2653 I-PF, AC, J/HRRZ
; 2654 W, M, J/HRRZ
; 2655 RW, S, J/HRRZ
; 2656
; 2657 R-PF, AC, J/HLRZ
; 2658 I-PF, AC, J/HLRZ
; 2659 W, M, J/HLRZ
; 2660 RW, S, J/HLRZ
; 2661
; 2662 560: R-PF, AC, J/HRRO
; 2663 I-PF, AC, J/HRRO
; 2664 W, M, J/HRRO
; 2665 RW, S, J/HRRO
; 2666
; 2667 R-PF, AC, J/HLRO
; 2668 I-PF, AC, J/HLRO
; 2669 W, M, J/HLRO
; 2670 RW, S, J/HLRO
; 2671
; 2672 570: R-PF, AC, J/HRRE
; 2673 I-PF, AC, J/HRRE
; 2674 W, M, J/HRRE
; 2675 RW, S, J/HRRE
; 2676
; 2677 R-PF, AC, J/HLRE
; 2678 I-PF, AC, J/HLRE
; 2679 W, M, J/HLRE
; 2680 RW, S, J/HLRE
; 2681
; 2682 .UCODE
; 2683

;HRRM = HLL EXCEPT FOR STORE
;HRRS = MOVES

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

U 1411, 1410,3770,0303,4344,4007,0700,0000,0000,0000

U 1412, 1407,3770,0303,4344,4007,0700,0000,0000,0000

U 1413, 0511,3770,0303,4344,4007,0700,0000,0000,0000

U 0511, 1402,3771,0003,0270,6007,0700,0000,0000,0000

U 1414, 1500,3770,0303,4344,0003,7700,0200,0003,0001

U 1415, 0512,3770,0303,4344,4007,0700,0000,0000,0000

U 0512, 1402,3771,0003,0276,0007,0700,0000,0000,0000

U 1416, 1500,3770,0303,4340,4003,7700,0200,0003,0001

```
; 2684 ;FIRST THE GUYS THAT LEAVE THE OTHER HALF ALONE
; 2685
; 2686 ;THE AR CONTAINS THE MEMORY OPERAND. SO WE WANT TO PUT THE LH OF
; 2687 ; AC INTO AR TO DO A HRR. OBVIOUS THING FOR HLL.
; 2688 1407:
; 2689 HRR: [AR]_AC,HOLD RIGHT,EXIT
; 2690 1410:
; 2691 HLL: [AR]_AC,HOLD LEFT,EXIT
; 2692
; 2693 ;HRL FLOW:
; 2694 ;AT HRL AR CONTAINS:
; 2695 ;
; 2696 ; !-----!
; 2697 ; ! LH OF (E) ! RH OF (E) !
; 2698 ; !-----!
; 2699 ;
; 2700 ;AR_AR SWAP GIVES:
; 2701 ;
; 2702 ; !-----!
; 2703 ; ! RH OF (E) ! LH OF (E) !
; 2704 ; !-----!
; 2705 ;
; 2706 ;AT HLL, AR_AC,HOLD LEFT GIVES:
; 2707 ;
; 2708 ; !-----!
; 2709 ; ! RH OF (E) ! RH OF AC !
; 2710 ; !-----!
; 2711 ;
; 2712 ;THE EXIT MACRO CAUSES THE AR TO BE STORED IN AC (AT STAC).
; 2713 ; THE REST OF THE HALF WORD IN THIS GROUP ARE VERY SIMILAR.
; 2714
; 2715 1411:
; 2716 HRL: [AR]_[AR] SWAP,J/HLL
; 2717 1412:
; 2718 HLR: [AR]_[AR] SWAP,J/HRR
; 2719
; 2720 1413:
; 2721 HRLM: [AR]_[AR] SWAP
; 2722 [AR]_AC,HOLD LEFT,J/MOVS
; 2723 1414:
; 2724 HRLS: [AR]_[AR] SWAP,HOLD RIGHT,EXIT
; 2725
; 2726 1415:
; 2727 HRLM: [AR]_[AR] SWAP
; 2728 [AR]_AC,HOLD RIGHT,J/MOVS
; 2729 1416:
; 2730 HLRS: [AR]_[AR] SWAP,HOLD LEFT,EXIT
; 2731
```

; T10KL.MIC[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 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

; 2732 ;NOW THE HALFWORD OPS WHICH CONTROL THE "OTHER" HALF.
; 2733 ; ENTER WITH O,,E (E) OR (AC) IN AR
; 2734
; 2735 1417:
; 2736 HRRE: READ [AR],SKIP DP18
; 2737 1420:
; 2738 HRRZ: [AR] LEFT_O, EXIT
; 2739 1421:
; 2740 HRRO: [AR] LEFT_-1, EXIT
; 2741
; 2742 1422:
; 2743 HRLE: READ [AR],SKIP DP18
; 2744 1424:
; 2745 HRLZ: [AR]_#,#/O,HOLD RIGHT,J/MOVS
; 2746 1425:
; 2747 HRLO: [AR]_#,#/777777,HOLD RIGHT,J/MOVS
; 2748
; 2749 1423:
; 2750 HLRE: READ [AR],SKIP DPO
; 2751 1426:
; 2752 HLRZ: [AR]_#,#/O,HOLD LEFT,J/MOVS
; 2753 1427:
; 2754 HLRO: [AR]_#,#/777777,HOLD LEFT,J/MOVS
; 2755
; 2756 1430:
; 2757 HLLR: READ [AR],SKIP DPO
; 2758 1432:
; 2759 HLLZ: [AR] RIGHT_O, EXIT
; 2760 1433:
; 2761 HLLR: [AR] RIGHT_-1, EXIT
; 2762

; T10KL.MCR[10,1141] 15:34 27-JULY-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

D 0120, 0205, 1505, 1100
D 0121, 0215, 1434, 1100

U 1434, 3063, 4551, 0404, 4374, 0007, 0700, 0010, 0037, 7777
U 1436, 1515, 3440, 0404, 1174, 4007, 0700, 0400, 0000, 1441

D 0124, 0300, 1567, 0100
D 0125, 0100, 1565, 0500

U 1565, 3062, 3771, 0004, 1276, 6007, 0701, 0010, 0000, 1441

U 1567, 0531, 0113, 0207, 4174, 4007, 0700, 0200, 0003, 0312
U 0531, 0532, 3333, 0004, 4175, 5007, 0701, 0200, 0000, 0002

U 0532, 0455, 3443, 0200, 4174, 4007, 0700, 0200, 0003, 0312

```
; 2763 .TOC "DMOVE, DMOVN, DMOVEM, DMOVNM"
; 2764
; 2765 .DCODE
; 2766 120: DBL R, DAC, J/DAC
; 2767 DBL R, AC, J/DMOVN
; 2768 .UCODE
; 2769
; 2770 1434:
; 2771 DMOVN: CLEAR ARX0, CALL [DBLNGA]
; 2772 1436: AC[1]_[ARX], J/STAC
; 2773
; 2774 .DCODE
; 2775 124: DBL AC, J/DMOVN1
; 2776 W, J/DMOVNM
; 2777 .UCODE
; 2778
; 2779
; 2780 1565:
; 2781 DMOVNM: [ARX]_AC[1],CALL [DBLNEG]
; 2782 1567:
; 2783 DMOVN1: [HR]+[ONE], ;GET E+1
; 2784 LOAD VMA, ;PUT THAT IN VMA
; 2785 START WRITE, ;STORE IN E+1
; 2786 PXCT DATA ;DATA CYCLE
; 2787 MEM WRITE, MEM_[ARX] ;STORE LOW WORD
; 2788 VMA_[HR], ;GET E
; 2789 LOAD VMA, ;SAVE IN VMA
; 2790 PXCT DATA, ;OPERAND STORE
; 2791 START WRITE, ;START MEM CYCLE
; 2792 J/STORE ;GO STORE AR
; 2793
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```

; 2794 .TOC "BOOLEAN GROUP"
; 2795
; 2796 .DCODE
D 0400, 0015, 1441, 3000 ; 2797 400: I-PF, AC, J/SETZ
D 0401, 0015, 1441, 3000 ; 2798 I-PF, AC, J/SETZ
D 0402, 0016, 1441, 2700 ; 2799 IW, M, J/SETZ
D 0403, 0017, 1441, 2700 ; 2800 IW, B, J/SETZ
; 2801 .UCODE
; 2802
; 2803 1441:
U 1441, 1500, 4221, 0003, 4174, 4003, 7700, 0200, 0003, 0001 ; 2804 SETZ: [AR]_O, EXIT
; 2805
; 2806 .DCODE
D 0404, 1015, 1442, 1100 ; 2807 404: R-PF, AC, J/AND
D 0405, 0015, 1442, 3000 ; 2808 I-PF, AC, J/AND
D 0406, 0016, 1442, 1700 ; 2809 RW, M, J/AND
D 0407, 0017, 1442, 1700 ; 2810 RW, B, J/AND
; 2811 .UCODE
; 2812
; 2813 1442:
U 1442, 1500, 4551, 0303, 0274, 4003, 7700, 0200, 0003, 0001 ; 2814 AND: [AR]_[AR].AND.AC,EXIT
; 2815
; 2816 .DCODE
D 0410, 1015, 1443, 1100 ; 2817 410: R-PF, AC, J/ANDCA
D 0411, 0015, 1443, 3000 ; 2818 I-PF, AC, J/ANDCA
D 0412, 0016, 1443, 1700 ; 2819 RW, M, J/ANDCA
D 0413, 0017, 1443, 1700 ; 2820 RW, B, J/ANDCA
; 2821 .UCODE
; 2822
; 2823 1443:
U 1443, 1500, 5551, 0303, 0274, 4003, 7700, 0200, 0003, 0001 ; 2824 ANDCA: [AR]_[AR].AND.NOT.AC,EXIT
; 2825
; 2826 .DCODE
D 0414, 1015, 1404, 1100 ; 2827 414: R-PF, AC, J/MOVE ;SETM = MOVE
D 0415, 0015, 1404, 3000 ; 2828 I-PF, AC, J/MOVE
D 0416, 0016, 1404, 1700 ; 2829 RW, M, J/MOVE ;SETMM = NOP THAT WRITES MEMORY
D 0417, 0017, 1404, 1700 ; 2830 RW, B, J/MOVE ;SETMB = MOVE THAT WRITES MEMORY
; 2831
; 2832 420:
D 0420, 1015, 1444, 1100 ; 2832 R-PF, AC, J/ANDCM
D 0421, 0015, 1444, 3000 ; 2833 I-PF, AC, J/ANDCM
D 0422, 0016, 1444, 1700 ; 2834 RW, M, J/ANDCM
D 0423, 0017, 1444, 1700 ; 2835 RW, B, J/ANDCM
; 2836 .UCODE
; 2837
; 2838 1444:
U 1444, 1442, 7441, 0303, 4174, 4007, 0700, 0000, 0000, 0000 ; 2839 ANDCM: [AR]_.NOT.[AR],J/AND
; 2840
; 2841 .DCODE
D 0424, 0000, 1400, 1100 ; 2842 424: R, J/DONE
D 0425, 0000, 1400, 2100 ; 2843 I, J/DONE
D 0426, 0116, 1404, 0700 ; 2844 W, M, J/MOVE ;SETAM = MOVEM
D 0427, 0116, 1404, 0700 ; 2845 W, M, J/MOVE ;SETAB, TOO
; 2846 .UCODE
; 2847
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/ES6, DTN: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```

D 0430, 1015, 1445, 1100      ; 2848      .DCODE
D 0431, 0015, 1445, 3000      ; 2849      430:  R-PF,  AC,    J/XOR
D 0432, 0016, 1445, 1700      ; 2850      I-PF,  AC,    J/XOR
D 0433, 0017, 1445, 1700      ; 2851      RW,    M,     J/XOR
                                ; 2852      RW,    B,     J/XOR
                                ; 2853      .UCODE
                                ; 2854
U 1445, 1500, 6551, 0303, 0274, 4003, 7700, 0200, 0003, 0001 ; 2855
                                ; 2856      1445:  XOR:  [AR]_[AR].XOR.AC,EXIT
                                ; 2857
                                ; 2858      .DCODE
D 0434, 1015, 1446, 1100      ; 2859      434:  R-PF,  AC,    J/IO
D 0435, 0015, 1446, 3000      ; 2860      I-PF,  AC,    J/IO
D 0436, 0016, 1446, 1700      ; 2861      RW,    M,     J/IO
D 0437, 0017, 1446, 1700      ; 2862      RW,    B,     J/IO
                                ; 2863      .UCODE
                                ; 2864
U 1446, 1500, 3551, 0303, 0274, 4003, 7700, 0200, 0003, 0001 ; 2865
                                ; 2866      1446:  IOR:  [AR]_[AR].OR.AC,EXIT
                                ; 2867
                                ; 2868      .DCODE
D 0440, 1015, 1447, 1100      ; 2869      440:  R-PF,  AC,    J/AN
D 0441, 0015, 1447, 3000      ; 2870      I-PF,  AC,    J/AN
D 0442, 0016, 1447, 1700      ; 2871      RW,    M,     J/AN
D 0443, 0017, 1447, 1700      ; 2872      RW,    B,     J/AN
                                ; 2873      .UCODE
                                ; 2874
U 1447, 1443, 7441, 0303, 4174, 4007, 0700, 0000, 0000, 0000 ; 2875
                                ; 2876      1447:  ANDCB: [AR]_.NOT.[AR],J/ANDCA
                                ; 2877
                                ; 2878      .DCODE
D 0444, 1015, 1450, 1100      ; 2879      444:  R-PF,  AC,    J/EQ
D 0445, 0015, 1450, 3000      ; 2880      I-PF,  AC,    J/EQ
D 0446, 0016, 1450, 1700      ; 2881      RW,    M,     J/EQ
D 0447, 0017, 1450, 1700      ; 2882      RW,    B,     J/EQ
                                ; 2883      .UCODE
                                ; 2884
U 1450, 1500, 7551, 0303, 0274, 4003, 7700, 0200, 0003, 0001 ; 2885
                                ; 2886      1450:  EQV:  [AR]_[AR].EQV.AC,EXIT
                                ; 2887
                                ; 2888      .DCODE
D 0450, 0015, 1451, 3000      ; 2889      450:  I-PF,  AC,    J/SETCA
D 0451, 0015, 1451, 3000      ; 2890      I-PF,  AC,    J/SETCA
D 0452, 0016, 1451, 2700      ; 2891      IW,    M,     J/SETCA
D 0453, 0017, 1451, 2700      ; 2892      IW,    B,     J/SETCA
                                ; 2893      .UCODE
                                ; 2894
U 1451, 1500, 7771, 0003, 0274, 4003, 7700, 0200, 0003, 0001 ; 2895
                                ; 2896      1451:  SETCA: [AR]_.NOT.AC,EXIT
                                ; 2897
```


; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```

; 2898      .DCODE
D 0454, 1015,1452,1100 ; 2899 454: R-PF, AC, J/ORCA
D 0455, 0015,1452,3000 ; 2900      I-PF, AC, J/ORCA
D 0456, 0016,1452,1700 ; 2901      RW, M, J/ORCA
D 0457, 0017,1452,1700 ; 2902      RW, B, J/ORCA
; 2903      .UCODE
; 2904
; 2905 1452:
U 1452, 0552,7771,0005,0274,4007,0700,0000,0000,0000 ; 2906 ORCA: [BR]_.NOT.AC
U 0552, 1500,3111,0503,4174,4003,7700,0200,0003,0001 ; 2907      [AR]_[AR].OR.[BR],EXIT
; 2908
; 2909      .DCODE
D 0460, 1015,1453,1100 ; 2910 460: R-PF, AC, J/SETCM
D 0461, 0015,1453,3000 ; 2911      I-PF, AC, J/SETCM
D 0462, 0016,1453,1700 ; 2912      RW, M, J/SETCM
D 0463, 0017,1453,1700 ; 2913      RW, B, J/SETCM
; 2914      .UCODE
; 2915
; 2916 1453:
U 1453, 1500,7441,0303,4174,4003,7700,0200,0003,0001 ; 2917 SETCM: [AR]_.NOT.[AR],EXIT
; 2918
; 2919      .DCODE
D 0464, 1015,1454,1100 ; 2920 464: R-PF, AC, J/ORCM
D 0465, 0015,1454,3000 ; 2921      I-PF, AC, J/ORCM
D 0466, 0016,1454,1700 ; 2922      RW, M, J/ORCM
D 0467, 0017,1454,1700 ; 2923      RW, B, J/ORCM
; 2924      .UCODE
; 2925
; 2926 1454:
U 1454, 1446,7441,0303,4174,4007,0700,0000,0000,0000 ; 2927 ORCM: [AR]_.NOT.[AR],J/IOR
; 2928
; 2929      .DCODE
D 0470, 1015,1455,1100 ; 2930 470: R-PF, AC, J/ORCB
D 0471, 0015,1455,3000 ; 2931      I-PF, AC, J/ORCB
D 0472, 0016,1455,1700 ; 2932      RW, M, J/ORCB
D 0473, 0017,1455,1700 ; 2933      RW, B, J/ORCB
; 2934      .UCODE
; 2935
; 2936 1455:
U 1455, 1453,4551,0303,0274,4007,0700,0000,0000,0000 ; 2937 ORCB: [AR]_[AR].AND.AC,J/SETCM
; 2938
; 2939      .DCODE
D 0474, 0015,1456,3000 ; 2940 474: I-PF, AC, J/SETO
D 0475, 0015,1456,3000 ; 2941      I-PF, AC, J/SETO
D 0476, 0016,1456,2700 ; 2942      IW, M, J/SETO
D 0477, 0017,1456,2700 ; 2943      IW, B, J/SETO
; 2944      .UCODE
; 2945
; 2946 1456:
U 1456, 1500,2441,0703,4174,4003,7700,4200,0003,0001 ; 2947 SETO: [AR]_-[ONE], EXIT
; 2948
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 2949 .TOC "ROTATES AND LOGICAL SHIFTS -- ROT, LSH, JFFO"  
; 2950  
; 2951 .DCODE  
; 2952 240: SH, J/ASH  
; 2953 SH, J/ROT  
; 2954 SH, J/LSH  
; 2955 I, J/JFFO  
; 2956 I-PF, J/ASHC  
; 2957 245: SHC, J/ROTC  
; 2958 SHC, J/LSHC  
; 2959 .UCODE  
; 2960  
; 2961
```

U 1612, 0564, 3771, 0003, 0276, 6007, 0700, 1000, 0031, 1777

U 1613, 0572, 4551, 1203, 0276, 6007, 0700, 1000, 0041, 0001

U 0564, 1515, 3445, 0303, 4174, 4007, 0700, 1020, 0041, 0001

```
; 2962 ;HERE IS THE CODE FOR LOGICAL SHIFT. THE EFFECTIVE ADDRESS IS  
; 2963 ; IN AR.  
; 2964 1612:  
; 2965 LSH: [AR] AC, ;PICK UP AC  
; 2966 FE -FE-1, ;NEGATIVE SHIFT  
; 2967 J/LSHL ;SHIFT LEFT  
; 2968 1613: [AR] AC.AND.MASK, ;MAKE IT LOOK POSITIVE  
; 2969 FE_FE+1, ;UNDO -1 AT SHIFT  
; 2970 J/ASHR ;GO SHIFT RIGHT  
; 2971  
; 2972 LSHL: [AR]_[AR]*2, ;SHIFT LEFT  
; 2973 SHIFT, J/STAC ;FAST SHIFT & GO STORE AC  
; 2974
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

U 1622, 0604,4222,0000,4174,4007,0700,0000,0000,0000
U 1623, 0572,3771,0003,0276,6007,0700,1000,0041,0001
U 0572, 1515,3447,0303,4174,4007,0700,1020,0041,0001
U 0604, 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

```
; 2975 ;HERE IS THE CODE FOR ARITHMETIC SHIFT. THE EFFECTIVE ADDRESS IS
; 2976 ; IN AR.
; 2977
; 2978 ASH36 LEFT      "[AR]_[AR]*2 LONG, ASHC, STEP SC, ASH AROV"
; 2979
; 2980 1622:
; 2981 ASH:   Q_0, J/ASHLO      ;HARDWARE ONLY DOES ASHC
; 2982 1623: [AR]_AC,          ;GET THE ARGUMENT
; 2983      FE_FE+1            ;FE HAS NEGATIVE SHIFT COUNT
; 2984 ASHR: [AR]_[AR]*.5,     ;SHIFT RIGHT
; 2985      ASH, SHIFT,        ;FAST SHIFT
; 2986      J/STAC             ;STORE AC WHEN DONE
; 2987
; 2988 ASHLO: [AR]_AC*.5,       ;GET INTO 9 CHIPS
; 2989      STEP SC            ;SEE IF NULL SHIFT
; 2990 =0
; 2991 ASHL:  ASH36 LEFT, J/ASHL ;SHIFT LEFT
; 2992      ;SLOW BECAUSE WE HAVE TO
; 2993      ; TEST FOR OVERFLOW
; 2994
; 2995 ASHX:  [AR]_[AR]*2, J/STAC ;SHIFT BACK INTO 10 CHIPS
; 2996
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

U 1632, 0652,3777,0003,0274,4007,0701,1000,0031,1777
U 1633, 0612,3777,0003,0274,4007,0701,1000,0041,0001
U 0612, 0631,3447,0303,4174,4007,0700,0000,0000,0000
U 0631, 0646,3447,0303,4174,4037,0700,1020,0041,0001
U 0646, 0503,3445,0303,4174,4007,0700,0000,0000,0000
U 0652, 0701,3447,0303,4174,4007,0700,0000,0000,0000
U 0701, 0646,3445,0303,4174,4037,0700,1020,0041,0001

```
; 2997 ;HERE IS THE CODE FOR ROTATE. THE EFFECTIVE ADDRESS IS
; 2998 ; IN AR.
; 2999 1632:
; 3000 ROT: [AR]_AC*.5, ;PICK UP THE AC (& SHIFT)
; 3001 FE_-FE-1, ;NEGATIVE SHIFT COUNT
; 3002 J/ROTL ;ROTATE LEFT
; 3003 1633: [AR]_AC*.5, ;PICK UP THE AC (& SHIFT)
; 3004 FE_FE+1 ;NEGATIVE SHIFT COUNT
; 3005 [AR]_[AR]*.5 ;PUT IN 9 DIPS
; 3006 [AR]_[AR]*.5, ;SHIFT RIGHT
; 3007 ROT, SHIFT ;FAST SHIFT
; 3008 ASHXX: [AR]_[AR]*2,J/ASHX ;SHIFT TO STD PLACE
; 3009
; 3010 ROTL: [AR]_[AR]*.5 ;PUT IN RIGHT 36-BITS
; 3011 [AR]_[AR]*2, ;ROTATE LEFT
; 3012 ROT, SHIFT, ;FAST SHIFT
; 3013 J/ASHXX ;ALL DONE--SHIFT BACK
; 3014
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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
; 3015 1462:
; 3016 JFFO: [BR]_AC.AND.MASK, 4T, ;GET AC WITH NO SIGN
; 3017 SKIP AD.EQ.0 ; EXTENSION. SKIP IF
; 3018 ; ZERO.
; 3019 =0 [PC]_[AR], ;NOT ZERO--JUMP
; 3020 LOAD VMA, FETCH, ;GET NEXT INST
; 3021 J/JFFO1 ;ENTER LOOP
; 3022 AC[1]_O, J/DONE ;ZERO--DONE
; 3023
; 3024 JFFO1: FE_-12. ;WHY -12.? WELL THE
; 3025 ; HARDWARE LOOKS AT
; 3026 ; BIT -2 SO THE FIRST
; 3027 ; 2 STEPS MOVE THE BR
; 3028 ; OVER. WE ALSO LOOK AT
; 3029 ; THE DATA BEFORE THE SHIFT
; 3030 ; SO WE END UP GOING 1 PLACE
; 3031 ; TOO MANY. THAT MEANS THE
; 3032 ; FE SHOULD START AT -3.
; 3033 ; HOWEVER, WE COUNT THE FE BY
; 3034 ; 4 (BECAUSE THE 2 LOW ORDER
; 3035 ; BITS DO NOT COME BACK) SO
; 3036 ; FE_-12.
; 3037 =0
; 3038 JFFOL: [BR]_[BR]*2, ;SHIFT LEFT
; 3039 FE_FE+4, ;COUNT UP BIT NUMBER
; 3040 SKIP DPO, J/JFFOL ;LOOP TILL WE FIND THE BIT
; 3041 [AR]_FE ;GET ANSWER BACK
; 3042 [AR]_[AR].AND.# CLR LH,#/77 ;MASK TO 1 COPY
; 3043 AC[1]_[AR], NEXT INST ;STORE AND EXIT
; 3044

U 0514, 0706,3441,0301,4174,4007,0700,0200,0014,0012
U 0515, 1400,4223,0000,1174,4007,0700,0400,0000,1441
U 0706, 0534,4443,0000,4174,4007,0700,1000,0071,1764

U 0534, 0534,3445,0505,4174,4007,0520,1000,0041,0004
U 0535, 0747,3777,0003,4334,4057,0700,0000,0041,0000
U 0747, 0767,4251,0303,4374,4007,0700,0000,0000,0077
U 0767, 0100,3440,0303,1174,4156,4700,0400,0000,1441
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 3045 .TOC "ROTATES AND LOGICAL SHIFTS -- LSHC"
; 3046
; 3047 ;SHIFT CONNECTIONS WHEN THE SPECIAL FUNCTION "LSHC" IS DONE:
; 3048 ;
; 3049 ; !-! !-----!-----!-----!-----!-----!-----!-----!-----!
; 3050 ; !O!-->!0000! HIGH ORDER 36 BITS ! RAM FILE
; 3051 ; !-! !-----!-----!-----!-----!-----!-----!-----!-----!
; 3052 ; ^
; 3053 ; :
; 3054 ; :
; 3055 ; :
; 3056 ; !-----!-----!-----!-----!-----!-----!-----!-----!
; 3057 ; !0000! LOW ORDER 36 BITS ! Q-REGISTER
; 3058 ; !-----!-----!-----!-----!-----!-----!-----!-----!
; 3059 ; ^
; 3060 ; :
; 3061 ; !-!
; 3062 ; !O!
; 3063 ; !-!
; 3064 ;
; 3065 ;
; 3066 1464:
; 3067 LSHC: STEP SC, J/LSHCL
; 3068 1465: READ [AR], SC_-SHIFT-1
; 3069 STEP SC
; 3070 =0
; 3071 LSHCR: [BR]_[BR]*.5 LONG,STEP SC,LSHC,J/LSHCR
; 3072 [BR]_[BR]*2 LONG,J/LSHCX
; 3073
; 3074 =0
; 3075 LSHCL: [BR]_[BR]*2 LONG,LSHC,STEP SC,J/LSHCL
; 3076 [BR]_[BR]*2 LONG
; 3077 LSHCX: [BR]_[BR]*2 LONG
; 3078 AC_[BR], J/ASHCQ1
; 3079
```

```
U 1464, 0554, 4443, 0000, 4174, 4007, 0630, 2000, 0060, 0000
U 1465, 1006, 3333, 0003, 4174, 4007, 0700, 2000, 0031, 5777
U 1006, 0544, 4443, 0000, 4174, 4007, 0630, 2000, 0060, 0000

U 0544, 0544, 3446, 0505, 4174, 4057, 0630, 2000, 0060, 0000
U 0545, 1014, 3444, 0505, 4174, 4007, 0700, 0000, 0000, 0000

U 0554, 0554, 3444, 0505, 4174, 4057, 0630, 2000, 0060, 0000
U 0555, 1014, 3444, 0505, 4174, 4007, 0700, 0000, 0000, 0000
U 1014, 1033, 3444, 0505, 4174, 4007, 0700, 0000, 0000, 0000
U 1033, 1053, 3440, 0505, 0174, 4007, 0700, 0400, 0000, 0000
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
; 3080 .TOC "ROTATES AND LOGICAL SHIFTS -- ASHC"
; 3081
; 3082
; 3083 1466:
; 3084 ASHC: READ [AR], ;PUT AR ON DP
; 3085 SC_SHIFT, LOAD FE, ;PUT SHIFT IN BOTH SC AND FE
; 3086 SKIP ADR.EQ.0 ;SEE IF NULL SHIFT
; 3087 =0 Q AC[1], ;NOT NULL--GET LOW WORD
; 3088 J/ASHC1 ;CONTINUE BELOW
; 3089 NIDISP: NEXT INST ;NULL--ALL DONE
; 3090 ASHC1: [BR]_AC*.5 LONG, ;GET HIGH WORD
; 3091 ;AND SHIFT Q
; 3092 SKIP/SC ;SEE WHICH DIRECTION
; 3093 =0 [BR]_[BR]*.5, ;ADJUST POSITION
; 3094 SC_FE+S#, S#/1776, ;SUBTRACT 2 FROM FE
; 3095 J/ASHCL ;GO LEFT
; 3096 [BR]_[BR]*.5, ;ADJUST POSITION
; 3097 SC_S#-FE, S#/1776 ;SC -2-FE, SC_+ # OF STEPS
; 3098 =0 ;HERE TO GO RIGHT
; 3099 ASHCR: [BR]_[BR]*.5 LONG, ;GO RIGHT
; 3100 ASHC, ;SET DATA PATHS FOR ASHC (SEE DPE1)
; 3101 STEP SC, J/ASHCR ;COUNT THE STEP AND KEEP LOOPING
; 3102 [BR]_[BR]*2 LONG, ;PUT BACK WHERE IT GOES
; 3103 ASHC, J/ASHCX ;COMPLETE INSTRUCTION
; 3104
; 3105 =0
; 3106 ASHCL: [BR]_[BR]*2 LONG, ;GO LEFT
; 3107 ASHC, ASH AROV, ;SEE IF OVERFLOW
; 3108 STEP SC, J/ASHCL ;LOOP OVER ALL PLACES
; 3109 [BR]_[BR]*2 LONG, ;SHIFT BACK WHERE IT GOES
; 3110 ASHC, ASH AROV ;CAN STILL OVERFLOW
; 3111 ASHCX: AC_[BR]+[BR], 3T, ;PUT BACK HIGH WORD
; 3112 SKIP DPO ;SEE HOW TO FIX LOW SIGN
; 3113 =0 Q_Q.AND.#, #/377777, ;POSITIVE, CLEAR LOW SIGN
; 3114 HOLD RIGHT, J/ASHCQ1 ;GO STORE ANSWER
; 3115 Q_Q.OR.#, #/400000, ;NEGATIVE, SET LOW SIGN
; 3116 HOLD RIGHT ;IN LEFT HALF
; 3117 ASHCQ1: AC[1]_Q, NEXT INST ;PUT BACK Q AND EXIT
; 3118
```

Produced on Advanced Information Services Electronic Laser Printer, PKC/JES, DTN: 223-7881

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 3119 .TOC "ROTATES AND LOGICAL SHIFTS -- ROTC"  
; 3120  
; 3121 ;SHIFT CONNECTIONS WHEN THE SPECIAL FUNCTION "ROTC" IS DONE:  
; 3122 ;  
; 3123 ;  
; 3124 ; .....>!0000! HIGH ORDER 36 BITS ! RAM FILE  
; 3125 ; !-----!-----!  
; 3126 ; : : ^  
; 3127 ; : : :  
; 3128 ; : : :  
; 3129 ; : : :  
; 3130 ; !-----!-----!  
; 3131 ; ..!0000! LOW ORDER 36 BITS ! Q-REGISTER  
; 3132 ; !-----!-----!  
; 3133 ; : : ^  
; 3134 ; : : :  
; 3135 ; : : :  
; 3136 ; : : :  
; 3137 ; : : :  
; 3138 1470:  
; 3139 ROTC: STEP SC, J/ROTCL  
; 3140 1471: READ [AR], SC_-SHIFT-1  
; 3141 STEP SC  
; 3142 =0  
; 3143 ROTCR: [BR]_[BR]*.5 LONG,STEP SC,ROTC,J/ROTCR  
; 3144 [BR]_[BR]*2 LONG,J/LSHCX  
; 3145  
; 3146 =0  
; 3147 ROTCL: [BR]_[BR]*2 LONG,ROTC,STEP SC,J/ROTCL  
; 3148 [BR]_[BR]*2 LONG,  
; 3149 J/LSHCX  
; 3150
```

```
U 1470, 0742,4443,0000,4174,4007,0630,2000,0060,0000  
U 1471, 1062,3333,0003,4174,4007,0700,2000,0031,5777  
U 1062, 0644,4443,0000,4174,4007,0630,2000,0060,0000  
  
U 0644, 0644,3446,0505,4174,4077,0630,2000,0060,0000  
U 0645, 1014,3444,0505,4174,4007,0700,0000,0000,0000  
  
U 0742, 0742,3444,0505,4174,4077,0630,2000,0060,0000  
U 0743, 1014,3444,0505,4174,4007,0700,0000,0000,0000
```


; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 3151 .TOC "TEST GROUP"
; 3152
; 3153 .DCODE
; 3154
; 3155 ;SPECIAL MACROS USED ONLY IN B-FIELD OF TEST INSTRUCTIONS
; 3156 TN- "B/4"
; 3157 TNE "B/O"
; 3158 WORD-TNE "B/10" ;USED IN TIOE
; 3159 TNA "B/O"
; 3160 TNN "B/4"
; 3161 WORD-TNN "B/14" ;USED IN TION
; 3162 TZ- "B/5"
; 3163 TZE "B/1"
; 3164 TZA "B/1"
; 3165 TZN "B/5"
; 3166 TC- "B/6"
; 3167 TCE "B/2"
; 3168 TCA "B/2"
; 3169 TCN "B/6"
; 3170 TO- "B/7"
; 3171 TOE "B/3"
; 3172 TOA "B/3"
; 3173 TON "B/7"
; 3174
; 3175 600: I, J/DONE ;TRN- IS NOP
; 3176 I, J/DONE ;SO IS TLN-
; 3177 I, TNE, J/TDXX
; 3178 I, TNE, J/TSXX
; 3179 I, TNA, J/TDX
; 3180 I, TNA, J/TSX
; 3181 I, TNN, J/TDXX
; 3182 I, TNN, J/TSXX
; 3183
; 3184 610: I, J/DONE ;TDN- IS A NOP
; 3185 I, J/DONE ;TSN- ALSO
; 3186 R, TNE, J/TDXX
; 3187 R, TNE, J/TSXX
; 3188 R, TNA, J/TDX
; 3189 R, TNA, J/TSX
; 3190 R, TNN, J/TDXX
; 3191 R, TNN, J/TSXX
; 3192
; 3193 620: I, TZ-, J/TDX
; 3194 I, TZ-, J/TSX
; 3195 I, TZE, J/TDXX
; 3196 I, TZE, J/TSXX
; 3197 I, TZA, J/TDX
; 3198 I, TZA, J/TSX
; 3199 I, TZN, J/TDXX
; 3200 I, TZN, J/TSXX
; 3201
```

```
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
```

D 0630,	0005,1473,1100	; 3202	630:	R,	TZ-	J/TDX
D 0631,	0005,1472,1100	; 3203		R,	TZ-	J/TSX
D 0632,	0001,1475,1100	; 3204		R,	TZE,	J/TDXX
D 0633,	0001,1474,1100	; 3205		R,	TZE,	J/TSXX
D 0634,	0001,1473,1100	; 3206		R,	TZA,	J/TDX
D 0635,	0001,1472,1100	; 3207		R,	TZA,	J/TSX
D 0636,	0005,1475,1100	; 3208		R,	TZN,	J/TDXX
D 0637,	0005,1474,1100	; 3209		R,	TZN,	J/TSXX
		; 3210				
D 0640,	0006,1473,2100	; 3211	640:	I,	TC-	J/TDX
D 0641,	0006,1472,2100	; 3212		I,	TC-	J/TSX
D 0642,	0002,1475,2100	; 3213		I,	TCE,	J/TDXX
D 0643,	0002,1474,2100	; 3214		I,	TCE,	J/TSXX
D 0644,	0002,1473,2100	; 3215		I,	TCA,	J/TDX
D 0645,	0002,1472,2100	; 3216		I,	TCA,	J/TSX
D 0646,	0006,1475,2100	; 3217		I,	TCN,	J/TDXX
D 0647,	0006,1474,2100	; 3218		I,	TCN,	J/TSXX
		; 3219				
D 0650,	0006,1473,1100	; 3220	650:	R,	TC-	J/TDX
D 0651,	0006,1472,1100	; 3221		R,	TC-	J/TSX
D 0652,	0002,1475,1100	; 3222		R,	TCE,	J/TDXX
D 0653,	0002,1474,1100	; 3223		R,	TCE,	J/TSXX
D 0654,	0002,1473,1100	; 3224		R,	TCA,	J/TDX
D 0655,	0002,1472,1100	; 3225		R,	TCA,	J/TSX
D 0656,	0006,1475,1100	; 3226		R,	TCN,	J/TDXX
D 0657,	0006,1474,1100	; 3227		R,	TCN,	J/TSXX
D 0660,	0007,1473,2100	; 3228	660:	I,	TO-	J/TDX
D 0661,	0007,1472,2100	; 3229		I,	TO-	J/TSX
D 0662,	0003,1475,2100	; 3230		I,	TOE,	J/TDXX
D 0663,	0003,1474,2100	; 3231		I,	TOE,	J/TSXX
D 0664,	0003,1473,2100	; 3232		I,	TOA,	J/TDX
D 0665,	0003,1472,2100	; 3233		I,	TOA,	J/TSX
D 0666,	0007,1475,2100	; 3234		I,	TON,	J/TDXX
D 0667,	0007,1474,2100	; 3235		I,	TON,	J/TSXX
		; 3236				
D 0670,	0007,1473,1100	; 3237	670:	R,	TO-	J/TDX
D 0671,	0007,1472,1100	; 3238		R,	TO-	J/TSX
D 0672,	0003,1475,1100	; 3239		R,	TOE,	J/TDXX
D 0673,	0003,1474,1100	; 3240		R,	TOE,	J/TSXX
D 0674,	0003,1473,1100	; 3241		R,	TOA,	J/TDX
D 0675,	0003,1472,1100	; 3242		R,	TOA,	J/TSX
D 0676,	0007,1475,1100	; 3243		R,	TON,	J/TDXX
D 0677,	0007,1474,1100	; 3244		R,	TON,	J/TSXX
		; 3245				

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 3246          .UCODE
; 3247
; 3248          ;THESE 64 INSTRUCTIONS ARE DECODED BY MASK MODE(IMMEDIATE OR MEMORY)
; 3249          ; IN THE A FIELD, DISPATCH TO HERE ON THE J FIELD, AND RE-DISPATCH
; 3250          ; FOR THE MODIFICATION ON THE B FIELD.
; 3251
; 3252          ; ENTER WITH O,E OR (E) IN AR, B FIELD BITS 2 AND 3 AS FOLLOWS:
; 3253          ; 0 0  NO MODIFICATION
; 3254          ; 0 1  OS
; 3255          ; 1 0  COMPLEMENT
; 3256          ; 1 1  ONES
; 3257          ; THIS ORDER HAS NO SIGNIFICANCE EXCEPT THAT IT CORRESPONDS TO THE
; 3258          ; ORDER OF INSTRUCTIONS AT TGROUP.
; 3259
; 3260          ;THE BIT 1 OF THE B FIELD IS USED TO DETERMINE THE SENSE
; 3261          ; OF THE SKIP
; 3262          ; 1    SKIP IF AC.AND.MASK .NE. 0 (TXX- AND TXXN)
; 3263          ; 0    SKIP IF AC.AND.MASK .EQ. 0 (TXXA AND TXXE)
; 3264
; 3265          ;BIT 0 IS UNUSED AND MUST BE ZERO
; 3266
; 3267
; 3268          1472:
; 3269          TSX:  [AR]_[AR] SWAP          ;TSXX AND TLXX
; 3270          1473:
; 3271          TDX:  [BR]_0,TEST DISP      ; ALWAYS AND NEVER SKIP CASES
; 3272
; 3273          1474:
; 3274          TSXX: [AR]_[AR] SWAP          ;TSXE, TSXN, TLXE, TLXN
; 3275          1475:
; 3276          TDXX: [BR]_[AR].AND.AC,     ;TDXE, TDXN, TRXE, TRXN
; 3277          TEST DISP
; 3278
```

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

; T10KL.MCR[10,1141] 15:34 27-JULY-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, 1077,7441,0303,4174,4007,0700,0000,0000,0000

U 0016, 1117,6551,0303,0274,4007,0700,0000,0000,0000

U 0017, 1117,3551,0303,0274,4007,0700,0000,0000,0000

U 1077, 1117,4551,0303,0274,4007,0700,0000,0000,0000

U 1117, 0014,3440,0303,0174,4007,0700,0400,0000,0000

```
; 3279 ;TEST DISP DOES AN 8 WAY BRANCH BASED ON THE B-FIELD OF DROM
; 3280
; 3281 =1100
; 3282 TEST-TABLE:
; 3283
; 3284 ;CASE 0 & 4 -- TXNX
; 3285 TXXX: READ [BR], TXXX TEST, 3T, J/DONE
; 3286
; 3287 ;CASE 1 & 5 -- TXZ AND TXZX
; 3288 [AR]_.NOT.[AR],J/TXZX
; 3289
; 3290 ;CASE 2 & 6 -- TXC AND TXCX
; 3291 [AR]_[AR].XOR.AC,J/TDONE
; 3292
; 3293 ;CASE 3 & 7 -- TXO AND TXOX
; 3294 [AR]_[AR].OR.AC,J/TDONE
; 3295
; 3296 ;THE SPECIAL FUNCTION TXXX TEST CAUSES A MICROCODE SKIP IF
; 3297 ; AD.EQ.O AND DROM B IS 0-3 OR AD.NE.O AND DROM B IS 4-7.
; 3298
; 3299 TXZX: [AR]_[AR].AND.AC
; 3300 TDONE: AC_[AR],J/TXXX
; 3301 ; READ BR,TXXX TEST,J/DONE
; 3302
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 3303 .TOC "COMPARE -- CAI, CAM"
; 3304
; 3305 .DCODE
; 3306
; 3307 ;SPECIAL B-FIELD ENCODING USED BY SKIP-JUMP-COMPARE CLASS
; 3308 ; INSTRUCTIONS:
; 3309
; 3310 SJC- "B/0" ;NEVER
; 3311 SJCL "B/1" ;LESS
; 3312 SJCE "B/2" ;EQUAL
; 3313 SJCLE "B/3" ;LESS EQUAL
; 3314 SJCA "B/4" ;ALWAYS
; 3315 SJCGE "B/5" ;GREATER THAN OR EQUAL
; 3316 SJCN "B/6" ;NOT EQUAL
; 3317 SJCG "B/7" ;GREATER
; 3318
; 3319 .UCODE
; 3320
; 3321 ;COMPARE TABLE
; 3322 =1000
; 3323 SKIP-COMP-TABLE:
; 3324
; 3325 ;CASE 0 -- NEVER
; 3326 DONE
; 3327
; 3328 ;CASE 1 -- LESS
; 3329 READ [AR], SKIP DPO,J/DONE
; 3330
; 3331 ;CASE 2 -- EQUAL
; 3332 SKIPE: READ [AR], SKIP AD.EQ.O,J/DONE
; 3333
; 3334 ;CASE 3 -- LESS OR EQUAL
; 3335 READ [AR], SKIP AD.LE.O,J/DONE
; 3336
; 3337 ;CASE 4 -- ALWAYS
; 3338 VMA_[PC]+1, NEXT INST FETCH, FETCH
; 3339
; 3340 ;CASE 5 -- GREATER THAN OR EQUAL
; 3341 READ [AR], SKIP DPO,J/SKIP
; 3342
; 3343 ;CASE 6 -- NOT EQUAL
; 3344 READ [AR], SKIP AD.EQ.O,J/SKIP
; 3345
; 3346 ;CASE 7 -- GREATER
; 3347 READ [AR], SKIP AD.LE.O,J/SKIP
; 3348
```

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, 0260,3333,0003,4174,4007,0520,0000,0000,0000

U 0256, 0260,3333,0003,4174,4007,0621,0000,0000,0000

U 0257, 0260,3333,0003,4174,4007,0421,0000,0000,0000

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```

; 3349          .DCODE
D 0300, 0000, 1400, 2100 ; 3350 300: I,      SJC-,   J/DONE ;CAI
D 0301, 0001, 1476, 2100 ; 3351      I,      SJCL,   J/CAIM
D 0302, 0002, 1476, 2100 ; 3352      I,      SJCE,   J/CAIM
D 0303, 0003, 1476, 2100 ; 3353      I,      SJCLE,  J/CAIM
D 0304, 0004, 1476, 2100 ; 3354      I,      SJCA,   J/CAIM
D 0305, 0005, 1476, 2100 ; 3355      I,      SJCGE,  J/CAIM
D 0306, 0006, 1476, 2100 ; 3356      I,      SJCN,   J/CAIM
D 0307, 0007, 1476, 2100 ; 3357      I,      SJCG,   J/CAIM
; 3358
D 0310, 0000, 1476, 1100 ; 3359 310: R,      SJC-,   J/CAIM ;CAM
D 0311, 0001, 1476, 1100 ; 3360      R,      SJCL,   J/CAIM
D 0312, 0002, 1476, 1100 ; 3361      R,      SJCE,   J/CAIM
D 0313, 0003, 1476, 1100 ; 3362      R,      SJCLE,  J/CAIM
D 0314, 0004, 1476, 1100 ; 3363      R,      SJCA,   J/CAIM
D 0315, 0005, 1476, 1100 ; 3364      R,      SJCGE,  J/CAIM
D 0316, 0006, 1476, 1100 ; 3365      R,      SJCN,   J/CAIM
D 0317, 0007, 1476, 1100 ; 3366      R,      SJCG,   J/CAIM
; 3367          .UCODE
; 3368
; 3369          1476:
U 1476, 0250, 2551, 0303, 0274, 4003, 7701, 4000, 0000, 0000 ; 3370 CAIM:  [AR]_AC-[AR], 3T, SKIP-COMP DISP
; 3371
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

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, 1122, 0111, 0703, 4174, 4467, 0701, 0000, 0001, 0001
U 1122, 1127, 4443, 0000, 4174, 4007, 0700, 0200, 0003, 0002
U 1127, 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, 1122, 1111, 0703, 4174, 4467, 0701, 4000, 0001, 0001

```
; 3372 .TOC "ARITHMETIC SKIPS -- AOS, SOS, SKIP"  
; 3373 ;ENTER WITH (E) IN AR  
; 3374  
; 3375 .DCODE  
; 3376 330: R, SJC-, J/SKIPS ;NOT A NOP IF AC .NE. O  
; 3377 R, SJCL, J/SKIPS  
; 3378 R, SJCE, J/SKIPS  
; 3379 R, SJCLE, J/SKIPS  
; 3380 R, SJCA, J/SKIPS  
; 3381 R, SJCGE, J/SKIPS  
; 3382 R, SJCN, J/SKIPS  
; 3383 R, SJCG, J/SKIPS  
; 3384 .UCODE  
; 3385  
; 3386 1477:  
; 3387 SKIPS: FIX [AR] SIGN,  
; 3388 SKIP IF ACO  
; 3389 =0 AC [AR],SKIP-COMP DISP  
; 3390 SKIP-COMP DISP  
; 3391  
; 3392 .DCODE  
; 3393 350: RW, SJC-, J/AOS  
; 3394 RW, SJCL, J/AOS  
; 3395 RW, SJCE, J/AOS  
; 3396 RW, SJCLE, J/AOS  
; 3397 RW, SJCA, J/AOS  
; 3398 RW, SJCGE, J/AOS  
; 3399 RW, SJCN, J/AOS  
; 3400 RW, SJCG, J/AOS  
; 3401 .UCODE  
; 3402  
; 3403 1431:  
; 3404 AOS: [AR][AR]+1, 3T, AD FLAGS  
; 3405 XOS: START WRITE  
; 3406 MEM WRITE, MEM_[AR], J/SKIPS  
; 3407  
; 3408 .DCODE  
; 3409 370: RW, SJC-, J/SOS  
; 3410 RW, SJCL, J/SOS  
; 3411 RW, SJCE, J/SOS  
; 3412 RW, SJCLE, J/SOS  
; 3413 RW, SJCA, J/SOS  
; 3414 RW, SJCGE, J/SOS  
; 3415 RW, SJCN, J/SOS  
; 3416 RW, SJCG, J/SOS  
; 3417 .UCODE  
; 3418  
; 3419 1437:  
; 3420 SOS: [AR][AR]-1, 3T, AD FLAGS, J/XOS  
; 3421
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 3422 .TOC "CONDITIONAL JUMPS -- JUMP, AOJ, SOJ, AOBJ"  
; 3423 ; ENTER WITH E IN AR  
; 3424  
; 3425 =1000  
; 3426 JUMP-TABLE:  
; 3427  
; 3428 ;CASE 0 -- NEVER  
; 3429 AC_[BR], NEXT INST  
; 3430  
; 3431 ;CASE 1 -- LESS  
; 3432 AC_[BR] TEST, SKIP DPO, J/JUMP-  
; 3433  
; 3434 ;CASE 2 -- EQUAL  
; 3435 AC_[BR] TEST, SKIP AD.EQ.O, J/JUMP-  
; 3436  
; 3437 ;CASE 3 -- LESS THAN OR EQUAL  
; 3438 AC_[BR] TEST, SKIP AD.LE.O, J/JUMP-  
; 3439  
; 3440 ;CASE 4 -- ALWAYS  
; 3441 JMPA: AC_[BR], J/JUMPA  
; 3442  
; 3443 ;CASE 5 -- GREATER THAN OR EQUAL TO  
; 3444 AC_[BR] TEST, SKIP DPO, J/JUMPA  
; 3445  
; 3446 ;CASE 6 -- NOT EQUAL  
; 3447 AC_[BR] TEST, SKIP AD.EQ.O, J/JUMPA  
; 3448  
; 3449 ;CASE 7 -- GREATER  
; 3450 AC_[BR] TEST, SKIP AD.LE.O, J/JUMPA  
; 3451  
; 3452 =0  
; 3453 JUMP-: DONE  
; 3454 JUMPA  
; 3455  
; 3456 =0  
; 3457 JUMPA: JUMPA  
; 3458 DONE  
; 3459  
; 3460
```


; T1OKL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```

; 3461      .DCODE
D 0320, 0000,1400,2100 ; 3462 320: I,      SJC-,   J/DONE
D 0321, 0001,1440,2100 ; 3463      I,      SJCL,   J/JUMP
D 0322, 0002,1440,2100 ; 3464      I,      SJCE,   J/JUMP
D 0323, 0003,1440,2100 ; 3465      I,      SJCLE,  J/JUMP
D 0324, 0004,1520,2100 ; 3466      I,      SJCA,   J/JRST
D 0325, 0005,1440,2100 ; 3467      I,      SJCGE,  J/JUMP
D 0326, 0006,1440,2100 ; 3468      I,      SJCN,   J/JUMP
D 0327, 0007,1440,2100 ; 3469      I,      SJCG,   J/JUMP
; 3470      .UCODE
; 3471
; 3472 1440:
U 1440, 0270,3771,0005,0276,6003,7700,0000,0000,0000 ; 3473 JUMP:  [BR]_AC,JUMP DISP
; 3474
; 3475      .DCODE
D 0340, 0000,1611,3000 ; 3476 340: I-PF,   SJC-,   J/AOJ
D 0341, 0001,1611,2100 ; 3477      I,      SJCL,   J/AOJ
D 0342, 0002,1611,2100 ; 3478      I,      SJCE,   J/AOJ
D 0343, 0003,1611,2100 ; 3479      I,      SJCLE,  J/AOJ
D 0344, 0004,1611,2100 ; 3480      I,      SJCA,   J/AOJ
D 0345, 0005,1611,2100 ; 3481      I,      SJCGE,  J/AOJ
D 0346, 0006,1611,2100 ; 3482      I,      SJCN,   J/AOJ
D 0347, 0007,1611,2100 ; 3483      I,      SJCG,   J/AOJ
; 3484      .UCODE
; 3485
; 3486 1611:
U 1611, 0270,0551,0705,0274,4463,7702,0000,0001,0001 ; 3487 AOJ:  [BR]_AC+1, AD FLAGS, 4T, JUMP DISP
; 3488
; 3489      .DCODE
D 0360, 0000,1542,3000 ; 3490 360: I-PF,   SJC-,   J/SOJ
D 0361, 0001,1542,2100 ; 3491      I,      SJCL,   J/SOJ
D 0362, 0002,1542,2100 ; 3492      I,      SJCE,   J/SOJ
D 0363, 0003,1542,2100 ; 3493      I,      SJCLE,  J/SOJ
D 0364, 0004,1542,2100 ; 3494      I,      SJCA,   J/SOJ
D 0365, 0005,1542,2100 ; 3495      I,      SJCGE,  J/SOJ
D 0366, 0006,1542,2100 ; 3496      I,      SJCN,   J/SOJ
D 0367, 0007,1542,2100 ; 3497      I,      SJCG,   J/SOJ
; 3498      .UCODE
; 3499
; 3500
; 3501 1542:
U 1542, 0270,2551,0705,0274,4463,7702,4000,0001,0001 ; 3501 SOJ:  [BR]_AC-1, AD FLAGS, 4T, JUMP DISP
; 3502
; 3503      .DCODE
D 0252, 0005,1547,2100 ; 3504 252: I,      SJCGE,  J/AOBJ
D 0253, 0001,1547,2100 ; 3505      I,      SJCL,   J/AOBJ
; 3506      .UCODE
; 3507
; 3508 1547:
; 3509 AOBJ:  [BR]_AC+1000001, ;ADD 1 TO BOTH HALF WORDS
; 3510      INH CRY18, 3T, ;NO CARRY INTO LEFT HALF
; 3511      JUMP DISP ;HANDLE EITHER AOBJP OR AOBJN
; 3512
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

D 0254, 0000, 1520, 6000

D 0255, 0000, 1540, 2100

U 1520, 0110, 3441, 0301, 4170, 4156, 4700, 0200, 0014, 0012

U 1521, 0110, 3441, 0301, 4170, 4156, 4700, 0200, 0014, 0012

U 1522, 0150, 1113, 0701, 4170, 4007, 0700, 4200, 0004, 0012

U 1523, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1524, 1004, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000

U 1525, 2102, 3443, 0300, 4174, 4007, 0700, 0200, 0004, 0012

U 1526, 0320, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000

U 1527, 1034, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000

U 1530, 1024, 1113, 0701, 4170, 4007, 0040, 4200, 0004, 0012

U 1531, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1532, 0300, 1113, 0701, 4170, 4007, 0040, 4200, 0004, 0012

U 1533, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1534, 1044, 4443, 0000, 4174, 4007, 0340, 0000, 0000, 0000

U 1535, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1536, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1537, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

```
; 3513 .TOC "AC DECODE JUMPS -- JRST, JFCL"
; 3514
; 3515 .DCODE
; 3516 254: I,VMA/O, AC DISP, J/JRST ;DISPATCHES TO 1 OF 16
; 3517 ; PLACES ON AC BITS
; 3518 I. J/JFCL
; 3519 .UCODE
; 3520
; 3521 ;JRST DISPATCHES TO ONE OF 16 LOC'NS ON AC BITS
; 3522
; 3523 =0000
; 3524 1520:
; 3525 JRST: JUMPA ;(0) JRST O,
; 3526 1521: JUMPA ;(1) PORTAL IS SAME AS JRST
; 3527 1522: VMA [PC]-1, START READ, ;(2) JRSTF
; 3528 J/JRSTF
; 3529 1523: UUU ;(3)
; 3530 1524: SKIP KERNEL, J/HALT ;(4) HALT
; 3531 1525:
; 3532 XJRSTFO: VMA [AR], START READ, ;(5) XJRSTF
; 3533 J/XJRSTF
; 3534 1526: SKIP KERNEL, J/XJEN ;(6) XJEN
; 3535 1527: SKIP KERNEL, J/XPCW ;(7) XPCW
; 3536 1530: VMA [PC]-1, START READ, ;(10)
; 3537 SKIP IO LEGAL, J/JRST10
; 3538 1531: UUU ;(11)
; 3539 1532: VMA [PC]-1, START READ, ;(12) JEN
; 3540 SKIP IO LEGAL, J/JEN
; 3541 1533: UUU ;(13)
; 3542 1534: SKIP KERNEL, J/SFM ;(14) SFM
; 3543 1535: UUU ;(15)
; 3544 1536: UUU ;(16)
; 3545 1537: UUU ;(17)
; 3546
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

U 0150, 1142,3771,0002,4365,5217,0700,0210,0000,0002
U 0152, 0110,3441,0301,4170,4156,4700,0200,0014,0012
U 1142, 0030,4443,0000,2174,4006,6700,0000,0000,0000
U 0030, 0002,3773,0000,2274,4464,1700,0000,0001,0004
U 0032, 0002,3333,0002,4174,4464,1700,0000,0001,0004
U 0034, 1147,0551,0202,2270,4007,0700,0200,0004,0012
U 0036, 1147,3443,0200,4174,4007,0700,0200,0004,0112
U 1147, 1142,3771,0002,4365,5217,0700,0200,0000,0002

```
; 3547 =0*
; 3548 JRSTF: MEM READ, ;WAIT FOR DATA
; 3549 [HR]_MEM, ;STICK IN HR
; 3550 LOAD_INST EA, ;LOAD @ AND XR
; 3551 CALL [JRSTO] ;COMPUTE EA AGAIN
; 3552 JUMPA ;JUMP
; 3553
; 3554 JRSTO: EA MODE DISP ;WHAT TYPE OF EA?
; 3555 =100*
; 3556 READ XR, ;INDEXED
; 3557 LOAD_FLAGS, ;GET FLAGS FROM XR
; 3558 UPDATE_USER, ;ALLOW USER TO SET
; 3559 RETURN [2] ;ALL DONE
; 3560 READ [HR], ;PLAIN
; 3561 LOAD_FLAGS, ;LOAD FLAGS FROM INST
; 3562 UPDATE_USER, ;ALLOW USER TO SET
; 3563 RETURN [2] ;RETURN
; 3564 [HR]_[HR]+XR, ;BOTH
; 3565 LOAD_VMA, ;FETCH IND WORD
; 3566 START_READ, ;START MEM CYCLE
; 3567 J/JRST1 ;CONTINUE BELOW
; 3568 VMA [HR], ;INDIRECT
; 3569 START_READ, ;FETCH IND WORD
; 3570 PXCT EA, ;SETUP PXCT STUFF
; 3571 J/JRST1 ;CONTINUE BELOW
; 3572 JRST1: MEM READ, ;WAIT FOR DATA
; 3573 [HR]_MEM, ;LOAD THE HR
; 3574 LOAD_INST EA, ;LOAD @ AND XR
; 3575 J/JRSTO ;LOOP BACK
; 3576
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 99
AC DECODE JUMPS -- JRST, JFCL

```
U 1004, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3577 =0
U 1005, 1155,3441,0301,4174,4007,0700,0000,0000,0000 ; 3578 HALT: UUO ;USER MODE
U 1155, 0104,4751,1217,4374,4007,0700,0000,0000,0001 ; 3579 [PC]_[AR] ;EXEC MODE--CHANGE PC
; 3580 HALT [HALT] ;HALT INSTRUCTION
; 3581
; 3582 =0
U 1024, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3583 JRST10: UUO
U 1025, 0303,4443,0000,4174,4007,0700,0000,0000,0000 ; 3584 J/JEN2 ;DISMISS INTERRUPT
; 3585 =0000
U 0300, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3586 JEN: UUO ; FLAGS
; 3587 MEM READ,
; 3588 [HR] MEM, ;GET INST
; 3589 LOAD INST EA, ;LOAD XR & @
; 3590 CALL [JRSTO] ;COMPUTE FLAGS
; 3591 =0011
U 0301, 1142,3771,0002,4365,5217,0700,0210,0000,0002 ; 3592 JEN2: DISMISS ;DISMISS INTERRUPT
U 0303, 2452,4553,1400,4374,4007,0331,0010,0007,7400 ; 3593 =0111 CALL LOAD PI ;RELOAD PI HARDWARE
U 0307, 3623,3770,1416,4344,4007,0700,0010,0000,0000 ; 3594 =1111 JUMPA ;GO JUMP
U 0317, 0110,3441,0301,4170,4156,4700,0200,0014,0012 ; 3595 =
; 3596
; 3597 1540:
; 3598 JFCL: JFCL FLAGS, ;ALL DONE IN HARDWARE
; 3599 SKIP JFCL, ;SEE IF SKIPS
; 3600 3T, ;ALLOW TIME
; 3601 J/JUMP- ;JUMP IF WE SHOULD
; 3602
```

Produced on Advanced Information Services Electronic Laser Printer. PRO11555, DTN: 223-7981

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 3603 .TOC "EXTENDED ADDRESSING INSTRUCTIONS"
; 3604
; 3605 =0000
U 0320, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3606 XJEN: UUO ;HERE IF USER MODE
U 0321, 2452,4553,1400,4374,4007,0331,0010,0007,7400 ; 3607 DISMISS ;CLEAR HIGHEST INTERRUPT
U 0325, 0335,3333,0012,4174,4437,0700,0000,0000,0000 ; 3608 =0101 READ [MASK], LOAD PI ;NO MORE INTERRUPTS
; 3609 =1101 ABORT MEM CYCLE, ;AVOID INTERRUPT PAGE FAIL
U 0335, 1525,4223,0000,4364,4277,0700,0200,0000,0010 ; 3610 J/XJRSTFO ;START READING FLAG WORD
; 3611 =
; 3612
U 2102, 2105,3771,0005,4365,5007,0700,0200,0000,0002 ; 3613 XJRSTF: MEM READ, [BR]_MEM ;PUT FLAGS IN BR
; 3614 [AR]_[AR]+1, ;INCREMENT ADDRESS
; 3615 LOAD VMA, ;PUT RESULT IN VMA
U 2105, 2611,0111,0703,4174,4007,0700,0200,0004,0012 ; 3616 START READ ;START MEMORY
; 3617 MEM READ, [PC]_MEM, ;PUT DATA IN PC
U 2611, 2616,3771,0001,4361,5007,0700,0200,0000,0002 ; 3618 HOLD LEFT ;IGNORE SECTION NUMBER
; 3619 READ [BR], LOAD FLAGS, ;LOAD NEW FLAGS
U 2616, 2677,3333,0005,4174,4467,0700,0000,0001,0004 ; 3620 UPDATE USER ;BUT HOLD USER FLAG
; 3621 PISET: [FLG]_[FLG].AND.NOT.#, ;CLEAR PI CYCLE
U 2677, 0405,5551,1313,4374,4007,0700,0000,0001,0000 ; 3622 FLG.PI/1. J/PIEXIT ;RELOAD PI HARDWARE
; 3623 ; INCASE THIS IS AN
; 3624 ; INTERRUPT INSTRUCTION
; 3625
; 3626 =0
U 1034, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3627 XPCW: UUO ;USER MODE
U 1035, 0060,4521,1205,4074,4007,0700,0000,0000,0000 ; 3628 [BR]_FLAGS ;PUT FLAGS IN BR
; 3629 =0*0
U 0060, 3702,3443,0300,4174,4007,0700,0210,0003,0012 ; 3630 PIXPCW: VMA [AR], START WRITE, ;STORE FLAGS
; 3631 CALL [STOBR] ;PUT BR IN MEMORY
; 3632 =1*0 VMA [AR]+1, LOAD VMA,
; 3633 START WRITE, ;PREPEARE TO STORE PC
U 0064, 3703,0111,0703,4170,4007,0700,0210,0003,0012 ; 3634 CALL [STOPC] ;PUT PC IN MEMORY
; 3635 =1*1 [AR]_[AR]+1, ;DO NEW PC PART
U 0065, 2102,0111,0703,4174,4007,0700,0200,0004,0002 ; 3636 START READ, J/XJRSTF
; 3637 =
; 3638
; 3639 =0
U 1044, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 3640 SFM: UUO
U 1045, 2710,3443,0300,4174,4007,0700,0200,0003,0012 ; 3641 VMA [AR], START WRITE ;STORE FLAGS
U 2710, 0455,4521,1203,4074,4007,0700,0000,0000,0000 ; 3642 [AR]_FLAGS, J/STORE ;STORE AND EXIT
; 3643
```

Produced on Advanced Information Services Electronic Laser Printer, PKOJES6, DTN: 223-7881

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 101
XCT

```
D 0256, 0000, 1541, 1100
; 3644 .TOC "XCT"
; 3645
; 3646 .DCODE
; 3647 256: R, J/XCT ;OPERAND FETCHED AS DATA
; 3648 .UCODE
; 3649
; 3650 1541:
; 3651 XCT: SKIP KERNEL ;SEE IF MAY BE PXCT
; 3652 =0
; 3653 XCT1A: [HR] [AR], ;STUFF INTO HR
; 3654 DBUS/DP, ;PLACE ON DBUS FOR IR
; 3655 LOAD INST, ;LOAD IR, AC, XR, ETC.
; 3656 PXCT/E1, ;ALLOW XR TO BE PREVIOUS
; 3657 J/XCT1 ;CONTINUE BELOW
; 3658
; 3659 READ [HR], ;LOAD PXCT FLAGS
; 3660 LOAD PXCT, ;
; 3661 J/XCT1A ;CONTINUE WITH NORMAL FLOW
; 3662
; 3663 XCT1: WORK[YSAVE]_[HR] CLR LH, ;SAVE FOR IO INSTRUCTIONS
; 3664 J/XCT2 ;GO EXECUTE IT
; 3665
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
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, 2712,3771,0005,4365,5007,0700,0200,0000,0002

U 2712, 1156,0551,1504,0274,4407,0311,0200,0003,0712

U 1544, 2712,3741,0105,4074,4467,0700,0000,0005,0000

U 1156, 0220,3333,0005,4175,5003,7701,0200,0000,0002

U 1157, 2713,3333,0005,4175,5007,0701,0200,0000,0002

U 2713, 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

```
; 3666 .TOC "STACK INSTRUCTIONS -- PUSHJ, PUSH, POP, POPJ"
; 3667
; 3668 .DCODE
; 3669 260: I, B/O, J/PUSHJ
; 3670 IR, B/2, J/PUSH
; 3671 I, B/2, J/POP
; 3672 I, J/POPJ
; 3673 .UCODE
; 3674
; 3675 ;ALL START WITH E IN AR
; 3676 1543:
; 3677 PUSH: MEM READ, ;PUT MEMOP IN BR
; 3678 [BR] MEM ;
; 3679 PUSH1: [ARX] AC+1000001, ;BUMP BOTH HALVES OF AC
; 3680 INH CRY18, ;NO CARRY
; 3681 LOAD VMA, ;START TO STORE ITEM
; 3682 START WRITE, ;START MEM CYCLE
; 3683 PXCT STACK WORD, ;THIS IS THE STACK DATA WORD
; 3684 3T, ;ALLOW TIME
; 3685 SKIP CRYO, ;GO TO STMAC, SKIP IF PDL OV
; 3686 J/STMAC ;
; 3687
; 3688 1544:
; 3689 PUSHJ: [BR] PC WITH FLAGS, ;COMPUTE UPDATED FLAGS
; 3690 CLR FPD, ;CLEAR FIRST-PART-DONE
; 3691 J/PUSH1 ; AND JOIN PUSH CODE
; 3692
; 3693 =0
; 3694 STMAC: MEM WRITE, ;WAIT FOR MEMORY
; 3695 MEM [BR], ;STORE BR ON STACK
; 3696 B DISP, ;SEE IF PUSH OR PUSHJ
; 3697 J/JSTAC ;BELOW
; 3698 ;WE MUST STORE THE STACK WORD PRIOR TO SETTING PDL OV IN CASE OF
; 3699 ; PAGE FAIL.
; 3700 MEM WRITE, ;WAIT FOR MEMORY
; 3701 MEM [BR] ;STORE BR
; 3702 SETPDL: SET PDL OV, ;OVERFLOW
; 3703 B DISP, ;SEE IF PUSH OR PUSHJ
; 3704 J/JSTAC ;BELOW
; 3705
; 3706 =00
; 3707 JSTAC: [PC]_[AR], ;PUSHJ--LOAD PC
; 3708 LOAD VMA, ;LOAD ADDRESS
; 3709 FETCH ;GET NEXT INST
; 3710 JSTAC1: AC [ARX], ;STORE BACK STACK PTR
; 3711 NEXT INST ;DO NEXT INST
; 3712 AC [ARX], ;UPDATE STACK POINTER
; 3713 J/DONE ;DO NEXT INST
; 3714 =
; 3715
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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, 2714, 3771, 0004, 0276, 6007, 0701, 0200, 0004, 0712 ; 3716 1545:
; 3717 POP: [ARX] AC, ;GET POINTER
; 3718 LOAD VMA, ;ADDRESS OF STACK WORD
; 3719 START READ, 3T, ;START CYCLE
; 3720 PXCT STACK WORD ;FOR PXCT
; 3721
; 3722 MEM READ, ;LOAD BR (QUIT IF PAGE FAIL)
; 3723 [BR]_MEM ;STACK WORD TO BR
; 3724
; 3725 [ARX]_[ARX]+#, ;UPDATE POINTER
; 3726 #/777777, ;-1 IN EACH HALF
; 3727 INH CRY18, 3T, ;BUT NO CARRY
; 3728 SKIP CRYO ;SEE IF OVERFLOW
; 3729
; 3730 =0 VMA [AR], ;EFFECTIVE ADDRESS
; 3731 PXCT DATA, ;FOR PXCT
; 3732 START WRITE, ;WHERE TO STORE RESULT
; 3733 J/POPX1 ;OVERFLOW
; 3734
; 3735 VMA [AR], ;EFFECTIVE ADDRESS
; 3736 PXCT DATA, ;FOR PXCT
; 3737 START WRITE ;WHERE TO STORE RESULT
; 3738
; 3739 MEM WRITE, ;WAIT FOR MEM
; 3740 MEM [BR], ;STORE BR
; 3741 B DISP, ;POP OR POPJ?
; 3742 J/JSTAC ;STORE POINTER
; 3743
; 3744
; 3745 POPX1: MEM WRITE, ;WAIT FOR MEMORY
; 3746 MEM [BR], ;STORE BR
; 3747 J/SETPDL ;GO SET PDL OV
; 3748
; 3749
; 3750 1546:
; 3751 POPJ: [ARX] AC, ;GET POINTER
; 3752 LOAD VMA, ;POINT TO STACK WORD
; 3753 PXCT STACK WORD, 3T, ;FOR PXCT
; 3754 START READ ;START READ
; 3755 [ARX]_[ARX]+#, ;UPDATE POINTER
; 3756 #/777777, ;-1 IN BOTH HALFS
; 3757 INH CRY18, 3T, ;INHIBIT CARRY 18
; 3758 SKIP CRYO ;SEE IF OVERFLOW
; 3759 =0 SET PDL OV ;SET OVERFLOW
; 3760 MEM READ, [PC]_MEM, ;STICK DATA IN PC
; 3761 HOLD LEFT, ;NO FLAGS
; 3762 J/JSTAC1 ;STORE POINTER
; 3762
```


; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 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, 2721,3770,0303,4344,0007,0700,0000,0000,0000

U 2721, 1164,3771,0005,0276,6007,0521,0000,0000,0000

U 1164, 1166,0113,0503,0174,4407,0521,0400,0000,0000

U 1165, 1170,0113,0503,0174,4407,0521,0400,0000,0000

U 1166, 0100,4443,0000,4174,4156,4700,0000,0000,0000

U 1167, 0603,4443,0000,4174,4467,0700,0000,0001,2000

U 1170, 0603,4443,0000,4174,4467,0700,0000,0001,2000

U 1171, 0100,4443,0000,4174,4156,4700,0000,0000,0000

```
; 3763 .TOC "STACK INSTRUCTIONS -- ADJSP"
; 3764
; 3765 .DCODE
; 3766 105: I-PF, B/O, J/ADJSP
; 3767 .UCODE
; 3768
; 3769 1551:
; 3770 ADJSP: [AR]_[AR] SWAP, ;MAKE 2 COPIES OF RH
; 3771 HOLD RIGHT
; 3772 [BR]_AC, ;READ AC, SEE IF MINUS
; 3773 3T.
; 3774 SKIP DPO
; 3775 =0 AC [BR]+[AR], ;UPDATE AC
; 3776 INH CRY18, ;NO CARRY
; 3777 SKIP DPO, ;SEE IF STILL OK
; 3778 3T. ;ALLOW TIME
; 3779 J/ADJSP1 ;TEST FOR OFLO
; 3780 AC [BR]+[AR], ;UPDATE AC
; 3781 INH CRY18, ;NO CARRY
; 3782 SKIP DPO, ;SEE IF STILL MINUS
; 3783 3T. ;ALLOW TIME FOR SKIP
; 3784 J/ADJSP2 ;CONTINUE BELOW
; 3785
; 3786 =0
; 3787 ADJSP1: NEXT INST ;NO OVERFLOW
; 3788 SET PDL OV, ;SET PDL OV
; 3789 J/NIDISP ;GO DO NICOND DISP
; 3790
; 3791 =0
; 3792 ADJSP2: SET PDL OV, ;SET PDL OV
; 3793 J/NIDISP ;GO DO NICOND DISP
; 3794 NEXT INST ;NO OVERFLOW
; 3795
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```

; 3796 .TOC "SUBROUTINE CALL/RETURN -- JSR, JSP, JSA, JRA"
; 3797
; 3798 .DCODE
D 0264, 0000,1552,2100 ; 3799 264: I, J/JSR
D 0265, 0000,1550,2100 ; 3800 I, J/JSP
D 0266, 0000,1554,2100 ; 3801 I, J/JSA
D 0267, 0000,1555,2100 ; 3802 I, J/JRA
; 3803 .UCODE
; 3804
; 3805 1550:
U 1550, 2722,3741,0105,4074,4007,0700,0000,0000,0000 ; 3806 JSP: [BR]_PC WITH FLAGS ;GET PC WITH FLAGS
; 3807 CLR FPD, ;CLEAR FIRST-PART-DONE
; 3808 AC [BR], ;STORE FLAGS
; 3809 J/JUMPA ;GO JUMP
; 3810
; 3811 1552:
U 1552, 2723,3741,0105,4074,4467,0700,0000,0005,0000 ; 3812 JSR: [BR]_PC WITH FLAGS, ;GET PC WITH FLAGS
; 3813 CLR FPD ;CLEAR FIRST-PART-DONE
; 3814 VMA [AR], ;EFFECTIVE ADDRESS
U 2723, 2724,3443,0300,4174,4007,0700,0200,0003,0012 ; 3815 START WRITE ;STORE OLD PC WORD
; 3816 MEM WRITE, ;WAIT FOR MEMORY
; 3817 MEM [BR] ;STORE
; 3818 [PC]_[AR]+1000001, ;PC _ E+1
; 3819 HOLD LEFT, ;NO JUNK IN LEFT
; 3820 3T, ;ALLOW TIME FOR DBM
U 2724, 2725,3333,0005,4175,5007,0701,0200,0000,0002 ; 3821 J/START ;START AT E+1
; 3822
; 3823
; 3824 1554:
U 1554, 2726,3441,0305,4174,4007,0700,0200,0003,0002 ; 3825 JSA: [BR]_[AR], ;SAVE E
; 3826 START WRITE ;START TO STORE
U 2726, 0130,3770,0304,4344,4007,0700,0000,0000,0000 ; 3827 [ARX]_[AR] SWAP ;ARX LEFT _ E
; 3828 =0*0 [AR]_AC, ;GET OLD AC
; 3829 CALL [IBPX] ;SAVE AR IN MEMORY
U 0130, 3070,3771,0003,0276,6007,0700,0010,0000,0000 ; 3830 =1*0 [ARX]_[PC], ;ARX NOW HAS E,,PC
; 3831 HOLD LEFT, ;
; 3832 CALL [AC ARX] ;GO PUT ARX IN AC
; 3833 =1*1 [PC]_[BR]+1000001, ;NEW PC
; 3834 3T, ;ALLOW TIME
; 3835 HOLD LEFT, ;NO JUNK IN PC LEFT
U 0134, 3704,3441,0104,4170,4007,0700,0010,0000,0000 ; 3836 J/START ;START AT E+1
; 3837 =
; 3838
; 3839 1555:
U 1555, 2727,3771,0005,0276,6007,0700,0000,0000,0000 ; 3840 JRA: [BR]_AC ;GET AC
U 2727, 2730,3770,0505,4344,4007,0700,0000,0000,0000 ; 3841 [BR]_[BR] SWAP ;OLD E IN BR RIGHT
; 3842 VMA [BR], ;LOAD VMA
U 2730, 2731,3443,0500,4174,4007,0700,0200,0004,0012 ; 3843 START READ ;FETCH SAVED AC
; 3844 MEM READ, ;WAIT FOR MEMORY
; 3845 [BR] MEM, ;LOAD BR WITH SAVE AC
U 2731, 0274,3771,0005,4365,5007,0700,0200,0000,0002 ; 3846 J/JMPA ;GO JUMP
; 3847
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```

; 3848 .TOC "ILLEGAL INSTRUCTIONS AND UOO'S"
; 3849 ;LUOO'S TRAP TO CURRENT CONTEXT
; 3850
; 3851 .DCODE
D 0030, 0000,1557,2100 ; 3852 O30: I, B/0, J/LUOO
D 0031, 0001,1557,2100 ; 3853 I, B/1, J/LUOO
D 0032, 0002,1557,2100 ; 3854 I, B/2, J/LUOO
D 0033, 0003,1557,2100 ; 3855 I, B/3, J/LUOO
D 0034, 0004,1557,2100 ; 3856 I, B/4, J/LUOO
D 0035, 0005,1557,2100 ; 3857 I, B/5, J/LUOO
D 0036, 0006,1557,2100 ; 3858 I, B/6, J/LUOO
D 0037, 0007,1557,2100 ; 3859 I, B/7, J/LUOO
; 3860
; 3861 ;MONITOR UOO'S -- TRAP TO EXEC
; 3862
; 3863 O40: I, J/MUOO ;CALL
; 3864 I, J/MUOO ;INIT
; 3865 I, J/MUOO
; 3866 I, J/MUOO
; 3867 I, J/MUOO
; 3868 I, J/MUOO
; 3869 I, J/MUOO
; 3870 I, J/MUOO ;CALLI
; 3871 I, J/MUOO ;OPEN
; 3872 I, J/MUOO ;TTCALL
; 3873 I, J/MUOO
; 3874 I, J/MUOO
; 3875 I, J/MUOO
; 3876 I, J/MUOO ;RENAME
; 3877 I, J/MUOO ;IN
; 3878 I, J/MUOO ;OUT
; 3879 I, J/MUOO ;SETSTS
; 3880 I, J/MUOO ;STATO
; 3881 I, J/MUOO ;GETSTS
; 3882 I, J/MUOO ;STATZ
; 3883 I, J/MUOO ;INBUF
; 3884 I, J/MUOO ;OUTBUF
; 3885 I, J/MUOO ;INPUT
; 3886 I, J/MUOO ;OUTPUT
; 3887 I, J/MUOO ;CLOSE
; 3888 I, J/MUOO ;RELEAS
; 3889 I, J/MUOO ;MTAPE
; 3890 I, J/MUOO ;UGETF
; 3891 I, J/MUOO ;USETI
; 3892 I, J/MUOO ;USETO
; 3893 I, J/MUOO ;LOOKUP
; 3894 I, J/MUOO ;ENTER
; 3895
; 3896 ;EXPANSION OPCODES
; 3897
; 3898 100: I, J/UOO ;UJEN
; 3899 I, J/UOO101
; 3900 I, J/UOO102 ;GFAD
; 3901 I, J/UOO103 ;GFSB
; 3902
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

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

```
; 3903 ;RESERVED OPCODES
; 3904
; 3905 000: I, J/UUU
; 3906 104: I, J/JSYS ;JSYS
; 3907 106: I, J/UUU106 ;GFMP
; 3908 I, J/UUU107 ;GFDV
; 3909 130: I, B/0, J/FP-LONG ;UFA
; 3910 I, B/1, J/FP-LONG ;DFN
; 3911 141: I, B/2, J/FP-LONG ;FADL
; 3912 151: I, B/3, J/FP-LONG ;FSBL
; 3913 161: I, B/4, J/FP-LONG ;FMPL
; 3914 171: I, B/5, J/FP-LONG ;FDVL
; 3915 247: I, J/UUU247 ;RESERVED
; 3916 .UCODE
; 3917
; 3918 1661:
; 3919 UUU101: UUU
; 3920 1662:
; 3921 UUU102: UUU
; 3922 1663:
; 3923 UUU103: UUU
; 3924 1664:
; 3925 JSYS: UUU
; 3926 1666:
; 3927 UUU106: UUU
; 3928 1667:
; 3929 UUU107: UUU
; 3930 1660:
; 3931 FP-LONG:UUU
; 3932 1665:
; 3933 UUU247: UUU
; 3934
```

U 1661, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1662, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1663, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1664, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1666, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1667, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1660, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1665, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

U 1556, 2732,4551,0202,4374,0007,0700,0000,0077,7740

U 2732, 1172,4751,1204,4374,4007,0700,0000,0000,0424

U 1172, 1575,0111,1104,4174,4007,0700,0010,0000,0000

ED

U 1173, 1174,3770,0203,4344,4007,0700,0000,0000,0000

U 1174, 2740,4521,1203,4074,0007,0700,0010,0000,0000

U 1175, 0334,3333,0004,4174,4007,0700,0200,0021,1016

U 0334, 2741,3333,0003,4175,5007,0701,0210,0000,0002

U 0336, 0020,3333,0001,4175,5007,0701,0200,0000,0002

U 0020, 2741,4221,0002,4174,0007,0700,0010,0000,0000

U 0022, 3554,3333,0002,4175,5007,0701,0210,0000,0002

U 0023, 3702,0111,0704,4170,4007,0700,0210,0023,1016

```
; 3935 ;HERE FOR UUD'S WHICH TRAP TO EXEC
; 3936 1556:
; 3937 UUD: ;THIS TAG IS USED FOR ILLEGAL THINGS WHICH DO UUD TRAPS
; 3938 MUUD: ;THIS TAG IS USED FOR MONITOR CALL INSTRUCTIONS
; 3939 [HR][HR].AND.#, ;MASK OUT @ AND XR
; 3940 #/777740, ;MASK
; 3941 HOLD RIGHT ;KEEP RIGHT
; 3942 ;THE UUD MACRO DOES THE ABOVE INSTRUCTION AND GOES TO UUGO
; 3943 UUGO: [ARX]_O XWD [424] ;HERE FROM UUD MACRO
; 3944 ;GET OFFSET TO UPT
; 3945 =O [ARX][ARX]+[UBR], ;ADDRESS OF MUUD WORD
; 3946 CALL [ABORT] ;STOP MEMORY
;;3947 .IF/KIPAGE
;;3948 .IF/KLPAGE
;;3949 READ [EBR], ;IF BOTH POSSIBLE, SEE WHICH IS ENAB

;;3950 SKIP DPO ;KL PAGING ??
;;3951 =O
;;3952 .ENDIF/KLPAGE
;;3953 READ [ARX]. ;GET THE ADDRESS
;;3954 LOAD VMA, ;START WRITE
;;3955 VMA PHYSICAL WRITE, ;ABSOLUTE ADDRESS
;;3956 J/KIMUUD ;GO STORE KI STYLE
; 3957 .ENDIF/KIPAGE
; 3958 .IF/KLPAGE
; 3959 [AR][HR] SWAP ;PUT IN RIGHT HALF
; 3960 =O [AR]_FLAGS, ;FLAGS IN LEFT HALF
; 3961 HOLD RIGHT, ;JUST WANT FLAGS
; 3962 CALL [UUDFLG] ;CLEAR TRAP FLAGS
; 3963 READ [ARX], ;LOOK AT ADDRESS
; 3964 LOAD VMA, ;LOAD THE VMA
; 3965 VMA PHYSICAL WRITE ;STORE FLAG WORD
; 3966 =O* MEM WRITE, ;WAIT FOR MEMORY
; 3967 MEM_[AR], CALL [NEXT] ;STORE
; 3968 MEM WRITE, ;WAIT FOR MEMORY
; 3969 MEM_[PC] ;STORE FULL WORD PC
; 3970 =OOO [HR]_O, ;SAVE E
; 3971 HOLD RIGHT, CALL [NEXT] ;BUT CLEAR OPCODE
; 3972 .ENDIF/KLPAGE
; 3973 =O10
; 3974 UUDPCW: MEM WRITE, ;WAIT FOR MEMORY
; 3975 MEM_[HR], ;STORE INSTRUCTION IN KI
; 3976 ; OR FULL WORD E IN KL
; 3977 CALL [GETPCW] ;GET PROCESS-CONTEXT-WORD
; 3978
; 3979 =O11 NEXT [ARX] PHYSICAL WRITE, ;POINT TO NEXT WORD
; 3980 CALL [STOBR] ;STORE PROCESS CONTEXT WORD
; 3981
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 109
ILLEGAL INSTRUCTIONS AND UO0'S

```
U 0027, 2733,4751,1205,4374,4007,0700,0000,0000,0430 ; 3982 ;NOW WE MUST PICK ONE OF 8 NEW PC WORDS BASED ON PC FLAGS
; 3983 =111 [BR]_O XWD [430] ;OFFSET INTO UPT
; 3984 =
; 3985 [BR]_[BR]+[UBR] ;ADDRESS OF WORD
U 2733, 2734,0111,1105,4174,4007,0700,0000,0000,0000 ; 3986 [AR]_FLAGS ;GET FLAGS
U 2734, 2735,4521,1203,4074,4007,0700,0000,0000,0000 ; 3987 TL [AR], ;LOOK AT FLAGS
; 3988 #/600 ;TRAP SET?
U 2735, 1176,4553,0300,4374,4007,0321,0000,0000,0600 ; 3989 =0 [BR]_[BR].OR.#, ;YES--POINT TO TRAP CASE
; 3990 #/1, ;..
; 3991 HOLD LEFT ;LEAVE LEFT ALONE
U 1176, 1177,3551,0505,4370,4007,0700,0000,0000,0001 ; 3992 TL [AR], ;USER OR EXEC
; 3993 #/10000 ;..
U 1177, 1200,4553,0300,4374,4007,0321,0000,0001,0000 ; 3994 =0 [BR]_[BR].OR.#, ;USER
; 3995 #/4, ;POINT TO USER WORDS
; 3996 HOLD LEFT
U 1200, 1201,3551,0505,4370,4007,0700,0000,0000,0004 ; 3997 READ [BR], ;LOOK AT ADDRESS
; 3998 LOAD VMA, ;PLACE IN VMA
; 3999 VMA PHYSICAL, ;PHYSICAL ADDRESS
U 1201, 2736,3333,0005,4174,4007,0700,0200,0024,1016 ; 4000 START READ ;GET NEW PC WORD
; 4001 GOEXEC: MEM READ, ;WAIT FOR DATA
; 4002 [AR]_MEM ;STICK IN AR
; 4003 READ [AR], ;LOOK AT DATA
; 4004 LOAD FLAGS, ;LOAD NEW FLAGS
; 4005 LEAVE USER, ;ALLOW USER TO LOAD
; 4006 LOAD PCU, ;SET PCU FROM USER
; 4007 J/JUMPA ;JUMP
; 4008
; 4009 .IF/KIPAGE
; 4010 ;HERE FOR TOPS-10 STYLE PAGING
; 4011
; 4012 =00
; 4013 KIMU00: MEM WRITE, ;STORE INSTRUCTION
; 4014 MEM [HR], CALL [NEXT] ;IN MEMORY
; 4015 =10 [AR]_PC WITH FLAGS, ;GET PC WORD
; 4016 CALL [UUOFLG] ;CLEAR TRAP FLAGS
; 4017 =11 MEM WRITE, ;STORE PC WORD
; 4018 MEM [AR], ;..
; 4019 J/UUOPCW ;GO STORE PROCESS CONTEXT
; 4020 .ENDIF/KIPAGE
; 4021
; 4022 UUOFLG: [AR]_[AR].AND.NOT.#, ;CLEAR TRAP FLAGS
; 4023 #/600, HOLD RIGHT, ; IN WORD TO SAVE
; 4024 RETURN [1] ; BACK TO CALLER
; 4025
; 4026 NEXT: NEXT [ARX] PHYSICAL WRITE, ;POINT TO NEXT WORD
; 4027 RETURN [2]
; 4028
```

U 2740, 0001,5551,0303,4374,0004,1700,0000,0000,0600

U 2741, 0002,0111,0704,4170,4004,1700,0200,0023,1016

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

U 1557, 0400,4751,1203,4374,4007,0700,0000,0000,0040

U 0400, 2742,3333,0003,4174,4007,0700,0200,0003,0012

U 2742, 2743,4551,0202,4374,0007,0700,0000,0077,7740

U 2743, 2744,3333,0002,4175,5007,0701,0200,0000,0002

U 2744, 2521,0111,0703,4170,4007,0700,0200,0004,0012

```
; 4029 ;HERE FOR LUUU'S
; 4030 1557:
; 4031 LUUU: [AR]_O XWD [40] ;AR GET CONSTANT 40
; 4032 ;THE LUUU MACRO DOES THE ABOVE INSTRUCTION AND GOES TO LUUU1
; 4033 400: ;FOR SIMULATOR
; 4034 LUUU01: READ [AR], ;LOAD 40 INTO
; 4035 LOAD VMA, ; THE VMA AND
; 4036 START WRITE ; PREPARE TO STORE
; 4037 [HR]_[HR].AND.#, ;CLEAR OUT INDEX AND @
; 4038 #/777740, ; ...
; 4039 HOLD RIGHT
; 4040 MEM WRITE, ;STORE LUUU IN 40
; 4041 MEM [HR]
; 4042 VMA_[AR]+1, ;POINT TO 41
; 4043 LOAD VMA, ;PUT 41 IN VMA
; 4044 START READ, ;START FETCH
; 4045 J/CONT1 ;GO EXECUTE THE INSTRUCTION
; 4046
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 111
ARITHMETIC -- ADD, SUB

```

; 4047 .TOC "ARITHMETIC -- ADD, SUB"
; 4048
; 4049 .DCODE
; 4050 270: R-PF, AC, J/ADD
; 4051 I-PF, AC, J/ADD
; 4052 RW, M, J/ADD
; 4053 RW, B, J/ADD
; 4054 .UCODE
; 4055
; 4056 1560:
; 4057 ADD: [AR]_[AR]+AC, ;DO THE ADD
; 4058 AD FLAGS EXIT, 3T ;UPDATE CARRY FLAGS
; 4059 ;STORE ANSWER
; 4060 ;MISSES 3-TICKS BY 3 NS.
; 4061
; 4062
; 4063 .DCODE
; 4064 274: R-PF, AC, J/SUB
; 4065 I-PF, AC, J/SUB
; 4066 RW, M, J/SUB
; 4067 RW, B, J/SUB
; 4068 .UCODE
; 4069
; 4070 1561:
; 4071 SUB: [AR]_AC-[AR], ;DO THE SUBTRACT
; 4072 AD FLAGS EXIT, 3T ;UPDATE PC CARRY FLAGS
; 4073 ;ALL DONE
; 4074 ;MISSES 3-TICKS BY 3 NS.
; 4075

D 0270, 1015, 1560, 1100
D 0271, 0015, 1560, 3000
D 0272, 0016, 1560, 1700
D 0273, 0017, 1560, 1700

U 1560, 1500, 0551, 0303, 0274, 4463, 7701, 0200, 0001, 0001

D 0274, 1015, 1561, 1100
D 0275, 0015, 1561, 3000
D 0276, 0016, 1561, 1700
D 0277, 0017, 1561, 1700

U 1561, 1500, 2551, 0303, 0274, 4463, 7701, 4200, 0001, 0001
```


; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-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, 1202, 0551, 0404, 1274, 4007, 0562, 0000, 0000, 1441

U 1202, 2746, 0551, 0303, 0274, 4467, 0702, 4000, 0001, 0001
U 1203, 2745, 7441, 1205, 4174, 4007, 0700, 0000, 0000, 0000

U 2745, 1202, 3111, 0503, 4170, 4007, 0700, 0000, 0000, 0000

U 1615, 1204, 2551, 0404, 1274, 4007, 0562, 4000, 0000, 1441

U 1204, 2746, 2551, 0303, 0274, 4467, 0702, 0000, 0001, 0001

U 1205, 2746, 2551, 0303, 0274, 4467, 0702, 4000, 0001, 0001

U 2746, 1206, 3770, 0303, 4174, 0007, 0520, 0000, 0000, 0000
U 1206, 1404, 4551, 0404, 4374, 0007, 0700, 0000, 0037, 7777
U 1207, 1404, 3551, 0404, 4374, 0007, 0700, 0000, 0040, 0000

```
; 4076 .TOC "ARITHMETIC -- DADD, DSUB"  
; 4077  
; 4078 .DCODE  
; 4079 114: DBL R, DAC, J/DADD  
; 4080 DBL R, DAC, J/DSUB  
; 4081 .UCODE  
; 4082  
; 4083 1457:  
; 4084 DADD: [ARX]_[ARX]+AC[1], 4T, ;ADD LOW WORDS  
; 4085 SKIP CRY1 ;SEE IF CARRY TO HIGH WORD  
; 4086 =0  
; 4087 DADD1: [AR]_[AR]+AC, ;ADD HIGH WORDS  
; 4088 ADD .25, ;ADD IN ANY CARRY FROM LOW WORD  
; 4089 AD FLAGS, 4T, ;UPDATE PC FLAGS  
; 4090 J/CPYSGN ;COPY SIGN TO LOW WORD  
; 4091 [BR]_.NOT.[MASK] ;SET BITS 35 AND 36 IN  
; 4092 [AR]_[AR].OR.[BR], ; AR SO THAT ADD .25 WILL  
; 4093 HOLD LEFT, J/DADD1 ; ADD 1.  
; 4094  
; 4095 1615:  
; 4096 DSUB: [ARX]_AC[1]-[ARX], 4T, ;SUBTRACT LOW WORD  
; 4097 SKIP CRY1 ;SEE IF CARRY  
; 4098 =0 [AR]_AC-[AR]-.25, ;NO CARRY  
; 4099 AD FLAGS, 4T, ;UPDATE PC FLAGS  
; 4100 J/CPYSGN ;GO COPY SIGN  
; 4101 [AR]_AC-[AR], 4T, ;THERE WAS A CARRY  
; 4102 AD FLAGS ;UPDATE CARRY FLAGS  
; 4103  
; 4104 CPYSGN: FIX [AR] SIGN, SKIP DPO  
; 4105 =0 [ARX]_[ARX].AND.#, #/377777, HOLD RIGHT, J/MOVE  
; 4106 [ARX]_[ARX].OR.#, #/400000, HOLD RIGHT, J/MOVE  
; 4107  
; 4108
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 113
ARITHMETIC -- MUL, IMUL

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

U 1641, 2747, 3441, 0306, 0174, 4007, 0700, 0000, 0000, 0000
U 2747, 0021, 3772, 0000, 0275, 5007, 0700, 2000, 0071, 0043

U 0021, 2773, 3446, 0606, 4174, 4007, 0700, 0010, 0000, 0000
U 0025, 1210, 3333, 0004, 4174, 4007, 0621, 0000, 0000, 0000
U 1210, 2750, 3445, 0404, 4174, 4007, 0700, 0000, 0000, 0000
U 1211, 1500, 3221, 0003, 4174, 4003, 7700, 0200, 0003, 0001

U 2750, 1212, 5113, 0412, 4174, 4007, 0621, 0000, 0000, 0000

U 1212, 1214, 3770, 0404, 4174, 0007, 0520, 0000, 0000, 0000
U 1213, 1500, 7001, 0003, 4174, 4003, 7700, 0200, 0003, 0001

U 1214, 1404, 3221, 0003, 4174, 4467, 0700, 0000, 0041, 1000
U 1215, 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, 2751, 3442, 0300, 0174, 4007, 0700, 0000, 0000, 0000
U 2751, 2752, 3441, 0316, 4174, 4007, 0700, 0000, 0000, 0000
U 2752, 0031, 3771, 0006, 0276, 6007, 0700, 2000, 0071, 0043

U 0031, 2773, 3446, 0606, 4174, 4007, 0700, 0010, 0000, 0000
U 0035, 2753, 3445, 0403, 4174, 4007, 0700, 0000, 0000, 0000
U 2753, 1216, 3770, 0303, 4174, 0007, 0520, 0000, 0000, 0000

U 1216, 1500, 4001, 0004, 4174, 4003, 7700, 0200, 0003, 0001
U 1217, 1220, 4113, 0616, 4174, 4007, 0520, 0000, 0000, 0000
U 1220, 1500, 7001, 0004, 4174, 4003, 7700, 0200, 0003, 0001

U 1221, 1404, 7001, 0004, 4174, 4467, 0700, 0000, 0041, 1000

```
; 4109 .TOC "ARITHMETIC -- MUL, IMUL"
; 4110
; 4111 .DCODE
; 4112 220: R-PF, AC, J/IMUL
; 4113 I-PF, AC, J/IMUL
; 4114 RW, M, J/IMUL
; 4115 RW, B, J/IMUL
; 4116 .UCODE
; 4117 1641:
; 4118 IMUL: [BRX]_[AR], AC ;COPY C(E)
; 4119 Q AC, SC 35. ;GET THE AC
; 4120 =O** [BRX]_[BRX]*.5 LONG, ;SHIFT RIGHT
; 4121 CALL [MULSUB] ;MULTIPLY
; 4122 READ [ARX], SKIP AD.EQ.O ;SEE IF FITS
; 4123 =O [ARX]_[ARX]*2, J/IMUL2 ;NOT ZERO--SHIFT LEFT
; 4124 IMUL1: [AR]_Q, EXIT ;POSITIVE
; 4125
; 4126 IMUL2: [MASK].AND.NOT.[ARX], ;SEE IF ALL SIGN BITS
; 4127 SKIP AD.EQ.O ;...
; 4128 =O FIX [ARX] SIGN, ;NOT ALL SIGN BITS
; 4129 SKIP DPO, J/IMUL3 ;GIVE + OR - OVERFLOW
; 4130 [AR]_[MAG].EQV.Q, EXIT ;NEGATIVE
; 4131 =O
; 4132 IMUL3: [AR]_Q, SET AROV, J/MOVE
; 4133 [AR]_[MAG].EQV.Q, SET AROV, J/MOVE
; 4134
; 4135
; 4136 .DCODE
; 4137 224: R-PF, DAC, J/MUL
; 4138 I-PF, DAC, J/MUL
; 4139 RW, M, J/MUL
; 4140 RW, DBL B, J/MUL
; 4141 .UCODE
; 4142
; 4143
; 4144 1571:
; 4145 MUL: Q [AR], AC ;COPY C(E)
; 4146 [TO]_[AR] ;SAVE FOR OVERFLOW TEST
; 4147 [BRX] AC, SC 35. ;GET THE AC
; 4148 =O** [BRX]_[BRX]*.5 LONG, ;SHIFT OVER
; 4149 CALL [MULSUB] ;MULTIPLY
; 4150 [AR]_[ARX]*2 ;SHIFT OVER
; 4151 FIX [AR] SIGN, SKIP DPO ;SEE IF NEGATIVE
; 4152 =O [ARX]_[MAG].AND.Q, ;POSITIVE
; 4153 EXIT
; 4154 [TO].AND.[BRX], SKIP DPO ;TRIED TO SQUARE 1B0?
; 4155 =O [ARX]_[MAG].EQV.Q, EXIT ;NO
; 4156 [ARX]_[MAG].EQV.Q, ;YES
; 4157 SET AROV, J/MOVE
; 4158
; 4159
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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, 2754, 3447, 0303, 4174, 4007, 0700, 0000, 0000, 0000
U 2754, 2755, 4117, 0004, 4174, 4007, 0700, 0000, 0000, 0000

U 2755, 0120, 3441, 0405, 4174, 4007, 0350, 0000, 0000, 0000

U 0120, 2767, 4557, 0006, 1274, 4007, 0701, 0010, 0000, 1441

U 0121, 2757, 4557, 0004, 1274, 4007, 0701, 0000, 0000, 1442
U 0124, 0171, 3223, 0000, 1174, 4007, 0700, 0400, 0000, 1443

U 0171, 0563, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043
U 0175, 2756, 3441, 0416, 4174, 4007, 0700, 0000, 0000, 0000

U 2756, 0310, 3227, 0004, 1174, 4007, 0700, 0400, 0000, 1442
U 2757, 0310, 3777, 0016, 1276, 6007, 0701, 0000, 0000, 1441

U 0310, 2770, 3777, 0006, 0274, 4007, 0701, 0010, 0000, 0000

U 0311, 2772, 0113, 1616, 1174, 4007, 0701, 0400, 0000, 1441
U 0314, 2760, 3223, 0000, 1174, 4007, 0700, 0400, 0000, 1442

U 2760, 0543, 0111, 1604, 4174, 4007, 0700, 0000, 0000, 0000

U 0543, 0563, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043

U 0547, 2761, 3445, 0404, 4174, 4467, 0700, 0000, 0005, 0000

```
; 4160 .TOC "ARITHMETIC -- DMUL"
; 4161
; 4162 .DCODE
; 4163 116: DBL R, DAC, J/DMUL
; 4164 .UCODE
; 4165
; 4166 .IF/FULL
; 4167 1566:
; 4168 DMUL: [AR] [AR]*.5 ;SHIFT MEM OPERAND RIGHT
; 4169 [ARX] ([ARX].AND.[MAG])* .5
; 4170 [BR] [ARX], ;COPY LOW WORD
; 4171 SKIP FPD ;SEE IF FIRST PART DONE
; 4172 ;
; 4173 ; BRX * BR ==> C(E+1) * C(AC+1)
; 4174 ;
; 4175 =000 [BRX]_AC[1].AND.[MAG])* .5, 3T, ;GET LOW AC
; 4176 CALL [DMULGO] ;START MULTIPLY
; 4177 [ARX]_AC[2].AND.[MAG])* .5, 3T, ;FIRST PART DONE
; 4178 J/DMUL1 ;GO DO SECOND PART
; 4179 =100 AC[3]_Q ;SALT AWAY 1 WORD OF PRODUCT
; 4180 =
; 4181 ;
; 4182 ; BRX * Q ==> C(E) * C(AC+1)
; 4183 ;
; 4184 =0** Q_[AR], SC 35., ;GO MULT NEXT HUNK
; 4185 CALL [QMULT] ;
; 4186 [TO] [ARX] ;SAVE PRODUCT
; 4187 AC[2]_Q, [ARX]_Q*.5, ;SAVE PRODUCT
; 4188 J/DMUL2 ;GO DO HIGH HALF
; 4189 DMUL1: [TO]_AC[1]*.5 ;RESTORE TO
; 4190 =0*0
; 4191 ;
; 4192 ; BRX * BR ==> C(AC) * C(E+1)
; 4193 ;
; 4194 DMUL2: [BRX]_AC*.5, ;PREPARE TO DO HIGH HALF
; 4195 CALL [DBLMUL] ;GO DO IT
; 4196 AC[1]_[TO]*2, 3T, ;INTERRUPT, SAVE TO
; 4197 J/DMULINT ;SET FPD AND INTERRUPT
; 4198 AC[2]_Q ;SAVE PRODUCT
; 4199 =
; 4200 [ARX]_[ARX]+[TO] ;PREPARE FOR LAST MUL
; 4201 ;
; 4202 ; BRX * Q ==> C(AC) * C(E)
; 4203 ;
; 4204 =0** Q_[AR], SC 35., ;DO THE LAST MULTIPLY
; 4205 CALL [QMULT] ;GO DO IT
; 4206 [ARX]_[ARX]*2, ;SHIFT BACK
; 4207 CLR FPD ;CLEAR FPD
; 4208
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

U 2761, 1222,3770,0404,0174,4007,0520,0400,0000,0000
U 1222, 2766,3223,0000,1174,4007,0700,0400,0000,1441
U 1223, 2762,7003,0000,1174,4007,0700,0400,0000,1441
U 2762, 2763,3772,0000,1275,5007,0701,0000,0000,1442
U 2763, 2764,7003,0000,1174,4007,0700,0400,0000,1442
U 2764, 2765,3772,0000,1275,5007,0701,0000,0000,1443
U 2765, 2766,7003,0000,1174,4007,0700,0400,0000,1443

U 2766, 1224,3741,0103,4074,4007,0520,0000,0000,0000
U 1224, 0110,3443,0100,4174,4156,4700,0200,0014,0012
U 1225, 1400,4443,0000,4174,4467,0700,0000,0041,1000

U 2767, 2770,4221,0004,4174,4007,0700,0000,0000,0000
U 2770, 2771,3442,0500,4174,4007,0700,2000,0071,0043
U 2771, 0563,3447,0606,4174,4007,0700,0000,0000,0000

U 0563, 2775,3446,1200,4174,4007,0700,0010,0000,0000

U 0567, 0004,0113,0404,4174,4464,1701,0000,0001,0001

U 2772, 2675,4443,0000,4174,4467,0700,0000,0003,0000

; 4209 AC [ARX] TEST, SKIP DPO ;PUT BACK INTO AC
; 4210 =0 AC[1]_Q, J/DMTRAP ;POSITIVE
; 4211 AC[1]_[MAG].EQV.Q ;NEGATIVE
; 4212 Q_AC[2]
; 4213 AC[2]_[MAG].EQV.Q
; 4214 Q_AC[3]
; 4215 AC[3]_[MAG].EQV.Q
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
; 4222 ;WAYS TO CALL MULTIPLY
; 4223 DMULGD: [ARX]_O ;CLEAR ARX
; 4224 DBLMUL: Q_[BR], SC 35.
; 4225 [BRX]_[BRX]*.5
; 4226 =0**
; 4227 QMULT: Q_Q*.5,
; 4228 CALL [MULTIPLY]
; 4229 [ARX]+[ARX], AD FLAGS, ;TEST FOR OVERFLOW
; 4230 3T, RETURN [4] ;AND RETURN
; 4231
; 4232 DMLINT: SET FPD, J/FIXPC ;SET FPD, BACKUP PC
; 4233 ; INTERRUPT
; 4234 .IFNOT/FULL
; 4235 1566:
; 4236 DMUL: UUD
; 4237 .ENDIF/FULL
; 4238

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 116
ARITHMETIC -- DMUL

U 2773, 2774, 3446, 0606, 4174, 4007, 0700, 0000, 0000, 0000

U 2774, 0122, 4226, 0004, 4174, 4007, 0630, 2000, 0060, 0000

U 2775, 0122, 3446, 1200, 4174, 4007, 0630, 2000, 0060, 0000

```
; 4239 ;MULTIPLY SUBROUTINE
; 4240 ;ENTERED WITH:
; 4241 ;     MULTIPLIER IN Q
; 4242 ;     MULTIPLICAND IN BRX
; 4243 ;RETURNS 4 WITH PRODUCT IN ARX!Q
; 4244
; 4245 MUL STEP      "A/BRX,B/ARX,DEST/Q_Q*.5,ASHC,STEP SC,MUL DISP"
; 4246 MUL FINAL    "A/BRX,B/ARX,DEST/Q_Q*2"
; 4247
; 4248 MULSUB: [BRX]_[BRX]*.5 LONG
; 4249 MULSB1: [ARX]_O*.5 LONG,      ;CLEAR ARX AND SHIFT Q
; 4250         STEP SC,              ;COUNT FIRST STEP
; 4251         J/MUL+                ;ENTER LOOP
; 4252
; 4253 ;MULTIPLY SUBROUTINE
; 4254 ;ENTERED WITH:
; 4255 ;     MULTIPLIER IN Q
; 4256 ;     MULTIPLICAND IN BRX
; 4257 ;     PARTIAL PRODUCT IN ARX
; 4258 ;RETURNS 4 WITH Q*BRX+ARX IN ARX!Q
; 4259
; 4260 MULTIPLY:
; 4261         Q_Q*.5,                  ;SHIFT Q
; 4262         STEP SC,                  ;COUNT FIRST STEP
; 4263         J/MUL+                    ;ENTER LOOP
; 4264
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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
; 4265 ;HERE FOR POSITIVE STEPS
; 4266 =010 ;0 IN A POSITIVE STEP
; 4267 MUL+: AD/B, ;DON'T ADD
; 4268 MUL STEP, ;SHIFT
; 4269 J/MUL+ ;KEEP POSITIVE
; 4270 =011 ;DONE
; 4271 AD/B, ;DON'T ADD
; 4272 MUL FINAL, ;SHIFT
; 4273 RETURN [4] ;SHIFT Q AND RETURN
; 4274 =110 ;1 IN A POSITIVE STEP
; 4275 AD/B-A-.25, ADD .25, ;SUBTRACT
; 4276 MUL STEP, ;SHIFT AND COUNT
; 4277 J/MUL- ;NEGATIVE NOW
; 4278 =111 ;DONE
; 4279 AD/B-A-.25, ADD .25, ;SUBTRACT
; 4280 MUL FINAL, ;SHIFT
; 4281 RETURN [4] ; AND RETURN
; 4282
; 4283 ;HERE FOR NEGATIVE STEPS
; 4284 =010 ;0 IN NEGATIVE STEP
; 4285 MUL-: AD/A+B, ;ADD
; 4286 MUL STEP, ;SHIFT AND COUNT
; 4287 J/MUL+ ;POSITIVE NOW
; 4288 =011 ;DONE
; 4289 AD/A+B, ;ADD
; 4290 MUL FINAL, ;SHIFT
; 4291 RETURN [4] ;FIX Q AND RETURN
; 4292 =110 ;1 IN NEGATIVE STEP
; 4293 AD/B, ;DON'T ADD
; 4294 MUL STEP, ;SHIFT AND COUNT
; 4295 J/MUL- ;STILL NEGATIVE
; 4296 =111 ;DONE
; 4297 AD/B, ;DON'T ADD
; 4298 MUL FINAL, ;SHIFT
; 4299 RETURN [4] ;FIX Q AND RETURN
; 4300
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

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, 2776, 3441, 0305, 0174, 4007, 0700, 0000, 0000, 0000

U 2776, 1226, 3772, 0000, 0275, 5007, 0520, 0000, 0000, 0000

U 1226, 0161, 4221, 0003, 4174, 4007, 0700, 0000, 0000, 0000

U 1227, 0161, 2441, 0703, 4174, 4007, 0700, 4000, 0000, 0000

U 1601, 2777, 3441, 0305, 4174, 4007, 0700, 0000, 0000, 0000

U 2777, 3000, 3771, 0003, 0276, 6007, 0700, 0000, 0000, 0000

U 3000, 3001, 3772, 0000, 1275, 5007, 0701, 0000, 0000, 1441

U 3001, 0160, 3333, 0003, 4174, 4007, 0621, 0000, 0000, 0000

U 0160, 1230, 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, 1211, 3441, 0304, 4174, 4007, 0700, 0000, 0000, 0000

U 1230, 3002, 3441, 0306, 4174, 4007, 0700, 0000, 0000, 0000

U 1231, 0164, 3333, 0005, 4174, 4007, 0621, 0000, 0000, 0000

```
; 4301 .TOC "ARITHMETIC -- DIV, IDIV"  
; 4302  
; 4303 .DCODE  
; 4304 230: R-PF, DAC, J/IDIV  
; 4305 I-PF, DAC, J/IDIV  
; 4306 RW, M, J/IDIV  
; 4307 RW, DBL B, J/IDIV  
; 4308  
; 4309 234: R-PF, DAC, J/DIV  
; 4310 I-PF, DAC, J/DIV  
; 4311 RW, M, J/DIV  
; 4312 RW, DBL B, J/DIV  
; 4313 .UCODE  
; 4314  
; 4315 1600:  
; 4316 IDIV: [BR]_[AR], AC ;COPY MEMORY OPERAND  
; 4317 Q_AC, ;LOAD Q  
; 4318 SKIP DPO ;SEE IF MINUS  
; 4319 =0 [AR]_O, ;EXTEND + SIGN  
; 4320 J/DIV1 ;NOW SAME AS DIV  
; 4321 [AR]_-1, ;EXTEND - SIGN  
; 4322 J/DIV1 ;SAME AS DIV  
; 4323  
; 4324 1601:  
; 4325 DIV: [BR]_[AR] ;COPY MEM OPERAND  
; 4326 [AR]_AC ;GET AC  
; 4327 Q_AC[1] ;AND AC+1  
; 4328 READ [AR], ;TEST FOR NO DIVIDE  
; 4329 SKIP AD.EQ.O  
; 4330 =000 .NOT.[AR], ;SEE IF ALL SIGN BITS IN AR  
; 4331 SKIP AD.EQ.O, ;...  
; 4332 J/DIVA ;CONTINUE BELOW  
; 4333 =001  
; 4334 DIV1: READ [BR], ;SEE IF DIVIDE BY  
; 4335 SKIP AD.EQ.O ;ZERO  
; 4336 =100  
; 4337 DIV2: SC_34, ;NOT ZERO--LOAD STEP COUNT  
; 4338 CALL [DIVSUB] ;DIVIDE  
; 4339 =101 NO DIVIDE ;DIVIDE BY ZERO  
; 4340 =110 [ARX]_[AR], ;COPY REMAINDER  
; 4341 J/IMUL1 ;STORE ANSWER  
; 4342 =  
; 4343  
; 4344  
; 4345 =0  
; 4346 DIVA: [BRX]_[AR], ;HIGH WORD IS NOT SIGNS  
; 4347 J/DIVB ;GO TEST FOR NO DIVIDE  
; 4348 READ [BR], ;ALL SIGN BITS  
; 4349 SKIP AD.EQ.O, ;SEE IF ZERO DIVIDE  
; 4350 J/DIV2 ;BACK TO MAIN FLOW  
; 4351
```

```

U 3002, 3003,3221,0004,4174,4007,0700,0000,0000,0000 ; 4352 DIVB: [ARX]_0 ;MAKE ABS VALUES
; 4353 READ [AR], ;SEE IF +
U 3003, 0330,3333,0003,4174,4007,0520,0000,0000,0000 ; 4354 SKIP DPO
; 4355 =00 READ [BR], ;SEE IF +
; 4356 SKIP DPO,
; 4357 J/DIVC ;CONTINUE BELOW
U 0330, 1232,3333,0005,4174,4007,0520,0000,0000,0000 ; 4358 CLEAR [ARX]0, ;FLUSH DUPLICATE SIGN
; 4359 CALL [DBLNG1] ;NEGATE ARI!ARX
U 0331, 3064,4551,0404,4374,0007,0700,0010,0037,7777 ; 4360 =11 READ [BR], ;SEE IF TOO BIG
; 4361 SKIP DPO,
; 4362 J/DIVC
U 0333, 1232,3333,0005,4174,4007,0520,0000,0000,0000 ; 4363 =
; 4364 =0
; 4365 DIVC: [AR]-[BR], ;COMPUTE DIFFERENCE
; 4366 SKIP DPO, ;SEE IF IT GOES
; 4367 3T, ;ALLOW TIME
U 1232, 1234,2113,0305,4174,4007,0521,4000,0000,0000 ; 4368 J/NODIV ;TEST
; 4369 [AR]+[BR],
; 4370 SKIP DPO, ;SAME TEST FOR -VE BR
; 4371 3T,
; 4372 J/NODIV
U 1233, 1234,0113,0305,4174,4007,0521,0000,0000,0000 ; 4373 =0
; 4374 NODIV: NO DIVIDE ;TOO BIG
U 1234, 0603,4443,0000,4174,4467,0700,0000,0051,1000 ; 4375 [AR]_[BRX], ;FITS
; 4376 J/DIV1 ;GO BACK AND DIVIDE
U 1235, 0161,3441,0603,4174,4007,0700,0000,0000,0000 ; 4377

```

Produced on Advanced Information Services Electronic Laser Printer. PKO/JES. DTN. 223-7881

D 0117, 0205, 1627, 1100

U 1627, 3004, 4112, 0400, 4174, 4007, 0700, 0000, 0000, 0000

U 3004, 1236, 3447, 0305, 4174, 4007, 0421, 0000, 0000, 0000

U 1236, 1242, 3446, 0505, 4174, 4007, 0700, 0000, 0000, 0000

U 1237, 1240, 3446, 0505, 4174, 4007, 0520, 0000, 0000, 0000

U 1240, 1242, 4003, 0000, 4174, 4007, 0621, 0000, 0000, 0000

U 1241, 3005, 4751, 1217, 4374, 4007, 0700, 0000, 0000, 0005

U 3005, 3006, 3662, 0000, 4374, 0007, 0700, 0000, 0060, 0000

U 3006, 3007, 2222, 0000, 4174, 4007, 0700, 4000, 0000, 0000

U 3007, 3011, 2446, 0505, 4174, 4047, 0700, 0040, 0000, 0000

U 1242, 3010, 3446, 0505, 4174, 4047, 0700, 0000, 0000, 0000

U 1243, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0051, 1000

U 3010, 3011, 4751, 1217, 4374, 4007, 0700, 0000, 0000, 0004

U 3011, 3012, 3221, 0006, 0174, 4007, 0700, 0000, 0000, 0000

U 3012, 0054, 3777, 0003, 0274, 4007, 0520, 0000, 0000, 0000

U 0054, 1244, 4552, 0000, 1275, 5007, 0701, 0000, 0000, 1441

U 0055, 3053, 6551, 1717, 4374, 4007, 0700, 0010, 0000, 0007

U 0075, 0054, 3447, 0303, 4174, 4007, 0700, 0000, 0000, 0000

U 1244, 3035, 3446, 0303, 4174, 4007, 0700, 0010, 0000, 0000

U 1245, 1246, 2113, 0305, 4174, 4007, 0521, 4000, 0000, 0000

U 1246, 1250, 2113, 0305, 4174, 4007, 0620, 4000, 0000, 0000

U 1247, 3013, 3221, 0004, 4174, 4007, 0700, 0000, 0000, 0000

U 1250, 0033, 3333, 0017, 4174, 4003, 5701, 0000, 0000, 0000

U 1251, 1252, 1003, 0600, 4174, 4007, 0521, 4000, 0000, 0000

U 1252, 0033, 3333, 0017, 4174, 4003, 5701, 0000, 0000, 0000

U 1253, 3013, 3221, 0004, 4174, 4007, 0700, 0000, 0000, 0000

```
; 4378 .TOC "ARITHMETIC -- DDIV"
; 4379
; 4380 .DCODE
; 4381 117: DBL R, DAC, J/DDIV
; 4382 .UCODE
; 4383
; 4384 .IF/FULL
; 4385 1627:
; 4386 DDIV: Q [ARX].AND.[MAG] ;COPY LOW WORD
; 4387 [BR]_[AR]*.5, ;COPY MEMORY OPERAND
; 4388 SKIP AD.LE.O ;SEE IF POSITIVE
; 4389 =0 [BR][BR]*.5 LONG, ;POSITIVE
; 4390 J/DDIV1 ;CONTINUE BELOW
; 4391 [BR]_[BR]*.5 LONG, ;NEGATIVE OR ZERO
; 4392 SKIP DPO ;SEE WHICH?
; 4393 =0 [MAG].AND.Q, ;SEE IF ALL ZERO
; 4394 SKIP AD.EQ.O, J/DDIV1 ;CONTINUE BELOW
; 4395 [T1]_O XWD [5] ;NEGATE MEM OP
; 4396 Q_Q.OR.#, #/600000, ;SIGN EXTEND THE LOW
; 4397 HOLD RIGHT ;WORD
; 4398 Q-Q ;MAKE Q POSITIVE
; 4399 [BR]_(-[BR]-.25)*.5 LONG, ;NEGATE HIGH WORD
; 4400 ASHC, MULTI PREC/1, ;USE CARRY FROM LOW WORD
; 4401 J/DDIV3 ;CONTINUE BELOW
; 4402 =0
; 4403 DDIV1: [BR]_[BR]*.5 LONG, ;SHIFT OVER 1 PLACE
; 4404 ASHC, J/DDIV2 ;CONTINUE BELOW
; 4405 NO DIVIDE ;DIVIDE BY ZERO
; 4406 DDIV2: [T1]_O XWD [4] ;MEM OPERAND IS POSITIVE
; 4407 DDIV3: [BRX]_Q, AC ;COPY Q
; 4408
; 4409 [AR]_AC*.5, 2T, SKIP DPO ;GET AC--SEE IF NEGATIVE
; 4410 =0*1*0
; 4411 DDIV3A: Q_AC[1].AND.[MAG], ;POSITIVE (OR ZERO)
; 4412 J/DDIV4 ;CONTINUE BELOW
; 4413 [T1]_[T1].XOR.#, ;NEGATIVE
; 4414 #/7, CALL [QDNEG] ;UPDATE SAVED FLAGS
; 4415 =1*1*1 [AR]_[AR]*.5, ;SHIFT AR OVER
; 4416 J/DDIV3A ;GO BACK AND LOAD Q
; 4417 =
; 4418 =0
; 4419 DDIV4: [AR]_[AR]*.5 LONG, ;SHIFT AR OVER
; 4420 CALL [DDIVS] ;SHIFT 1 MORE PLACE
; 4421 [AR]_[BR], 3T, SKIP DPO ;TEST MAGNITUDE
; 4422 =0 [AR]-[BR], 2T,
; 4423 SKIP AD.EQ.O, J/DDIV5
; 4424 [ARX]_Q, J/DDIV5A ;ANSWER FITS
; 4425 =0
; 4426 =0
; 4427 DDIV5: READ [T1], 3T, DISP/DP, J/NO DDIV
; 4428 Q-[BRX], 3T, SKIP DPO
; 4429 =0 READ [T1], 3T, DISP/DP, J/NO DDIV
; 4430 [ARX]_Q ;COPY LOW WORD
; 4431
```

Produced on Advanced Information Services Electronic Laser Printer, PKC/IES6, DTN: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
U 3013, 0354,4552,0000,1275,5007,0701,0000,0000,1442 ; 4432 ;HERE WITH EVERYTHING SETUP AND READY TO GO
U 0354, 1276,3446,1200,4174,4007,0700,2010,0071,0042 ; 4433 DDIV5A: Q_AC[2].AND.[MAG]
U 0356, 3014,3224,0016,4174,4007,0700,0000,0000,0000 ; 4434 =0* Q_Q*.5, SC_34., CALL [DBLDIV]
U 3014, 3015,0002,1600,4174,4007,0700,0000,0000,0000 ; 4435 [TO]_Q*2 LONG
U 3015, 1254,4003,0000,1174,4007,0700,0400,0000,1440 ; 4436 Q_Q+[TO]
U 1254, 3035,3442,0400,4174,4007,0700,0010,0000,0000 ; 4437 AC[0]_Q.AND.[MAG] ;STORE ANSWER
U 1255, 3016,3221,0004,4174,4007,0700,0000,0000,0000 ; 4438 =0 Q_[ARX], CALL [DDIVS] ;SHIFT OUT EXTRA ZERO BIT
U 3016, 0411,4552,0000,1275,5007,0701,0000,0000,1443 ; 4439 [ARX]_Q ; ..
; 4440 Q_AC[3].AND.[MAG]
; 4441 =0* [TO]_[AR]*.5 LONG, ;SHIFT Q, PUT AR ON DP
; 4442 SC_34., ;LOAD SHIFT COUNT
; 4443 SKIP DPO, ;LOOK AT AR SIGN
; 4444 CALL [DBLDIV] ;GO DIVIDE
U 0411, 1276,3446,0316,4174,4007,0520,2010,0071,0042 ; 4445 [TO]_Q*2 LONG
U 0413, 3017,3224,0016,4174,4007,0700,0000,0000,0000 ; 4446 READ [T1], 3T, DISP/DP ;WHAT SIGN IS QUO
U 3017, 0056,3333,0017,4174,4003,5701,0000,0000,0000 ; 4447 =1110 [TO]_[TO]+Q, ;POSITIVE QUO
; 4448 J/DDIV5B ;CONTINUE BELOW
; 4449 [TO]_-Q*2 ;NEGATIVE QUO
; 4450 AD/-D-.25, DBUS/RAM, 3T,
; 4451 RAMADR/AC#, DEST/Q_AD,
; 4452 MULTI PREC/1
U 3020, 3021,1772,0000,0274,4007,0701,0040,0000,0000 ; 4453 AC_Q, SKIP AD.EQ.O
U 3021, 1256,3223,0000,0174,4007,0621,0400,0000,0000 ; 4454 =0 AC[1]_[TO], J/DDIV5C
U 1256, 3023,3440,1616,1174,4007,0700,0400,0000,1441 ; 4455 AC[1]_O, J/DDIV6
U 1257, 3025,4223,0000,1174,4007,0700,0400,0000,1441 ; 4456
; 4457 DDIV5B: AC[1]_[TO].AND.[MAG], J/DDIV6 ;STORE LOW WORD IN + CASE
; 4458
; 4459 DDIV5C: [TO]_[TO].OR.#, #/400000, HOLD RIGHT
U 3022, 3025,4113,1600,1174,4007,0700,0400,0000,1441 ; 4460 AC[1]_[TO]
; 4461
; 4462 DDIV6: READ [AR], SKIP DPO ;LOOK AT AR SIGN
U 3023, 3024,3551,1616,4374,0007,0700,0000,0040,0000 ; 4463 =0
U 3024, 3025,3440,1616,1174,4007,0700,0400,0000,1441 ; 4464 DDIV7: Q_[ARX], J/DDIV8
; 4465 Q_[ARX]+[BRX]
; 4466 [AR]_[AR]+[BR],
; 4467 MULTI PREC/1
; 4468 Q_Q+[BRX]
; 4469 [AR]_- [AR]+[BR],
; 4470 MULTI PREC/1
U 3025, 1260,3333,0003,4174,4007,0520,0000,0000,0000 ; 4471 DDIV8: READ [T1], 3T, DISP/DP
; 4472 =1101
; 4473 DDIV8A: [AR]_[AR]*2 LONG, ASHC, ;POSITIVE REMAINDER
; 4474 J/DDIV9 ;CONTINUE BELOW
; 4475 Q_-Q ;NEGATE REMAINDER IN AR!Q
; 4476 [AR]_(-[AR]-.25)*2 LONG,
; 4477 MULTI PREC/1, ASHC
; 4478
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

U 3033, 1262,0113,0303,1174,4007,0521,0400,0000,1442
U 1262, 0100,4003,0000,1174,4156,4700,0400,0000,1443
U 1263, 3034,4002,0000,1174,4007,0700,0000,0000,1443
U 3034, 0100,7003,0000,1174,4156,4700,0400,0000,1443

U 0033, 3053,4443,0000,4174,4007,0700,0010,0000,0000
U 0037, 0603,4443,0000,4174,4467,0700,0000,0051,1000

U 3035, 0001,3446,0303,4174,4044,1700,0000,0000,0000

```
; 4479 DDIV9: AC[2]_[AR]+[AR], 3T,  
; 4480 SKIP DPO  
; 4481 =0 AC[3]_Q.AND.[MAG],  
; 4482 NEXT INST  
; 4483 Q_Q.AND.[MAG], AC[3]  
; 4484 AC[3]_[MAG].EQV.Q,  
; 4485 NEXT INST  
; 4486  
; 4487  
; 4488 ;HERE IF WE WANT TO SET NO DIVIDE  
; 4489 =11011  
; 4490 NOODIV: CALL [QDNEG] ;FIXUP AC TO AC+3  
; 4491 NO DIVIDE ;ABORT DIVIDE  
; 4492  
; 4493 DDIVS: [AR]_[AR]*.5 LONG, ASHC, RETURN [1]  
; 4494 .IFNOT/FULL  
; 4495 1627:  
; 4496 DDIV: UUD  
; 4497 .ENDIF/FULL  
; 4498
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/ISS, DTN: 223-7881

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

U 0370, 3036,4662,0000,4374,0007,0700,0010,0037,7777
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

```
; 4499 .TOC "ARITHMETIC -- DIVIDE SUBROUTINE"  
; 4500  
; 4501 ;HERE IS THE SUBROUTINE TO DO DIVIDE  
; 4502 ;ENTER WITH:  
; 4503 ; AR!Q = D'END  
; 4504 ; BR = D'SOR  
; 4505 ;RETURN 2 WITH:  
; 4506 ; AR = REMAINDER  
; 4507 ; Q = QUOTIENT  
; 4508 ;CALLER MUST CHECK FOR ZERO DIVIDE PRIOR TO CALL  
; 4509 ;  
; 4510 =1000  
; 4511 DIVSUB: Q_Q.AND.#, ;CLEAR SIGN BIT IN  
; 4512 #/377777, ;MASK  
; 4513 HOLD RIGHT, ;JUST CLEAR BIT 0  
; 4514 CALL [DIVSGN] ;DO REAL DIVIDE  
; 4515 =1100 RETURN [2] ;ALL POSITIVE  
; 4516 =1101 Q_-Q, RETURN [2] ;-QUO +REM  
; 4517 =1110 Q_-Q ;ALL NEGATIVE  
; 4518 =1111 [AR]_-[AR], RETURN [2] ;NEGATIVE REMAINDER  
; 4519
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 4520 ;HERE IS THE INNER DIVIDE SUBROUTINE
; 4521 ;SAME SETUP AS DIVSUB
; 4522 ;RETURNS WITH AR AND Q POSITIVE AND
; 4523 ; 14 IF ALL POSITIVE
; 4524 ; 15 IF -QUO
; 4525 ; 16 IF ALL NEGATIVE
; 4526 ; 17 IF NEGATIVE REMAINDER
; 4527
; 4528 BASIC DIV STEP "DEST/Q_Q*2, DIV, A/BR, B/AR, STEP SC"
; 4529 DIV STEP "BASIC DIV STEP, AD/A+B, DIVIDE/1"
; 4530 FIRST DIV STEP "BASIC DIV STEP, AD/B-A-.25, ADD .25"
; 4531
; 4532 DIVSGN: READ [AR], SKIP DPO
; 4533 =0 [AR]_0, J/DVSUB2 ;REMAINDER IS POSITIVE
; 4534 Q_-Q, SKIP AD.EQ.0 ;COMPLEMENT LOW WORD
; 4535 =0 [AR]_.NOT.[AR], J/DVSUB1 ;COMPLEMENT HI WORD
; 4536 [AR]_-[AR] ;TWO'S COMPLEMENT HI WORD SINCE
; 4537 ; LOW WORD WAS ZERO
; 4538 DVSUB1: [AR]_#, #/100000 ;REMAINDER IS NEGATIVE
; 4539 DVSUB2: READ [BR], SKIP DPO ;IS THE DIVISOR NEGATIVE
; 4540 =0
; 4541 DVSUB3: [AR]_[AR]*.5 LONG, ;START TO PUT IN 9-CHIPS
; 4542 J/DIVSET ;JOIN MAIN STREAM
; 4543 [BR]_-[BR] ;COMPLEMENT DIVISOR
; 4544 [AR]_[AR].OR.#, ;ADJUST SIGN OF QUOTIENT
; 4545 #/40000, J/DVSUB3 ;USE 9 CHIPS
; 4546 DIVSET: [AR]_[AR]*.5
; 4547 [BR]_[BR]*.5
; 4548 [BR]_[BR]*.5
; 4549 FIRST DIV STEP
; 4550 ;HERE IS THE MAIN DIVIDE LOOP
; 4551 =0
; 4552 DIVIDE: DIV STEP, J/DIVIDE
; 4553 [T1]_[T1]*2 LONG, DIVIDE/1, DIV
; 4554 [AR]_[AR]*.5, SKIP DPO
; 4555 =0
; 4556 FIX++: [AR]_[AR]*2 LONG, J/FIX1++
; 4557 [AR]_[AR]+[BR], J/FIX++
; 4558 FIX1++: [AR]_[AR]*2 LONG
; 4559 Q_[MASK].AND.Q
; 4560 READ [ARX], 3T, ;RETURN TO 1 OF 4 PLACES
; 4561 DISP/1, ;BASED ON SIGN OF RESULT
; 4562 J/14 ;RETURN
; 4563
```

```
U 3036, 1264,3333,0003,4174,4007,0520,0000,0000,0000
U 1264, 3040,4221,0004,4174,4007,0700,0000,0000,0000
U 1265, 1266,2222,0000,4174,4007,0621,4000,0000,0000
U 1266, 3037,7441,0303,4174,4007,0700,0000,0000,0000
U 1267, 3037,2441,0303,4174,4007,0700,4000,0000,0000

U 3037, 3040,3771,0004,4374,4007,0700,0000,0010,0000
U 3040, 1270,3333,0005,4174,4007,0520,0000,0000,0000

U 1270, 3042,3446,0303,4174,4007,0700,0000,0000,0000
U 1271, 3041,2441,0505,4174,4007,0700,4000,0000,0000

U 3041, 1270,3551,0404,4374,4007,0700,0000,0004,0000
U 3042, 3043,3447,0303,4174,4007,0700,0000,0000,0000
U 3043, 3044,3447,0505,4174,4007,0700,0000,0000,0000
U 3044, 3045,3447,0505,4174,4007,0700,0000,0000,0000
U 3045, 1272,1114,0503,4174,4067,0630,6000,0060,0000

U 1272, 1272,0114,0503,4174,4067,0630,2100,0060,0000
U 1273, 3046,3444,1717,4174,4067,0700,0100,0000,0000
U 3046, 1274,3447,0303,4174,4007,0520,0000,0000,0000

U 1274, 3047,3444,0303,4174,4007,0700,0000,0000,0000
U 1275, 1274,0111,0503,4174,4007,0700,0000,0000,0000
U 3047, 3050,3444,0303,4174,4007,0700,0000,0000,0000
U 3050, 3051,4002,1200,4174,4007,0700,0000,0000,0000

U 3051, 0014,3333,0004,4174,4000,1701,0000,0000,0000
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 4564 .TOC "ARITHMETIC -- DOUBLE DIVIDE SUBROUTINE"  
; 4565 .IF/FULL  
; 4566 ;CALL WITH:  
; 4567 ; AR!ARX!Q = 3 WORD DV'END  
; 4568 ; BR!BRX = 2 WORD DV'SOR  
; 4569 ;RETURN 2 WITH:  
; 4570 ; AR!ARX = 2 WORD REMAINDER  
; 4571 ; CORRECT IF POSITIVE (Q IS ODD)  
; 4572 ; WRONG (BY BR!BRX) IF NEGATIVE (Q IS EVEN)  
; 4573 ; Q = 1 WORD QUOTIENT  
; 4574 ;CALLER MUST CHECK FOR ZERO DIVIDE PRIOR TO CALL  
; 4575 ;  
; 4576 ;NOTE: THIS SUBROUTINE ONLY WORKS FOR POSITIVE NUMBERS  
; 4577 ;  
; 4578 =0  
; 4579 ;HERE FOR NORMAL STARTUP  
; 4580 DBLDIV: [ARX]_([ARX]-[BRX])*2 LONG, ;SUBTRACT LOW WORD  
; 4581 LSHC, J/DIVHI ;GO ENTER LOOP  
; 4582 ;SKIP ENTRY POINT IF FINAL STEP IN PREVIOUS ENTRY WAS IN ERROR  
; 4583 [ARX]_([ARX]+[BRX])*2 LONG, ;CORRECTION STEP  
; 4584 LSHC, J/DIVHI ;GO ENTER LOOP  
; 4585 ;  
; 4586 ;HERE IS DOUBLE DIVIDE LOOP  
; 4587 DIVHI: AD/A+B, ;ADD (HARDWARE MAY OVERRIDE)  
; 4588 A/BR, B/AR, ;OPERANDS ARE AR AND BR  
; 4589 DEST/AD*2, ;SHIFT LEFT  
; 4590 SHSTYLE/NORM, ;SET SHIFT PATHS (SEE DPE1)  
; 4591 MULTI PREC/1, ;INJECT SAVED BITS  
; 4592 STEP SC ;COUNT DOWN LOOP  
; 4593 =0 AD/A+B, ;ADD (HARDWARE MAY OVERRIDE)  
; 4594 A/BRX, B/ARX, ;LOW WORDS  
; 4595 DEST/Q_Q*2, ;SHIFT WHOLE MESS LEFT  
; 4596 SHSTYLE/DIV, ;SET SHIFT PATHS (SEE DPE1)  
; 4597 DIVIDE/1, ;SAVE BITS  
; 4598 J/DIVHI ;KEEP LOOPING  
; 4599 ;HERE WHEN ALL DONE  
; 4600 DEST/Q_Q*2, DIV, ;SHIFT IN LAST Q BIT  
; 4601 DIVIDE/1, ;GENERATE BIT  
; 4602 B/HR, RETURN [2] ;ZERO HR AND RETURN  
; 4603
```

U 1276, 3052, 1114, 0604, 4174, 4057, 0700, 4000, 0000, 0000

U 1277, 3052, 0114, 0604, 4174, 4057, 0700, 0000, 0000, 0000

U 3052, 1300, 0115, 0503, 4174, 4007, 0630, 2040, 0060, 0000

U 1300, 3052, 0114, 0604, 4174, 4067, 0700, 0100, 0000, 0000

U 1301, 0002, 4444, 0002, 4174, 4064, 1700, 0100, 0000, 0000

; T10KL.MCR[10,1141] 15:34 27-JULY-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 3053, 3054, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1443
U 3054, 1302, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1443
U 1302, 3057, 7772, 0000, 1274, 4007, 0701, 0000, 0000, 1442
U 1303, 3055, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1442
U 3055, 1304, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1442
U 1304, 3060, 7772, 0000, 1274, 4007, 0701, 0000, 0000, 1441
U 1305, 3056, 1772, 0000, 1274, 4007, 0701, 4000, 0000, 1441
U 3056, 1306, 4003, 0000, 1174, 4007, 0621, 0400, 0000, 1441
U 1306, 3061, 7771, 0003, 0274, 4007, 0700, 0000, 0000, 0000
U 1307, 3061, 1771, 0003, 0274, 4007, 0701, 4000, 0000, 0000
U 3057, 1304, 4003, 0000, 1174, 4007, 0700, 0400, 0000, 1442
U 3060, 1306, 4003, 0000, 1174, 4007, 0700, 0400, 0000, 1441
U 3061, 0024, 3440, 0303, 0174, 4004, 1700, 0400, 0000, 0000

U 3062, 3063, 4551, 0404, 4374, 0007, 0700, 0000, 0037, 7777
U 3063, 1310, 2441, 0404, 4174, 4007, 0621, 4000, 0000, 0000
U 1310, 2236, 7441, 0303, 4174, 4467, 0700, 0000, 0001, 0001
U 1311, 2236, 2441, 0303, 4174, 4467, 0701, 4000, 0001, 0001

U 3064, 1312, 2441, 0404, 4174, 4007, 0621, 4000, 0000, 0000
U 1312, 2236, 7441, 0303, 4174, 4007, 0700, 0000, 0000, 0000
U 1313, 2236, 2441, 0303, 4174, 4007, 0700, 4000, 0000, 0000

```
; 4604 .TOC "ARITHMETIC -- SUBROUTINES FOR ARITHMETIC"
; 4605
; 4606 ;QUAD WORD NEGATE
; 4607 ;ARGUMENT IN AC!AC1!AC2!AC3
; 4608 ;LEAVES COPY OF AC!AC1 IN AR!Q
; 4609 ;RETURNS TO CALL!24
; 4610 QDNEG: Q_-AC[3]
; 4611 AC[3]_Q.AND.[MAG], ;PUT BACK LOW WORD
; 4612 SKIP AD.EQ.O ;SEE IF ANY CARRY
; 4613 =O
; 4614 COM2A: Q_.NOT.AC[2], J/COM2 ;CARRY--DO 1'S COMPLEMENT
; 4615 Q_-AC[2] ;NEXT WORD
; 4616 AC[2]_Q.AND.[MAG], ;PUT BACK WORD
; 4617 SKIP AD.EQ.O
; 4618 =O
; 4619 COM1A: Q_.NOT.AC[1], J/COM1
; 4620 Q_-AC[1]
; 4621 AC[1]_Q.AND.[MAG],
; 4622 SKIP AD.EQ.O
; 4623 =O
; 4624 COMOA: [AR]_.NOT.AC, J/COMO
; 4625 [AR]_-AC, 3T, J/COMO
; 4626
; 4627 COM2: AC[2]_Q.AND.[MAG], J/COM1A
; 4628 COM1: AC[1]_Q.AND.[MAG], J/COMOA
; 4629 COMO: AC_[AR], RETURN [24]
; 4630 .ENDIF/FULL
; 4631
; 4632 ;DOUBLE WORD NEGATE
; 4633 ;ARGUMENT IN AR AND ARX
; 4634 ;RETURNS TO CALL!2
; 4635
; 4636 DBLNEG: CLEAR ARXO ;FLUSH DUPLICATE SIGN
; 4637 DBLNGA: [ARX]_-[ARX], ;FLIP LOW WORD
; 4638 SKIP AD.EQ.O ;SEE IF CARRY
; 4639 =O [AR]_.NOT.[AR], ;NO CARRY-- 1 COMP
; 4640 AD FLAGS, J/CLARXO ;CLEAR LOW SIGN
; 4641 [AR]_-[AR], ;CARRY
; 4642 AD FLAGS, 3T, J/CLARXO
; 4643
; 4644 ;SAME THING BUT DOES NOT SET PC FLAGS
; 4645 DBLNG1: [ARX]_-[ARX], SKIP AD.EQ.O
; 4646 =O [AR]_.NOT.[AR], J/CLARXO
; 4647 [AR]_-[AR], J/CLARXO
; 4648
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 4649 .NOBIN
; 4650 .TOC "BYTE GROUP -- IBP, ILDB, LDB, IDPB, DPB"
; 4651
; 4652
; 4653 ;ALL FIVE INSTRUCTIONS OF THIS GROUP ARE CALLED WITH THE BYTE POINTER
; 4654 ;IN THE AR. ALL INSTRUCTIONS SHARE COMMON SUBROUTINES.
; 4655
; 4656 ;IBP OR ADJBP
; 4657 ;IBP IF AC#0, ADJBP OTHERWISE
; 4658 ; HERE WITH THE BASE POINTER IN AR
; 4659
; 4660 ;HERE IS A MACRO TO DO IBP. WHAT HAPPENS IS:
; 4661 ; THE AR IS PUT ON THE DP.
; 4662 ; THE BR IS LOADED FROM THE DP WITH BITS 0-5 FROM SCAD
; 4663 ; THE SCAD COMPUTES P-S
; 4664 ; IBPS IS CALLED WITH A 4-WAY DISPATCH ON SCADO AND FIRST-PART-DONE
; 4665 ;THE MACRO IS WRITTEN WITH SEVERAL SUB-MACROS BECAUSE OF RESTRICTIONS
; 4666 ; IN THE MICRO ASSEMBLER
; 4667
; 4668 IBP DP "AD/D, DEST/A, A/AR, B/BR, DBUS/DBM, DBM/DP, BYTE/BYTE1"
; 4669 IBP SCAD "SCAD/A-B, SCADA/BYTE1, SCADB/SIZE"
; 4670 IBP SPEC "SCAD DISP, SKIP FPD"
; 4671 CALL IBP "IBP DP, IBP SCAD, IBP SPEC, CALL [IBPS], DT/3T"
; 4672
; 4673 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"
; 4674
; 4675 ;THE FOLLOWING MACRO IS USED FOR COUNTING SHIFTS IN THE BYTE ROUTINES. IT
; 4676 ; USES THE FE AND COUNTS BY 8. NOTE: BYTE STEP IS A 2S WEIGHT SKIP NOT 1S.
; 4677 BYTE STEP "SCAD/A+B,SCADA/S#,S#/1770,SCADB/FE,LOAD FE, 3T,SCAD DISP"
; 4678
```


; T1OKL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```

; 4679      .BIN
; 4680
; 4681      .DCODE
D 0133, 0015, 1610, 1100      ; 4682 133:  R,      AC,      J/IBP      ;OR ADJBP
D 0134, 0000, 1620, 1500      ; 4683 134:  R,W TEST,  J/ILDB      ;CAN'T USE RPW BECAUSE OF FP
; 4684      R,      J/LDB
D 0135, 0000, 1624, 1100      ; 4685      R,W TEST,  J/IDPB
D 0136, 0000, 1630, 1500      ; 4686      R,      J/DPB
D 0137, 0000, 1634, 1100      ; 4687      .UCODE
; 4688      1610:
U 1610, 0240, 4443, 0000, 4174, 4007, 0360, 0000, 0000, 0000      ; 4688      IBP:      SKIP IF ACO      ;SEE IF ADJBP
; 4689      =000      WORK[ADJPTR]_[AR],      ;SAVE POINTER
U 0240, 3122, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0223      ; 4690      J/ADJBP      ;GO ADJUST BYTE POINTER
U 0241, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000      ; 4691      =001      CALL IBP      ;BUMP BYTE POINTER
U 0245, 0110, 3443, 0100, 4174, 4156, 4700, 0200, 0014, 0012      ; 4692      =101      DONE      ;POINTER STORED
; 4693      =
; 4694      =
; 4695
; 4696      1620:
U 1620, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000      ; 4697      ILDB:      CALL IBP      ;BUMP BYTE POINTER
; 4698      1624:
; 4699      LDB:      READ [AR],      ;LOOK AT POINTER
; 4700      LOAD BYTE EA, FE_P, 3T,      ;GET STUFF OUT OF POINTER
U 1624, 3072, 3333, 0003, 4174, 4217, 0701, 1010, 0073, 0500      ; 4701      CALL [BYTEA]      ;COMPUTE EFFECTIVE ADDRESS
U 1625, 0414, 3443, 0100, 4174, 4007, 0700, 0200, 0014, 0012      ; 4702 1625:  VMA [PC], FETCH      ;START FETCH OF NEXT INST
; 4703      =0*      READ [AR],      ;LOOK AT POINTER
; 4704      FE_FE.AND.S#, S#/0770,      ;MASK OUT JUNK IN FE
; 4705      BYTE DISP,      ;DISPATCH ON BYTE SIZE
U 0414, 0340, 3333, 0003, 4174, 4006, 5701, 1010, 0051, 0770      ; 4706      CALL [LDB1]      ;GET BYTE
U 0416, 0603, 3440, 0303, 0174, 4467, 0700, 0400, 0005, 0000      ; 4707      AC [AR], CLR FPD,      ;STORE AC
; 4708      J/NIDISP      ;GO DO NEXT INST
; 4709
; 4710      1630:
U 1630, 0350, 3770, 0305, 4334, 4016, 7351, 0010, 0033, 6000      ; 4711      IDPB:      CALL IBP      ;BUMP BYTE POINTER
; 4712      1634:
U 1634, 3065, 3775, 0004, 0274, 4007, 0701, 0000, 0000, 0000      ; 4713      DPB:      [ARX]_AC*2      ;PUT 7 BIT BYTE IN 28-34
; 4714      AD/A, A/ARX, SCAD/A,      ;PUT THE BYTE INTO
; 4715      SCADA/BYTES, 3T,      ; INTO THE FE REGISTER
; 4716      LOAD FE      ; FE REGISTER
U 3065, 3066, 3443, 0400, 4174, 4007, 0701, 1000, 0077, 0000      ; 4717      [ARX]_AC      ;PUT BYTE IN ARX
U 3066, 0264, 3771, 0004, 0276, 6007, 0700, 0000, 0000, 0000      ; 4718      =100      READ [AR],      ;LOOK AT BYTE POINTER
; 4719      LOAD BYTE EA,      ;LOAD UP EFFECTIVE ADDRESS
U 0264, 3072, 3333, 0003, 4174, 4217, 0700, 0010, 0000, 0500      ; 4720      CALL [BYTEA]      ;COMPUTE EFFECTIVE ADDRESS
; 4721      READ [AR],      ;LOOK AT POINTER AGAIN
; 4722      BYTE DISP,      ;DISPATCH ON SIZE
U 0265, 0360, 3333, 0003, 4174, 4006, 5701, 0010, 0000, 0000      ; 4723      CALL [DPB1]      ;GO STORE BYTE
U 0267, 1400, 4443, 0000, 4174, 4467, 0700, 0000, 0005, 0000      ; 4724      =111      CLR FPD, J/DONE      ;ALL DONE
; 4725      =
; 4726
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 4727 .TOC "BYTE GROUP -- INCREMENT BYTE POINTER SUBROUTINE"
; 4728
; 4729 =00
U 0350, 3070,3441,0503,4174,4007,0700,0200,0003,0002 ; 4730 IBPS: [AR]_[BR], START WRITE, J/IBPX ;NO OVERFLOW, BR HAS ANSWER
U 0351, 0004,4443,0000,4174,4004,1700,0000,0000,0000 ; 4731 RETURN [4] ;FIRST PART DONE SET
U 0352, 3067,3770,0503,4334,4017,0700,0000,0032,6000 ; 4732 SET P TO 36-S, J/NXTWRD ;WORD OVERFLOW
U 0353, 0004,4443,0000,4174,4004,1700,0000,0000,0000 ; 4733 RETURN [4] ;FPD WAS SET IGNORE OVERFLOW
; 4734 ; AND RETURN
; 4735
; 4736 NXTWRD: [AR]_[AR]+1, HOLD LEFT, START WRITE ;BUMP Y AND RETURN
U 3067, 3070,0111,0703,4170,4007,0700,0200,0003,0002 ; 4737 IBPX: MEM WRITE, MEM_[AR], RETURN [4]
U 3070, 0004,3333,0003,4175,5004,1701,0200,0000,0002 ; 4738
; 4739
; 4740 .TOC "BYTE GROUP -- BYTE EFFECTIVE ADDRESS EVALUATOR"
; 4741
; 4742 ;ENTER WITH POINTER IN AR
; 4743 ;RETURN1 WITH (EA) IN VMA AND WORD IN BR
; 4744 BYTEAS: EA MODE DISP, ;HERE TO AVOID FPD
; 4745 J/BYTEAO ;GO COMPUTE EA
U 3071, 0070,4443,0000,2174,4006,6700,0000,0000,0000 ; 4746 BYTEA: SET FPD, ;SET FIRST-PART-DONE
; 4747 EA MODE DISP ;DISPATCH
U 3072, 0070,4443,0000,2174,4466,6700,0000,0003,0000 ; 4748 =100*
; 4749 BYTEAO: VMA_[AR]+XR, ;INDEXING
; 4750 START READ, ;FETCH DATA WORD
; 4751 PXCT BYTE DATA, ;FOR PXCT
; 4752 J/BYTFET ;GO WAIT
U 0070, 3074,0553,0300,2274,4007,0700,0200,0004,0712 ; 4753 VMA_[AR], ;PLAIN
; 4754 START READ, ;START CYCLE
; 4755 PXCT BYTE DATA, ;FOR PXCT
U 0072, 3074,3443,0300,4174,4007,0700,0200,0004,0712 ; 4756 J/BYTFET ;GO WAIT
; 4757 VMA_[AR]+XR, ;BOTH
; 4758 START READ, ;START CYCLE
; 4759 PXCT BYTE PTR EA, ;FOR PXCT
U 0074, 3073,0553,0300,2274,4007,0700,0200,0004,0512 ; 4760 J/BYTIND ;GO DO INDIRECT
; 4761 VMA_[AR], ;JUST @
; 4762 START READ, ;START READ
U 0076, 3073,3443,0300,4174,4007,0700,0200,0004,0512 ; 4763 PXCT BYTE PTR EA ;FOR PXCT
; 4764 BYTIND: MEM READ, ;WAIT FOR @ WORD
; 4765 [AR]_MEM, ;PUT IN AR
; 4766 HOLD LEFT, ;JUST IN RH (SAVE P & S)
; 4767 LOAD BYTE EA, ;LOOP BACK
U 3073, 3071,3771,0003,4361,5217,0700,0200,0000,0502 ; 4768 J/BYTEAS ; ..
; 4769
; 4770 BYTFET: MEM READ, ;WAIT FOR BYTE DATA
; 4771 [BR] MEM.AND.MASK, ; WORD. UNSIGNED
; 4772 RETURN [1] ;RETURN TO CALLER
; 4773
```

; T10KLMICR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

KS.J MICROCODE V124, 27-JUL-84 Page 130
BYTE GROUP -- LOAD BYTE SUBROUTINE

```
; 4774 .TOC "BYTE GROUP -- LOAD BYTE SUBROUTINE"  
; 4775  
; 4776 ;CALL WITH:  
; 4777 ; WORD IN BR  
; 4778 ; POINTER IN AR  
; 4779 ; P IN FE  
; 4780 ; BYTE DISPATCH  
; 4781 ;RETURN2 WITH BYTE IN AR  
; 4782 LDB SCAD "SCAD/A,BYTE/BYTE5"  
; 4783 7-BIT LDB "AD/D,DBUS/DBM,DBM/DP,DEST/A,A/BR,B/BR, LDB SCAD"  
; 4784  
; 4785 =000  
; 4786 LDB1: GEN 17-FE, 3T, ;GO SEE IF ALL THE BITS  
; 4787 SCAD DISP, ; ARE IN THE LEFT HALF  
; 4788 J/LDBSWP ;GO TO LDBSWP & SKIP IF LH  
; 4789  
; 4790 ;HERE ARE THE 7-BIT BYTES  
; 4791 =001 7-BIT LDB, SCADA/BYTE1, J/LDB7  
; 4792 =010 7-BIT LDB, SCADA/BYTE2, J/LDB7  
; 4793 =100 7-BIT LDB, SCADA/BYTE3, J/LDB7  
; 4794 =101 7-BIT LDB, SCADA/BYTE4, J/LDB7  
; 4795 =111 7-BIT LDB, SCADA/BYTE5, J/LDB7  
; 4796 =  
; 4797  
; 4798 ;FOR 7-BIT BYTES WE HAVE BYTE IN BR 28-35 AND JUNK IN REST OF BR.  
; 4799 ; WE JUST MASK THE SELECTED BYTE AND SHIFT ONE PLACE RIGHT.  
; 4800 LDB7: AD/ZERO,RSRC/DA, ;LH_ZERO, RH_D.AND.A  
; 4801 DBUS/DBM,DBM/#,#/376, ;D INPUT IS 376  
; 4802 A/BR, ;A IS BR  
; 4803 B/AR, ;PUT RESULT IN AR  
; 4804 DEST/AD*.5, 3T, ;SHIFT RESULT 1 PLACE  
; 4805 RETURN [2] ;RETURN TO CALLER  
; 4806
```

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

U 0341, 3075,3770,0505,4334,4057,0700,0000,0073,0000

U 0342, 3075,3770,0505,4334,4057,0700,0000,0074,0000

U 0344, 3075,3770,0505,4334,4057,0700,0000,0075,0000

U 0345, 3075,3770,0505,4334,4057,0700,0000,0076,0000

U 0347, 3075,3770,0505,4334,4057,0700,0000,0077,0000

U 3075, 0002,4257,0503,4374,4004,1701,0000,0000,0376

; T1OKL.MCR[10,1141] 15:34 27-JULY-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, 3077,4443,0000,4174,4007,0700,1000,0031,0000
U 0412, 3076,3770,0505,4344,4007,0700,0000,0000,0000

U 3076, 3077,4221,0005,4174,0007,0700,1000,0031,0220

U 3077, 3100,3447,0505,4174,4007,0700,1020,0041,0010
U 3100, 3101,3333,0003,4174,4007,0700,1000,0031,7770
U 3101, 3102,4222,0000,4174,4007,0700,0000,0000,0000

U 3102, 3103,4224,0003,4174,4027,0700,1020,0041,0010
U 3103, 3104,4224,0003,4174,4027,0700,0000,0000,0000
U 3104, 0002,4001,0503,4174,4004,1700,0000,0000,0000

```
; 4807 ;HERE FOR NORMAL BYTES
; 4808 =00
; 4809 LDBSWP: FE_-FE, ;MAKE P NEGATIVE
; 4810 J/LDBSH ;JOIN MAIN LDB LOOP
; 4811 =10 [BR]_[BR] SWAP ;SHIFT 18 STEPS
; 4812 =
; 4813 [BR]_0, HOLD RIGHT, ;PUT ZERO IN LH
; 4814 FE_-FE+S#, S#/220 ;UPDATE FE
; 4815 LDBSH: [BR]_[BR]*.5, ;SHIFT RIGHT
; 4816 FE_FE+10, ;UPDATE THE FE
; 4817 MULTI SHIFT/1 ;FAST SHIFT
; 4818 READ [AR], FE_-S-10 ;GET SIZE
; 4819 Q_0 ;CLEAR Q
; 4820 GEN MSK [AR], ;PUT MASK IN Q (WIPEOUT AR)
; 4821 FE_FE+10, ;COUNT UP ALL STEPS
; 4822 MULTI SHIFT/1 ;FAST SHIFT
; 4823 GEN MSK [AR] ;ONE MORE BIT
; 4824 [AR]_[BR].AND.Q, RETURN [2]
; 4825
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
; 4826 .NOBIN
; 4827 .TOC "BYTE GROUP -- DEPOSIT BYTE IN MEMORY"
; 4828
; 4829 ;FLOW FOR DPB (NOT 7-BIT BYTE)
; 4830 ;
; 4831 ;FIRST SET ARX TO -1 AND Q TO ZERO AND ROTATE LEFT
; 4832 ; S PLACES GIVING:
; 4833
; 4834 ;
; 4835 ; ARX Q
; 4836 ; +-----+-----+
; 4837 ; !111111111111000000!000000000000111111!
; 4838 ; +-----+-----+
; 4839 ; !<--->!
; 4840 ; S BITS
; 4841
; 4842 ;NOW THE AC IS LOAD INTO THE ARX AND BOTH THE ARX AND Q
; 4843 ; ARE SHIFTED LEFT P BITS GIVING:
; 4844
; 4845 ; +-----+-----+
; 4846 ; !?????BBBBBB000000!000000111111000000!
; 4847 ; +-----+-----+
; 4848 ; <---><---> <---><--->
; 4849 ; JUNK BYTE MASK P BITS
; 4850
; 4851
; 4852 ;AT THIS POINT WE ARE ALMOST DONE. WE NEED TO AND
; 4853 ; THE BR WITH .NOT. Q TO ZERO THE BITS FOR THE BYTE
; 4854 ; AND AND ARX WITH Q TO MASK OUT THE JUNK THIS GIVES:
; 4855
; 4856 ; ARX
; 4857 ; +-----+
; 4858 ; !00000BBBBBB000000!
; 4859 ; +-----+
; 4860
; 4861 ; AR
; 4862 ; +-----+
; 4863 ; !DDDDDD000000DDDDDD!
; 4864 ; +-----+
; 4865
; 4866 ;WE NOW OR THE AR WITH ARX TO GENERATE THE ANSWER.
; 4867
```

Produced on Advanced Information Services Electronic Laser Printer. PKO/ISS. DTN: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 4868          .BIN
; 4869
; 4870 ;DEPOSIT BYTE SUBROUTINE
; 4871 ;CALL WITH:
; 4872 ;     BYTE POINTER IN AR
; 4873 ;     BYTE TO STORE IN ARX
; 4874 ;     WORD TO MERGE WITH IN BR
; 4875 ;     (E) OF BYTE POINTER IN VMA
; 4876 ;     7-BIT BYTE IN FE
; 4877 ;     BYTE DISPATCH
; 4878 ;RETURN2 WITH BYTE IN MEMORY
; 4879 ;
; 4880 DPB SCAD          "SCAD/A+B,SCADA/S#,SCADB/FE,S#/O"
; 4881 7-BIT DPB       "AD/D,DEST/A,A/BR,DBUS/DBM,DBM/DP,B/AR, DPB SCAD"
; 4882
; 4883 =000
; 4884 DPB1:  READ [AR], FE_-S-10, J/DPBSLO ;NOT SPECIAL
; 4885 =001  7-BIT DPB, BYTE/BYTE1, J/DPB7
; 4886 =010  7-BIT DPB, BYTE/BYTE2, J/DPB7
; 4887 =100  7-BIT DPB, BYTE/BYTE3, J/DPB7
; 4888 =101  7-BIT DPB, BYTE/BYTE4, J/DPB7
; 4889 =111  7-BIT DPB, BYTE/BYTE5, J/DPB7
; 4890 =
; 4891 DPB7:  [MAG]_[MASK]*.5, START WRITE
; 4892 MEM WRITE, MEM_[AR], RETURN [2]
; 4893
; 4894
; 4895 DPBSLO: Q Q          ;CLEAR Q
; 4896 GEN MSK [MAG],      ;GENERATE MASK IN Q (ZAP MAG)
; 4897 FE_FE+10,          ;COUNT STEPS
; 4898 MULTI SHIFT/1     ;FAST SHIFT
; 4899 GEN MSK [MAG]     ;ONE MORE BITS
; 4900 READ [AR], 3T, FE_P ;AMOUNT TO SHIFT
; 4901 FE_FE.AND.S#, S#/O770 ;MASK OUT JUNK
; 4902 Q_Q.AND.[MASK],   ;CLEAR BITS 36 AND 37
; 4903 FE_-FE           ;MINUS NUMBER OF STEPS
; 4904 [ARX]_[ARX]*2 LONG, ;SHIFT BYTE AND MASK
; 4905 FE_FE+10,        ;COUNT OUT STEPS
; 4906 MULTI SHIFT/1     ;FAST SHIFT
; 4907 ;AT THIS POINT WE HAVE DONE ALL THE SHIFTING WE NEED. THE BYTE IS
; 4908 ; IN ARX AND THE MASK IS IN Q.
; 4909 [AR]_.NOT.Q
; 4910 [AR]_[AR].AND.[BR]
; 4911 [ARX]_[ARX].AND.Q
; 4912 [AR]_[AR].OR.[ARX],
; 4913 J/DPB7
; 4914
```

U 0360, 3107,3333,0003,4174,4007,0700,1000,0031,7770
U 0361, 3105,3770,0503,4334,4017,0700,0000,0041,0000
U 0362, 3105,3770,0503,4334,4027,0700,0000,0041,0000
U 0364, 3105,3770,0503,4334,4037,0700,0000,0041,0000
U 0365, 3105,3770,0503,4334,4047,0700,0000,0041,0000
U 0367, 3105,3770,0503,4334,4057,0700,0000,0041,0000

U 3105, 3106,3447,1200,4174,4007,0700,0200,0003,0002
U 3106, 0002,3333,0003,4175,5004,1701,0200,0000,0002

U 3107, 3110,4222,0000,4174,4007,0700,0000,0000,0000

U 3110, 3111,4224,0000,4174,4027,0700,1020,0041,0010
U 3111, 3112,4224,0000,4174,4027,0700,0000,0000,0000
U 3112, 3113,3333,0003,4174,4007,0701,1000,0073,0000
U 3113, 3114,4443,0000,4174,4007,0700,1000,0051,0770

U 3114, 3115,4002,1200,4174,4007,0700,1000,0031,0000

U 3115, 3116,3444,0404,4174,4007,0700,1020,0041,0010

U 3116, 3117,7221,0003,4174,4007,0700,0000,0000,0000
U 3117, 3120,4111,0503,4174,4007,0700,0000,0000,0000
U 3120, 3121,4001,0404,4174,4007,0700,0000,0000,0000

U 3121, 3105,3111,0403,4174,4007,0700,0000,0000,0000

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 4915 .TOC "BYTE GROUP -- ADJUST BYTE POINTER"
; 4916 .IF/FULL
; 4917 ;FIRST THE NUMBER OF BYTES PER WORD IS COMPUTED FROM THE
; 4918 ; FOLLOWING FORMULA:
; 4919 ;
; 4920 ; ( P ) ( 36-P )
; 4921 ; BYTES PER WORD = INT( --- ) + INT( ---- )
; 4922 ; ( S ) ( S )
; 4923 ;
; 4924 ;THIS GIVES 2 BYTES PER WORD FOR THE FOLLOWING 12 BIT BYTE:
; 4925 ; !=====!
; 4926 ; ! 6 !//////////! 12 ! 6 !
; 4927 ; !=====!
; 4928 ; P=18 AND S=12
; 4929 ;
; 4930 ;WE GET 3 BYTES/WORD IF THE BYTES FALL IN THE NATURAL PLACE:
; 4931 ; !=====!
; 4932 ; ! 12 !\\\\\\\\\\\\\\\\! 12 !
; 4933 ; !=====!
; 4934 ; P=12 AND S=12
; 4935 ;
; 4936 ;WE COME HERE WITH THE BYTE POINTER IN AR, AND ADJPTR
; 4937 ADJBP: [ARX]_[AR] SWAP, ;MOVE SIZE OVER
; 4938 SC_9. ;READY TO SHIFT
; 4939 =0
; 4940 ADJBPO: [ARX]_[ARX]*.5, ;SHIFT P OVER
; 4941 STEP SC, ; ..
; 4942 J/ADJBPO ; ..
; 4943 [ARX]_([ARX].AND.#)*.5, ;SHIFT AND MASK
; 4944 3T, ;WAIT
; 4945 #/176 ;6 BIT MASK
; 4946 [ARX]_#, ;CLEAR LH
; 4947 #/0, ; ..
; 4948 HOLD RIGHT ; ..
; 4949 WORK[ADJP]_[ARX] ;SAVE P
; 4950 [BR]_([AR].AND.#)*.5, ;START ON S
; 4951 3T, ;EXTRACT S
; 4952 #/007700 ; ..
; 4953 [BR]_[BR] SWAP, ;SHIFT 18 PLACES
; 4954 SC_3 ; ..
; 4955 [BR]_0, ;CLEAR LH
; 4956 HOLD RIGHT ; ..
; 4957
```

U 3122, 1314,3770,0304,4344,4007,0700,2000,0071,0011
U 1314, 1314,3447,0404,4174,4007,0630,2000,0060,0000
U 1315, 3123,4557,0404,4374,4007,0701,0000,0000,0176
U 3123, 3124,3771,0004,4374,0007,0700,0000,0000,0000
U 3124, 3125,3333,0004,7174,4007,0700,0400,0000,0221
U 3125, 3126,4557,0305,4374,4007,0701,0000,0000,7700
U 3126, 3127,3770,0505,4344,4007,0700,2000,0071,0003
U 3127, 1316,4221,0005,4174,0007,0700,0000,0000,0000

; T1OKL.MCR[10,1141] 15:34 27-JULY-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 1316, 1316,3447,0505,4174,4007,0630,2000,0060,0000

U 1317, 1320,3333,0005,7174,4007,0621,0400,0000,0222

U 1320, 0660,3442,0400,4174,4007,0700,2000,0071,0042

U 1321, 1404,3771,0003,7274,4007,0701,0000,0000,0223

U 0660, 0370,3771,0003,4374,4007,0700,0010,0000,0000

U 0662, 3130,3223,0000,7174,4007,0700,0400,0000,0224

U 3130, 3131,3772,0000,4370,4007,0700,0000,0000,0044

U 3131, 3132,1662,0000,7274,4007,0701,4000,0000,0221

U 3132, 3133,3771,0005,7274,4007,0701,0000,0000,0222

U 3133, 0664,4443,0000,4174,4007,0700,2000,0071,0042

U 0664, 0370,3771,0003,4374,4007,0700,0010,0000,0000

U 0666, 3134,3333,0003,7174,4007,0700,0400,0000,0225

U 3134, 3135,3771,0003,4374,4007,0700,0000,0077,7777

U 3135, 1322,0661,0005,7274,4007,0622,0000,0000,0224

U 1322, 0550,0662,0000,0274,4007,0522,2000,0071,0042

U 1323, 0603,4443,0000,4174,4467,0700,0000,0051,1000

```
; 4958 =0
; 4959 ADJBP1: [BR]_[BR]*.5, ;SHIFT S OVER
; 4960 STEP_SC, ; ..
; 4961 J/ADJBP1 ; ..
; 4962 WORK[ADJS]_[BR], ;SALT S AWAY
; 4963 SKIP AD.EQ.0 ;SEE IF ZERO
; 4964 =0 Q_[ARX], ;DIVIDE P BY S
; 4965 SC_34., ;STEP COUNT
; 4966 J/ADJBP2 ;SKIP NEXT WORD
; 4967 [AR]_WORK[ADJPTR], J/MOVE ;S=0 -- SAME AS MOVE
; 4968 =0*
; 4969 ADJBP2: [AR]_#, ;FILL AR WITH SIGN BITS
; 4970 #/0, ;POSITIVE
; 4971 CALL [DIVSUB] ;GO DIVIDE
; 4972 WORK[ADJQ1]_Q ;SAVE QUOTIENT
; 4973 Q_#, ;COMPUTE (36-P)/S
; 4974 #/36., ; ..
; 4975 HOLD LEFT ;SMALL ANSWER
; 4976 Q_Q-WORK[ADJP] ;SUBTRACT P
; 4977 [BR]_WORK[ADJS] ;DIVIDE BY S
; 4978 SC_34. ;STEP COUNT
; 4979 =0* [AR]_#, ;MORE SIGN BITS
; 4980 #/0, ; ..
; 4981 CALL [DIVSUB] ;GO DIVIDE
; 4982 WORK[ADJR2]_[AR] ;SAVE REMAINDER
; 4983 [AR]_#, ;ASSUME NEGATIVE ADJ
; 4984 #/777777 ;EXTEND SIGN
; 4985 AD/D+Q, ;BR_(P/S)+((36-P)/S)
; 4986 DEST/AD, ; ..
; 4987 B/BR, ; ..
; 4988 RAMADR/#, ; ..
; 4989 DBUS/RAM, ; ..
; 4990 WORK/ADJQ1, ; ..
; 4991 4T, ; ..
; 4992 SKIP AD.EQ.0 ;SEE IF ZERO
; 4993 =0 Q_Q+AC, ;GET ADJUSTMENT
; 4994 SC_34., ;STEP COUNT
; 4995 SKIP DPO, ;GO DO DIVIDE
; 4996 4T, ;WAIT FOR DP
; 4997 J/ADJBP3 ;BELOW
; 4998 NO DIVIDE ;0 BYTES/WORD
; 4999
```


; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

U 0550, 0551, 3771, 0003, 4374, 4007, 0700, 0000, 0000, 0000
U 0551, 0370, 3333, 0005, 7174, 4007, 0700, 0410, 0000, 0226
U 0553, 3136, 3772, 0000, 4374, 0007, 0700, 0000, 0000, 0000
U 3136, 1324, 3333, 0003, 4174, 4007, 0421, 0000, 0000, 0000
U 1324, 3140, 0661, 0005, 7274, 4407, 0701, 0000, 0000, 0223
U 1325, 3137, 1002, 0700, 4170, 4007, 0700, 4000, 0000, 0000
U 3137, 1324, 0551, 0303, 7274, 4007, 0701, 0000, 0000, 0226

```
; 5000 ;WE NOW DIVIDE THE ADJUSTMENT BY THE BYTES PER WORD AND FORCE THE
; 5001 ; REMAINDER (R) TO BE A POSITIVE NUMBER (MUST NOT BE ZERO). THE
; 5002 ; QUOTIENT IS ADDED TO THE Y FIELD IN THE BYTE POINTER AND THE NEW
; 5003 ; P FIELD IS COMPUTED BY:
; 5004 ;
; 5005 ;           (           ( 36-P ))
; 5006 ; NEW P = 36-((R * S) + RMDR( ---- ))
; 5007 ;           (           (   S  ))
; 5008 ;
; 5009 ;WE NOW HAVE BYTES/WORD IN BR AND ADJUSTMENT IN Q. DIVIDE TO GET
; 5010 ; WORDS TO ADJUST BY.
; 5011 =00
; 5012 ADJBP3: [AR]_#, ;POSITIVE ADJUSTMENT
; 5013 #/O.
; 5014 WORK[ADJBPW]_[BR], ;SAVE BYTES/WORD & COMPUTE
; 5015 CALL [DIVSUB] ; ADJ/(BYTES/WORD)
; 5016 ;WE NOW WANT TO ADJUST THE REMAINDER SO THAT IT IS POSITIVE
; 5017 =11 Q_#, ;ONLY RIGHT HALF
; 5018 #/O, ; ..
; 5019 HOLD RIGHT ; ..
; 5020 =
; 5021 READ [AR], ;ALREADY +
; 5022 SKIP AD.LE.O ; ..
; 5023 =0
; 5024 ADJBP4: AD/D+Q, ;ADD Q TO POINTER AND STORE
; 5025 DEST/AD, ; ..
; 5026 B/BR, ;RESULT TO BR
; 5027 RAMADR/#, ;PTR IS IN RAM
; 5028 DBUS/RAM, ; ..
; 5029 WORK/ADJPTR, ; ..
; 5030 INH CRY18, ;JUST RH
; 5031 3T, ;WAIT FOR RAM
; 5032 J/ADJBP5 ;CONTINUE BELOW
; 5033 Q_Q-1, ;NO--MAKE Q SMALLER
; 5034 HOLD LEFT ; ..
; 5035 [AR]_[AR]+WORK[ADJBPW], ;MAKE REM BIGGER
; 5036 J/ADJBP4 ;NOW HAVE + REMAINDER
; 5037
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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 3140, 3141,3441,0306,4174,4007,0700,2000,0071,0043 ; 5038 ADJBP5: [BRX]_[AR], ; COMPUTE R*S
U 3141, 0062,3772,0000,7274,4007,0701,0000,0000,0222 ; 5039 SC 35. ; STEP COUNT
; 5040 Q_WORK[ADJS] ; GET S
U 0062, 2773,3446,0606,4174,4007,0700,0010,0000,0000 ; 5041 =01* [BRX]_[BRX]*.5 LONG, ; SHIFT OVER
; 5042 CALL [MULSUB] ; ..
; 5043 AD/D+Q, ; AR_(R*S)+RMDR(36-P)/S
; 5044 DEST/AD, ; ..
; 5045 B/AR, ; ..
; 5046 RAMADR/#, ; ..
; 5047 3T, ; ..
; 5048 DBUS/RAM, ; ..
U 0066, 3142,0661,0003,7274,4007,0701,0000,0000,0225 ; 5049 WORK/ADJR2 ; ..
; 5050 [AR]_(#[AR])*2, ; COMPUTE 36-AR
; 5051 3T, ; AND START LEFT
U 3142, 3143,2555,0303,4374,4007,0701,4000,0000,0044 ; 5052 #/36. ; ..
U 3143, 3144,3770,0303,4344,4007,0700,2000,0071,0011 ; 5053 [AR]_[AR] SWAP, ; PUT THE POSITION BACK
; 5054 SC 9. ; ..
; 5055 [AR]_#, ; CLEAR JUNK FROM RH
; 5056 #/O, ; ..
U 3144, 1326,3771,0003,4370,4007,0700,0000,0000,0000 ; 5057 HOLD LEFT ; ..
; 5058 =0 ; ..
; 5059 ADJBP6: [AR]_[AR]*2, ; LOOP OVER ALL BITS
; 5060 STEP SC, ; ..
; 5061 J/ADJBP6 ; ..
; 5062 [BR]_[BR].AND.#, ; ..
; 5063 #/007777, ; ..
U 1326, 1326,3445,0303,4174,4007,0630,2000,0060,0000 ; 5064 HOLD RIGHT ; ..
; 5065 AC [AR].OR.[BR], ; ALL DONE
; 5066 J/DONE ; ..
; 5067 .IFNOT/FULL ; ..
; 5068 ; ..
; 5069 ADJBP: UUO ; NO ADJBP IN SMALL
; 5070 ; MICROCODE
; 5071 .ENDIF/FULL
; 5072
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

; 5073 .NOBIN
; 5074 .TOC "BLT"
; 5075
; 5076 ;THIS CODE PROVIDES A GUARANTEED RESULT IN AC ON COMPLETION OF
; 5077 ; THE TRANSFER (EXCEPT IN THE CASE AC IS PART OF BUT NOT THE LAST WORD
; 5078 ; OF THE DESTINATION BLOCK). WHEN AC IS NOT PART OF THE DESTINATION
; 5079 ; BLOCK, IT IS LEFT CONTAINING THE ADDRESSES OF THE FIRST WORD FOLLOWING
; 5080 ; THE SOURCE BLOCK (IN THE LH), AND THE FIRST WORD FOLLOWING THE DEST-
; 5081 ; INATION BLOCK (IN THE RH). IF AC IS THE LAST WORD OF THE DESTINATION
; 5082 ; BLOCK, IT WILL BE A COPY OF THE LAST WORD OF THE SOURCE BLOCK.
; 5083
; 5084 ;IN ADDITION, A SPECIAL-CASE CHECK IS MADE FOR THE CASE IN WHICH EACH
; 5085 ; WORD STORED IS USED AS THE SOURCE OF THE NEXT TRANSFER. IN THIS CASE,
; 5086 ; ONLY ONE READ NEED BE PERFORMED, AND THAT DATA MAY BE STORED FOR EACH
; 5087 ; TRANSFER. THUS THE COMMON USE OF BLT TO CLEAR CORE IS SPEEDED UP.
; 5088

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

```
; 5089 .BIN
; 5090
; 5091 ;HERE TO SETUP FOR A BLT/UBABLT
; 5092
U 3146, 1330,3770,0604,4344,4007,0700,0000,0000,0000
; 5093 SETBLT: [ARX]_[BRX] SWAP ;COPY TO ARX (SRC IN RH)
; 5094 =0 VMA [ARX], ;ADDRESS OF FIRST WORD
; 5095 START READ,
; 5096 PXCT BLT SRC,
; 5097 CALL [CLARXL] ;CLEAR THE LEFT HALF OF
; 5098 [BRX] 0, ; BOTH SRC AND DEST
U 1330, 3676,3443,0400,4174,4007,0700,0210,0004,0712
; 5099 HOLD RIGHT
U 1331, 3147,4221,0006,4174,0007,0700,0000,0000,0000
; 5100 Q [AR]-[BRX] ;NUMBER OF WORDS TO MOVE
U 3147, 3150,2112,0306,4174,4007,0700,4000,0000,0000
; 5101 [BR] Q+1 ;LENGTH +1
U 3150, 3151,0001,0705,4174,4007,0700,0000,0000,0000
; 5102 [BR]_[BR] SWAP, ;COPY TO BOTH HALFS
; 5103 HOLD RIGHT
U 3151, 3152,3770,0505,4344,0007,0700,0000,0000,0000
; 5104 [BR] AC+[BR], ;FINAL AC
; 5105 INH CRY18 ;KEEP AC CORRECT IF DEST IS 777777
U 3152, 3153,0551,0505,0274,4407,0701,0000,0000,0000
; 5106 STATE_[BLT],RETURN [2] ;SET PAGE FAIL FLAGS
U 3153, 0002,3771,0013,4370,4004,1700,0000,0000,0001
; 5107
; 5108 .DCODE
D 0251, 0000,1640,2100
; 5109 251: I, J/BLT
; 5110 .UCODE
; 5111
; 5112 1640:
; 5113 BLT: [BRX] AC,CALL [SETBLT] ;FETCH THE AC (DEST IN RH)
; 5114 1642: AC [BR], ;STORE BACK IN AC
; 5115 CALL [LOADQ] ;LOAD FIRST WORD INTO Q
; 5116 1643: [BR]_[ARX]+1000001, ;SRC+1
; 5117 3T,
; 5118 HOLD LEFT
; 5119 [BR]-[BRX], 3T, ;IS THIS THE CORE CLEAR CASE
; 5120 SKIP ADR.EQ.0
; 5121 =0
; 5122 BLTLP1: VMA [BRX], ;NO, GET DEST ADR
; 5123 START WRITE, ;START TO STORE NEXT WORD
; 5124 PXCT BLT DEST, ;WHERE TO STORE
; 5125 J/BLTGO
; 5126
; 5127 ;SKIP TO NEXT PAGE IF CLEARING CORE
; 5128
```

; T10KL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

U 1333, 3155,3443,0600,4174,4007,0700,0200,0003,0312

U 3155, 1334,3223,0000,4174,4007,0671,0200,0000,0002
U 1334, 3160,4443,0000,4174,4007,0700,0000,0000,0000

U 1335, 1336,2113,0603,4174,4007,0521,4000,0000,0000

U 1336, 1400,4221,0013,4170,4007,0700,0000,0000,0000

U 1337, 1340,0111,0704,4174,4007,0370,0000,0000,0000

U 1340, 3155,0111,0706,4170,4007,0700,0200,0003,0312

U 1341, 3157,0111,0706,4170,4007,0700,0200,0003,0312

U 3156, 1332,3772,0000,4365,5007,0700,0200,0000,0002

U 3157, 3160,3223,0000,4174,4007,0701,0200,0000,0002

U 3160, 1342,2113,0603,4174,4007,0521,4000,0000,0000

U 1342, 1400,4221,0013,4170,4007,0700,0000,0000,0000

U 1343, 3161,0111,0706,4174,4007,0700,0000,0000,0000

U 3161, 3156,0111,0704,4170,4007,0700,0200,0004,0712

U 3162, 3163,3770,0303,4344,4007,0700,0000,0000,0000

U 3163, 3164,3771,0003,7270,4007,0701,0000,0000,0214

U 3164, 1100,3440,0303,0174,4007,0700,0400,0000,0000

```
; 5129 ;CLEAR CORE CASE
; 5130 VMA [BRX],
; 5131 START WRITE,
; 5132 PXCT BLT DEST
; 5133 BLTCLR: MEM WRITE, ;STORE WORD
; 5134 MEM Q,
; 5135 SKIP/-1 MS ;1 MS TIMER UP
; 5136 =0 J/BLTGOT ;GO TAKE INTERRUPT
; 5137 [BRX]-[AR], ;BELOW E?
; 5138 3T,
; 5139 SKIP DPO
; 5140 =0 END BLT, ;NO--STOP BLT
; 5141 J/DONE
; 5142 [ARX][ARX]+1, ;FOR PAGE FAIL LOGIC
; 5143 SKIP IRPT
; 5144 =0 VMA [BRX]+1,
; 5145 LOAD VMA,
; 5146 PXCT BLT DEST,
; 5147 START WRITE, ;YES--KEEP STORING
; 5148 J/BLTCLR
; 5149 VMA [BRX]+1, ;INTERRUPT
; 5150 LOAD VMA,
; 5151 PXCT BLT DEST,
; 5152 START WRITE,
; 5153 J/BLTGO
; 5154
; 5155 ;HERE FOR NORMAL BLT
; 5156 BLTLP: MEM READ, ;FETCH
; 5157 Q MEM,
; 5158 J/BLTLP1
; 5159 BLTGO: MEM WRITE, ;STORE
; 5160 MEM Q
; 5161 BLTGOT: [BRX]-[AR], ;BELOW E?
; 5162 3T,
; 5163 SKIP DPO
; 5164 =0 END BLT, ;NO--STOP BLT
; 5165 J/DONE
; 5166 [BRX][BRX]+1 ;UPDATE DEST ADDRESS
; 5167 VMA [ARX]+1,
; 5168 LOAD VMA,
; 5169 PXCT BLT SRC,
; 5170 START READ, ;YES--MOVE 1 MORE WORD
; 5171 J/BLTLP
; 5172
; 5173 ;HERE TO CLEAN UP AFTER BLT PAGE FAILS
; 5174 BLT-CLEANUP:
; 5175 [AR][AR] SWAP ;PUT SRC IN LEFT HALF
; 5176 [AR]_WORK[SV.BRX],
; 5177 HOLD LEFT
; 5178 AC [AR], ;STORE THE AC AND RETURN
; 5179 J/CLEANED
; 5180
```

; T1OKL.MCR[10,1141]
; SIMPLE.MIC[10,1141]

15:34 27-JULY-1984
15:31 27-JULY-1984

MICRO 31(254)

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

```
; 5181 .IF/UBABLT
; 5182 .TOC "UBABLT - BLT BYTES TO/FROM UNIBUS FORMAT"
; 5183
; 5184 ;THESE INSTRUCTION MOVE WORDS FROM BYTE TO UNIBUS AND UNIBUS TO BYTE
; 5185 ;FORMAT. FORMATS ARE:
; 5186 ;
; 5187 ;BYTE FORMAT:
; 5188 ;
; 5189 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5190 ; ;; BYTE 0 ;; BYTE 1 ;; BYTE 2 ;; BYTE 3 ;; 4 BITS ;;
; 5191 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5192 ;
; 5193 ;UNIBUS FORMAT:
; 5194 ;
; 5195 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5196 ; ;; 2 BITS ;; BYTE 1 ;; BYTE 0 ;; 2 BITS ;; BYTE 3 ;; BYTE 2 ;;
; 5197 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
; 5198 ;
; 5199 ;
; 5200 =0*
; 5201 BLTX: [BRX]_AC, ;FETCH THE AC (DEST IN RH)
; 5202 CALL [SETBLT] ;DO THE REST OF THE SETUP
; 5203 AC_[BR] ;STORE THE FINAL AC IN AC
; 5204
; 5205 BLTXLP: MEM READ, ;READ THE SOURCE WORD
; 5206 Q_MEM, ;FROM MEMORY
; 5207 B_DISP ;SKIP IF BLTUB (OPCODE 717)
; 5208 =110 Q_Q*.5, ;BLTBU (OPCODE 716) - SHIFT RIGHT 1

; 5209 J/BLTBU1 ;CONTINUE INSTRUCTION
; 5210
; 5211 AD/D.AND.Q,DBUS/DBM, ;BLTUB - MASK LOW BYTES, SHIFT LEFT
; 5212 DBM/#,#/377,DEST/AD*2,B/T1 ;AND STORE RESULT
; 5213 =00 FE_S#,S#/1767, ;-9 MORE BITS TO PUT LOW BYTE OF LH
; 5214 CALL [T1LSH] ; IN TOP OF LH SHIFT LEFT
; 5215 =01 FE_S#,S#/1772, ;-6 BITS TO PUT HI BYTE TO RIGHT
; 5216 CALL [Q_RSH] ; OF LOW BYTE.
; 5217 =11 Q_Q.AND.#,#/001774 ;KEEP ONLY HI BYTES
; 5218 =
; 5219 AD/A.OR.Q,A/T1,DEST/AD, ;MERGE PAIRS OF BYTES. NOW SWAPPED,
; 5220 B/T1 ;BUT STILL IN HALF-WORDS
; 5221 AD/57,RSRC/OA,A/T1, ;CLEAR LH OF Q WHILE LOADING
; 5222 DEST/Q_AD ;RH WITH LOW WORD
; 5223 Q_Q*2 ;SHIFT LOW WORD ACROSS 1/2 WORD
; 5224 Q_Q*2 ;AND INTO FINAL POSITION
; 5225 [T1]_[T1].AND.# CLR RH, ;CLEAR ALL BUT HIGH 16-BIT WORD
; 5226 #/777774,J/BLTXV ;FROM T1 AND CONTINUE
; 5227
```

U 0670, 3146,3771,0006,0276,6007,0700,0010,0000,0000
U 0672, 3165,3440,0505,0174,4007,0700,0400,0000,0000

U 3165, 0006,3772,0000,4365,5003,7700,0200,0000,0002

IT
U 0006, 3173,3446,1200,4174,4007,0700,0000,0000,0000

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

U 0570, 3200,4443,0000,4174,4007,0700,1010,0071,1767

U 0571, 3201,4443,0000,4174,4007,0700,1010,0071,1772

U 0573, 3166,4662,0000,4374,4007,0700,0000,0000,1774

U 3166, 3167,3001,1717,4174,4007,0700,0000,0000,0000

U 3167, 3170,5742,1700,4174,4007,0700,0000,0000,0000

U 3170, 3171,3444,0012,4174,4007,0700,0000,0000,0000

U 3171, 3172,3444,0012,4174,4007,0700,0000,0000,0000

U 3172, 3177,4521,1717,4374,4007,0700,0000,0077,7774

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; SIMPLE.MIC[10,1141] 15:31 27-JULY-1984

MICRO 31(254)

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

```
U 3173, 3174,3446,1200,4174,4007,0700,0000,0000,0000 ; 5228 BLTBU1: Q_Q*.5 ;NOW IN 1/2 WORDS
U 3174, 3175,3446,1200,4170,4007,0700,0000,0000,0000 ; 5229 Q_Q*.5,HOLD LEFT ;INSERT A NULL BIT IN RH
U 3175, 3176,3446,1200,4170,4007,0700,0000,0000,0000 ; 5230 Q_Q*.5,HOLD LEFT ;ONE MORE - NOW IN HALF WORDS
; 5231 AD/D.AND.Q,DBUS/DBM, ;BUT NOT SWAPPED. COPY RIGHT BYTE
U 3176, 0610,4665,0017,4374,4007,0700,0000,0000,0377 ; 5232 DBM/#,#/377,DEST/AD*2,B/T1 ;TO T1 AND SHIFT LEFT 1 POSI
ION
; 5233 =00 FE_S#,S#/1771, ; -7 BITS MORE
U 0610, 3200,4443,0000,4174,4007,0700,1010,0071,1771 ; 5234 CALL [T1LSH] ;TO FINAL RESTING PLACE
; 5235 =01 FE_S#,S#/1770, ; -8. LEFT BYTES MOVE RIGHT
U 0611, 3201,4443,0000,4174,4007,0700,1010,0071,1770 ; 5236 CALL [Q_RSH] ;TO FINAL RESTING PLACE
U 0613, 3177,4662,0000,4374,4007,0700,0000,0000,0377 ; 5237 =11 Q_Q.AND.#,#/377 ;WANT ONLY THE NEW BYTES
; 5238 =
; 5239
; 5240 BLTXV: Q_[T1].OR.Q, ;MERGE RESULTS
U 3177, 3202,3002,1700,4174,4007,0700,0000,0000,0000 ; 5241 J/BLTXW ;AND STUFF IN MEMORY
; 5242
U 3200, 0001,3445,1717,4174,4004,1700,1020,0041,0001 ; 5243 T1LSH: [T1]_[T1]*2,SHIFT,RETURN [1]
U 3201, 0002,3446,1200,4174,4004,1700,1020,0041,0001 ; 5244 Q_RSH: Q_Q*.5,SHIFT,RETURN [2]
; 5245
; 5246 BLTXW: VMA_[BRX],START WRITE, ;DEST TO VMA
U 3202, 3203,3443,0600,4174,4007,0700,0200,0003,0312 ; 5247 PXCT BLT DEST
U 3203, 3204,3223,0000,4174,4007,0701,0200,0000,0002 ; 5248 MEM WRITE, MEM_Q ;STORE
U 3204, 1344,2113,0603,4174,4007,0521,4000,0000,0000 ; 5249 [BRX]-[AR],3T,SKIP DPO ;DONE?
U 1344, 1400,4221,0013,4170,4007,0700,0000,0000,0000 ; 5250 =0 END BLT,J/DONE ;YES
U 1345, 3205,0111,0706,4174,4007,0700,0000,0000,0000 ; 5251 [BRX]_[BRX]+1 ;NO, INC DEST
; 5252 VMA_[ARX]+1,LOAD VMA, ; AND SOURCE (LOADING VMA)
U 3205, 3165,0111,0704,4170,4007,0700,0200,0004,0712 ; 5253 PXCT BLT SRC,START READ, ;START UP MEMORY
; 5254 J/BLTXLP ;AND CONTINUE WITH NEXT WORD
; 5255 .ENDIF/UBABLT
; 5256
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

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, 1350, 3333, 0005, 4174, 4007, 0520, 0000, 0000, 0000
U 0676, 3206, 3441, 0304, 4174, 4007, 0700, 0000, 0000, 0000
U 3206, 3207, 3441, 0503, 4174, 4007, 0700, 2000, 0041, 2000
U 3207, 3210, 3441, 0405, 4174, 4007, 0700, 2000, 0020, 0000
U 3210, 1346, 3333, 0003, 4174, 4007, 0520, 1000, 0041, 2000
U 1346, 3211, 4551, 0303, 4374, 0007, 0700, 0000, 0000, 0777
U 1347, 3211, 3551, 0303, 4374, 0007, 0700, 0000, 0077, 7000

U 1350, 3211, 4551, 0505, 4374, 0007, 0700, 0000, 0000, 0777
U 1351, 3211, 3551, 0505, 4374, 0007, 0700, 0000, 0077, 7000

U 3211, 1352, 4222, 0000, 4174, 4007, 0630, 2000, 0060, 0000

U 1352, 1352, 3446, 0505, 4174, 4047, 0630, 2000, 0060, 0000
U 1353, 0420, 0111, 0503, 4174, 4003, 4701, 0000, 0000, 0000

```
; 5257 .TOC "FLOATING POINT -- FAD, FSB"
; 5258
; 5259 .DCODE
; 5260 140: FL-R, FL-AC, J/FAD
; 5261 142: FL-RW, FL-MEM, J/FAD
; 5262 FL-RW, FL-BOTH, J/FAD
; 5263 FL-R, FL-AC, ROUND, J/FAD
; 5264 FL-I, FL-AC, ROUND, J/FAD
; 5265 FL-RW, FL-MEM, ROUND, J/FAD
; 5266 FL-RW, FL-BOTH, ROUND, J/FAD
; 5267
; 5268 150: FL-R, FL-AC, J/FSB
; 5269 152: FL-RW, FL-MEM, J/FSB
; 5270 FL-RW, FL-BOTH, J/FSB
; 5271 FL-R, FL-AC, ROUND, J/FSB
; 5272 FL-I, FL-AC, ROUND, J/FSB
; 5273 FL-RW, FL-MEM, ROUND, J/FSB
; 5274 FL-RW, FL-BOTH, ROUND, J/FSB
; 5275 .UCODE
; 5276
; 5277 ;BOTH FAD & FSB ARE ENTERED WITH THE MEMORY OPERAND IN AR
; 5278 ; SIGN SMEARED. THE EXPONENT IN BOTH SC AND FE.
; 5279 1576:
; 5280 FSB: [AR]_[AR] ;MAKE MEMOP NEGATIVE
; 5281
; 5282 1577:
; 5283 FAD: [BR]_AC, SC_SC-EXP-1, 3T, SCAD DISP
; 5284 =0*
; 5285 FAS1: READ [BR], SKIP DPO, J/FAS2 ;BR .LE. AR
; 5286 [ARX]_[AR] ;SWAP AR AND BR
; 5287 [AR]_[BR], SC_EXP
; 5288 [BR]_[ARX], SC_SC-FE-1 ;NUMBER OF SHIFT STEPS
; 5289 READ [AR], FE_EXP, 2T, SKIP DPO
; 5290 =0 [AR]_+SIGN, J/FAS3
; 5291 [AR]_-SIGN, J/FAS3
; 5292
; 5293 =0 ;SIGN SMEAR BR AND UNNORMALIZE
; 5294 FAS2: [BR]_+SIGN, J/FAS3
; 5295 [BR]_-SIGN, J/FAS3
; 5296
; 5297 FAS3: Q_0, STEP SC
; 5298 =0
; 5299 FAS4: [BR]_[BR]*.5 LONG, STEP SC, ASHC, J/FAS4
; 5300 [AR]_[AR]+[BR], NORM DISP, J/SNORM
; 5301
```


; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

D 0160, 0701, 1570, 1100
D 0162, 0702, 1570, 1700
D 0163, 0703, 1570, 1700

D 0164, 0711, 1570, 1100
D 0165, 0611, 1570, 0100
D 0166, 0712, 1570, 1700
D 0167, 0713, 1570, 1700

U 1570, 1354, 3771, 0006, 0276, 6007, 0521, 1000, 0040, 2000
U 1354, 3212, 4551, 0606, 4374, 0007, 0700, 0000, 0000, 0777
U 1355, 3212, 3551, 0606, 4374, 0007, 0700, 0000, 0077, 7000
U 3212, 0163, 3442, 0300, 4174, 4007, 0700, 2000, 0071, 0033

U 0163, 2773, 3446, 0606, 4174, 4007, 0700, 0010, 0000, 0000

U 0167, 3213, 4662, 0000, 4370, 4007, 0700, 0000, 0077, 7000
U 3213, 3214, 3441, 0403, 4174, 4007, 0700, 1000, 0041, 0002

U 3214, 0420, 3446, 0303, 4174, 4003, 4701, 1000, 0041, 1600

```
; 5302 .TOC "FLAOTING POINT -- FMP"
; 5303
; 5304 .DCODE
; 5305 160: FL-R, FL-AC, J/FMP
; 5306 162: FL-RW, FL-MEM, J/FMP
; 5307 FL-RW, FL-BOTH, J/FMP
; 5308
; 5309 FL-R, FL-AC, ROUND, J/FMP
; 5310 FL-I, FL-AC, ROUND, J/FMP
; 5311 FL-RW, FL-MEM, ROUND, J/FMP
; 5312 FL-RW, FL-BOTH, ROUND, J/FMP
; 5313 .UCODE
; 5314
; 5315 1570:
; 5316 FMP: [BRX]_AC, ;GET AC
; 5317 FE_SC+EXP, 3T, ;EXPONENT OF ANSWER
; 5318 SKIP DPO ;GET READY TO SMEAR SIGN
; 5319 =0 [BRX] +SIGN, J/FMP1 ;POSITIVE
; 5320 [BRX] -SIGN, J/FMP1 ;NEGATIVE
; 5321 FMP1: Q [AR], SC 27. ;GET MEMORY OPERAND
; 5322 =01* [BRX] [BRX]*.5 LONG, ;SHIFT RIGHT
; 5323 CALL [MULSUB] ;MULTIPLY
; 5324 Q_Q.AND.#, #/777000, ;WE ONLY COMPUTED
; 5325 HOLD LEFT ; 27 BITS
; 5326 [AR]_[ARX], FE_FE+2 ;SET SHIFT PATHS
; 5327 [AR]_[AR]*.5 LONG, ;SHIFT OVER
; 5328 FE_FE-200, ;ADJUST EXPONENT
; 5329 NORM DISP, J/SNORM ;NORMALIZE & EXIT
; 5330
```

Produced on Advanced Information Services Electronic Laser Printer. PKC/JES. DTN: 223-7881

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO.31(254)

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

D 0170, 0701, 1574, 1100
D 0172, 0702, 1574, 1700
D 0173, 0703, 1574, 1700

D 0174, 0711, 1574, 1100
D 0175, 0611, 1574, 0100
D 0176, 0712, 1574, 1700
D 0177, 0713, 1574, 1700

U 1574, 1356, 3441, 0305, 0174, 4007, 0621, 0000, 0000, 0000

U 1356, 1360, 3771, 0003, 0276, 6007, 0520, 1000, 0030, 2000
U 1357, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0071, 1000

U 1360, 3215, 4551, 0303, 4374, 0007, 0700, 0000, 0000, 0777
U 1361, 3216, 3551, 0303, 4374, 0007, 0700, 0000, 0077, 7000
U 3215, 3217, 3441, 0304, 4174, 4007, 0700, 1000, 0031, 0200
U 3216, 3217, 2441, 0304, 4174, 4007, 0700, 5000, 0031, 0200
U 3217, 1362, 3445, 0506, 4174, 4007, 0520, 0000, 0000, 0000

U 1362, 1364, 2113, 0406, 4174, 4007, 0311, 4000, 0000, 0000
U 1363, 1362, 2445, 0506, 4174, 4007, 0700, 4000, 0000, 0000

U 1364, 1366, 3447, 0606, 4174, 4007, 0700, 0000, 0000, 0000
U 1365, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0071, 1000

U 1366, 3700, 3445, 0303, 4174, 4007, 0700, 0010, 0000, 0000
U 1367, 0144, 2113, 0604, 4174, 4007, 0421, 4000, 0000, 0000

U 0144, 3036, 4222, 0000, 4174, 4007, 0700, 2010, 0071, 0033
U 0145, 0144, 3447, 0303, 4174, 4007, 0700, 1000, 0041, 0001

U 0154, 3220, 3227, 0003, 4174, 4007, 0700, 0000, 0000, 0000

U 0155, 2070, 3333, 0003, 4174, 4007, 0621, 0010, 0000, 0000

U 0156, 2070, 3333, 0003, 4174, 4007, 0621, 0010, 0000, 0000
U 0157, 3220, 3227, 0003, 4174, 4007, 0700, 0000, 0000, 0000
U 0177, 3220, 2227, 0003, 4174, 4007, 0700, 4000, 0000, 0000

U 3220, 1377, 4222, 0000, 4174, 4007, 0700, 0000, 0000, 0000

```
; 5331 .TOC "FLOATING POINT -- FDV"
; 5332
; 5333 .DCODE
; 5334 170: FL-R, FL-AC, J/FDV
; 5335 172: FL-RW, FL-MEM, J/FDV
; 5336 FL-RW, FL-BOTH, J/FDV
; 5337
; 5338 FL-R, FL-AC, ROUND, J/FDV
; 5339 FL-I, FL-AC, ROUND, J/FDV
; 5340 FL-RW, FL-MEM, ROUND, J/FDV
; 5341 FL-RW, FL-BOTH, ROUND, J/FDV
; 5342 .UCODE
; 5343
; 5344
; 5345 1574:
; 5346 FDV: [BR]_[AR], SKIP AD.EQ.O, AC ;COPY DIVSOR SEE IF 0
; 5347 =0
; 5348 [AR]_AC, FE_SC-EXP, SKIP DPO, ;GET AC & COMPUTE NEW
; 5349 J/FDVO ; EXPONENT
; 5350 FL NO DIVIDE ;DIVIDE BY ZERO
; 5351 =0
; 5352 FDVO: [AR]_+SIGN, J/FDV1
; 5353 [AR]_-SIGN, J/FDV2
; 5354 FDV1: [ARX]_[AR], FE -FE+200, J/FDV3 ;COMPUTE 2*DVND
; 5355 FDV2: [ARX]_-[AR], FE -FE+200, J/FDV3 ;ABSOLUTE VALUE
; 5356 FDV3: [BRX]_[BR]*2, SKIP DPO ;ABSOLUTE VALUE
; 5357 =0
; 5358 FDV4: [ARX]_[BRX], SKIP CRYO, 3T, J/FDV5 ;FLOATING NO DIV?
; 5359 [BRX]_-[BR]*2, J/FDV4 ;FORCE ABSOLUTE VALUE
; 5360 =0
; 5361 FDV5: [BRX]_[BRX]*.5, J/FDV6 ;SHIFT BACK ARX
; 5362 FL NO DIVIDE ;UNNORMALIZED INPUT
; 5363 =0
; 5364 FDV6: [AR]_[AR]*2, ;DO NOT DROP A BIT
; 5365 CALL [SBRL] ;AT FDV7+1
; 5366 [BRX]_[ARX], SKIP AD.LE.O ;IS ANSWER .LE. 1?
; 5367 =00100
; 5368 FDV7: Q_O, SC 27., CALL [DIVSGN] ;DIVIDE
; 5369 =00101 [AR]_[AR]*.5, FE_FE+1, J/FDV7 ;SCALE DV'END
; 5370 =01100
; 5371 FDV8: [AR]_Q*.5, J/FDV9 ;PUT ANSWER IN AR
; 5372 =01101 READ [AR], SKIP AD.EQ.O, ;-VE ANSWER, LOOK AT RMDR
; 5373 CALL [SETSN] ; SEE HOW TO NEGATE
; 5374 =01110 READ [AR], SKIP AD.EQ.O, ;-VE ANSWER, LOOK AT RMDR
; 5375 CALL [SETSN] ; SEE HOW TO NEGATE
; 5376 =01111 [AR]_Q*.5, J/FDV9 ;PUT ANSWER IN AR
; 5377 =11111 [AR]_-Q*.5, J/FDV9 ;ZERO RMDR
; 5378
; 5379 FDV9: Q_O, J/SNORMO ;GO NORMALIZE
; 5380
```

; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
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, 1370,4553,0300,4374,4007,0321,0000,0077,7000
U 1370, 1372,2441,0305,4174,4007,0521,4000,0000,0000
U 1371, 1377,4222,0000,4174,4007,0700,1000,0071,0233

U 1372, 1374,4553,0500,4374,4007,0321,0000,0077,7000
U 1373, 3221,4222,0000,4174,4007,0700,1000,0071,0244

U 1374, 3221,4222,0000,4174,4007,0700,1000,0071,0244
U 1375, 1377,4222,0000,4174,4007,0700,1000,0071,0233

U 3221, 1376,3446,0303,4174,4047,0700,2000,0071,0006

U 1376, 1376,3446,0303,4174,4047,0630,2000,0060,0000
U 1377, 0420,3333,0003,4174,4003,4701,0000,0000,0000

U 1621, 3222,3333,0003,4174,4007,0700,2000,0041,4000
U 3222, 3223,4222,0000,0174,4007,0700,0000,0000,0000
U 3223, 2000,3771,0003,0276,6007,0520,1000,0040,2000
U 2000, 1377,4551,0303,4374,0007,0700,0000,0000,0777
U 2001, 1377,3551,0303,4374,0007,0700,0000,0077,7000

```
; 5381 .TOC "FLOATING POINT -- FLTR, FSC"
; 5382
; 5383 .DCODE
; 5384 127: R, FL-AC,ROUND, J/FLTR
; 5385 132: I, FL-AC, J/FSC
; 5386 .UCODE
; 5387
; 5388 1616:
; 5389 FLTR: [AR].AND.#, #/777000, 3T, SKIP ADL.EQ.O ;SMALL POS NUMBER?
; 5390 =0 [BR]_[AR], SKIP DPO, 3T, J/FLTR1 ;NO--SEE IF MINUS
; 5391 Q_O, FE_S#, S#/233, J/SNORMO ;FITS IN 27 BITS
; 5392 =0
; 5393 FLTR1: [BR].AND.#, #/777000, 3T,
; 5394 SKIP ADL.EQ.O, J/FLTR1A ;SMALL NEGATIVE NUMBER
; 5395 Q_O, FE_S#, S#/244, J/FLTR2 ;LARGE POS NUMBER
; 5396 =0
; 5397 FLTR1A: Q_O, FE_S#, S#/244, J/FLTR2 ;BIG NUMBER
; 5398 Q_O, FE_S#, S#/233, J/SNORMO ;FITS IN 27 BITS
; 5399 ;AT THIS POINT WE KNOW THE NUMBER TAKES MORE THAN 27 BITS. WE JUST
; 5400 ; SHIFT 8 PLACES RIGHT AND NORMALIZE. WE COULD BE MORE CLEVER BUT
; 5401 ; THIS IS THE RARE CASE ANYWAY.
; 5402 FLTR2: [AR]_[AR]*.5 LONG, ASHC, SC_6 ;SHOVE OVER TO THE RIGHT
; 5403 =0
; 5404 FLTR3: [AR]_[AR]*.5 LONG, ASHC, ;SHIFT RIGHT 9 PLACES
; 5405 STEP SC, J/FLTR3 ; SO IT WILL FIT
; 5406 SNORMO: READ [AR], NORM DISP, J/SNORM ;NORMALIZE ANSWER
; 5407
; 5408
; 5409 1621:
; 5410 FSC: READ [AR], SC_SHIFT
; 5411 Q_O, AC ;DON'T SHIFT IN JUNK
; 5412 [AR]_AC, FE_SC+EXP, SKIP DPO ;SIGN SMEAR
; 5413 =0 [AR]_+SIGN, J/SNORMO
; 5414 [AR]_-SIGN, J/SNORMO
; 5415
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

D 0122, 0701, 1626, 1100
D 0126, 0711, 1626, 1100

U 1626, 0720, 4222, 0000, 4174, 4006, 7701, 0000, 0041, 1534
U 0720, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0041, 1000
U 0722, 0724, 4443, 0000, 4174, 4006, 7701, 2000, 0041, 1544
U 0724, 2004, 4443, 0000, 4174, 4007, 0630, 2000, 0060, 0000
U 0726, 3224, 4443, 0000, 4174, 4007, 0700, 2000, 0031, 0232

U 3224, 2002, 4443, 0000, 4174, 4007, 0630, 2000, 0060, 0000

U 2002, 2002, 3446, 0303, 4174, 4047, 0630, 2000, 0060, 0000
U 2003, 0063, 3447, 0705, 4174, 4003, 7700, 0000, 0000, 0000

U 2004, 2004, 3445, 0303, 4174, 4007, 0630, 2000, 0060, 0000
U 2005, 0100, 3440, 0303, 0174, 4156, 4700, 0400, 0000, 0000

U 0063, 2006, 3333, 0003, 4174, 4007, 0520, 0000, 0000, 0000
U 0073, 1514, 0111, 0503, 4174, 4003, 7700, 0200, 0003, 0001

U 2006, 0100, 3440, 0303, 0174, 4156, 4700, 0400, 0000, 0000
U 2007, 2010, 3223, 0000, 4174, 4007, 0621, 0000, 0000, 0000
U 2010, 1514, 0111, 0703, 4174, 4003, 7700, 0200, 0003, 0001

U 2011, 0073, 7441, 1205, 4174, 4007, 0700, 0000, 0000, 0000

```
; 5416 .TOC "FLOATING POINT -- FIX AND FIXR"
; 5417
; 5418 .DCODE
; 5419 122: FL-R, FL-AC, J/FIX
; 5420 126: FL-R, FL-AC,ROUND, J/FIX
; 5421 .UCODE
; 5422
; 5423 1626:
; 5424 FIX: Q_O, SCAD/A+B, SCADA/S#, ;CLEAR Q, SEE IF
; 5425 S#/1534, SCADB/FE, 3T, ; ANSWER FITS IN
; 5426 SCAD DISP ; 35 BITS.
; 5427 =0* SET AROV, J/NIDISP ;TOO BIG
; 5428 SC_FE+S#, S#/1544, 3T, SCAD DISP ;NEED TO MOVE LEFT?
; 5429 =0* STEP SC, J/FIXL
; 5430 SC_S#-FE, S#/232 ;NUMBER OF PLACES TO SHIFT
; 5431 ; RIGHT
; 5432 STEP SC ;ALREADY THERE
; 5433 =0
; 5434 FIXR: [AR]_[AR]*.5 LONG, ASHC, ;SHIFT BINARY POINT
; 5435 STEP SC, J/FIXR ; TO BIT 35.5
; 5436 [BR]_[ONE]*.5, B DISP, J/FIXX ;WHICH KIND OF FIX?
; 5437
; 5438 =0
; 5439 FIXL: [AR]_[AR]*2, STEP SC, J/FIXL ;SHIFT LEFT
; 5440 AC_[AR], NEXT INST ;WE ARE NOW DONE
; 5441
; 5442 =0*11
; 5443 FIXX: READ [AR], SKIP DPO, J/FIXT ;FIX--SEE IF MINUS
; 5444 FIXX1: [AR]_[AR]+[BR], FL-EXIT ;FIXR--ROUND UP
; 5445 =0
; 5446 FIXT: AC_[AR], NEXT INST ;FIX & +, TRUNCATE
; 5447 READ Q, SKIP AD.EQ.O ;NEGATIVE--ANY FRACTION?
; 5448 =0 [AR]_[AR]+1, FL-EXIT ;YES--ROUND UP
; 5449 [BR]_.NOT.[MASK], ;MAYBE--GENERATE .75
; 5450 J/FIXX1 ;ROUND UP IF BIT 36 OR
; 5451 ; 37 SET
; 5452
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

```
; 5453 .TOC "FLOATING POINT -- SINGLE PRECISION NORMALIZE"
; 5454
; 5455 ;NORMALIZE DISPATCH IS A 9-WAY DISPATCH. THE HARDWARE LOOKS AT
; 5456 ; 4 SIGNALS: DP=0, DP BIT 8, DP BIT 9, DP BIT -2. THE 9 CASES
; 5457 ; ARE:
; 5458
; 5459 ; DP=0 DPO8 DPO9 DPO0 ACTION TO TAKE
; 5460 ; 0 0 0 0 SHIFT LEFT
; 5461 ;
; 5462 ; 0 0 0 1 NEGATE AND RETRY
; 5463 ;
; 5464 ; 0 0 1 0 ALL DONE
; 5465 ;
; 5466 ; 0 0 1 1 NEGATE AND RETRY
; 5467 ;
; 5468 ; 0 1 0 0 SHIFT RIGHT
; 5469 ;
; 5470 ; 0 1 0 1 NEGATE AND RETRY
; 5471 ;
; 5472 ; 0 1 1 0 SHIFT RIGHT
; 5473 ;
; 5474 ; 0 1 1 1 NEGATE AND RETRY
; 5475 ;
; 5476 ; 1 - - - LOOK AT Q BITS
; 5477
; 5478 ;ENTER HERE WITH UNNORMALIZED NUMBER IN AR!Q. FE HOLDS THE NEW
; 5479 ; EXPONENT. CALL WITH NORM DISP
; 5480 =0000 ;9-WAY DISPATCH
; 5481 SNORM: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNORM
; 5482 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5483 READ [AR], NORM DISP, CALL [SROUND]
; 5484 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5485 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
; 5486 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5487 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
; 5488 Q_-Q, SKIP CRYO, 3T, J/SNNEG
; 5489 READ Q, SKIP AD.EQ.O, J/SNORM1
; 5490 =1110 [AR]_EXP, J/FLEX
; 5491 =
; 5492 =0
; 5493 SNORM1: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNORM
; 5494 FLEX: FL-EXIT
; 5495
```

```
U 0420, 0420,3444,0303,4174,4063,4701,1000,0041,1777
U 0421, 2014,2222,0000,4174,4007,0311,4000,0000,0000
U 0422, 0262,3333,0003,4174,4003,4701,0010,0000,0000
U 0423, 2014,2222,0000,4174,4007,0311,4000,0000,0000
U 0424, 0262,3447,0303,4174,4007,0700,1010,0041,0001
U 0425, 2014,2222,0000,4174,4007,0311,4000,0000,0000
U 0426, 0262,3447,0303,4174,4007,0700,1010,0041,0001
U 0427, 2014,2222,0000,4174,4007,0311,4000,0000,0000
U 0430, 2012,3223,0000,4174,4007,0621,0000,0000,0000
U 0436, 2013,3770,0303,4324,0457,0700,0000,0041,0000

U 2012, 0420,3444,0303,4174,4063,4701,1000,0041,1777
U 2013, 1514,4443,0000,4174,4003,7700,0200,0003,0001
```

; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 149
FLOATING POINT -- SINGLE PRECISION NORMALIZE

```
U 2014, 0440,7441,0303,4174,4003,4701,0000,0000,0000 ; 5496 =0
; 5497 SNNEG: [AR]_.NOT.[AR], NORM DISP, J/SNNORM ;NEGATE HIGH WORD
; 5498 ; (NO CARRY)
U 2015, 0440,2441,0303,4174,4003,4701,4000,0000,0000 ; 5499 [AR]_[AR], NORM DISP, J/SNNORM ;NEGATE HIGH WORD (W/CARRY)
; 5500 =0000
U 0440, 0440,3444,0303,4174,4063,4701,1000,0041,1777 ; 5501 SNNORM: [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNNORM
U 0442, 0262,3333,0003,4174,4003,4701,0010,0000,0000 ; 5502 =0010 READ [AR], NORM DISP, CALL [SROUND]
U 0444, 0262,3447,0303,4174,4007,0700,1010,0041,0001 ; 5503 =0100 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
U 0446, 0262,3447,0303,4174,4007,0700,1010,0041,0001 ; 5504 =0110 [AR]_[AR]*.5, FE_FE+1, CALL [SROUND]
U 0450, 0440,3444,0303,4174,4063,4701,1000,0041,1777 ; 5505 =1000 [AR]_[AR]*2 LONG, DIV, FE_FE-1, NORM DISP, J/SNNORM
U 0456, 0327,3770,0303,4324,0453,7700,0000,0041,0000 ; 5506 =1110 [AR]_EXP, B DISP
; 5507 =
U 0327, 2016,4553,1300,4374,4007,0321,0000,0000,2000 ; 5508 =0111 TL [FLG], FLG.SN/1, J/SNNOT
; 5509 [AR]_[AR].AND.[MASK], ;CLEAR ANY LEFT OVER BITS
U 0337, 2021,4111,1203,4174,4007,0700,0000,0000,0000 ; 5510 J/SNNOT1
; 5511 =0
U 2016, 3225,7441,0303,4174,4007,0700,0000,0000,0000 ; 5512 SNNOT: [AR]_.NOT.[AR], J/SNNOT2
U 2017, 2020,3223,0000,4174,4007,0621,0000,0000,0000 ; 5513 READ Q, SKIP AD.EQ.0
U 2020, 3225,7441,0303,4174,4007,0700,0000,0000,0000 ; 5514 =0 [AR]_.NOT.[AR], J/SNNOT2
U 2021, 3225,2441,0303,4174,4007,0700,4000,0000,0000 ; 5515 SNNOT1: [AR]_[AR], J/SNNOT2 ;NORMAL NEGATE AND EXIT
U 3225, 1514,4221,0013,4174,4003,7700,0200,0003,0001 ; 5516 SNNOT2: [FLG]_O, FL-EXIT
; 5517
; 5518
; 5519
; 5520 .TOC "FLOATING POINT -- ROUND ANSWER"
; 5521
; 5522 =*01*
U 0262, 0407,3447,0705,4174,4003,7700,0000,0000,0000 ; 5523 SROUND: [BR]_[ONE]*.5, B DISP, J/SRND1
U 0266, 0262,3447,0303,4174,4007,0700,1000,0041,0001 ; 5524 [AR]_[AR]*.5, FE_FE+1, J/SROUND ;WE WENT TOO FAR
; 5525 =0111
U 0407, 0016,4443,0000,4174,4004,1700,0000,0000,0000 ; 5526 SRND1: RETURN [16] ;NOT ROUNDING INSTRUCTION
U 0417, 0302,0111,0503,4174,4003,4701,0000,0000,0000 ; 5527 [AR]_[AR]+[BR], NORM DISP
U 0302, 0016,4443,0000,4174,4004,1700,0000,0000,0000 ; 5528 =*01* RETURN [16]
U 0306, 0016,3447,0303,4174,4004,1700,1000,0041,0001 ; 5529 [AR]_[AR]*.5, FE_FE+1, RETURN [16]
; 5530
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-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, 3226, 2441, 0404, 4174, 4007, 0700, 4000, 0000, 0000
U 3226, 1637, 2441, 0303, 4174, 4007, 0700, 0040, 0000, 0000
U 1637, 3227, 4557, 0006, 1274, 4007, 0701, 0000, 0000, 1441
U 3227, 2022, 3777, 0005, 0274, 4007, 0521, 2000, 0020, 2000
U 2022, 2024, 5547, 0505, 0374, 4007, 0631, 0000, 0077, 7400
U 2023, 2024, 3547, 0505, 0374, 4007, 0631, 0000, 0077, 7400
U 2024, 2026, 3442, 0600, 4174, 4007, 0700, 0000, 0000, 0000
U 2025, 3231, 3771, 0016, 0276, 6007, 0700, 2000, 0041, 2000

```
; 5531 .TOC "FLOATING POINT -- DFAD, DFSB"
; 5532
; 5533 .DCODE
; 5534 110: DBL FL-R, J/DFAD
; 5535 111: DBL FL-R, J/DFSB
; 5536 .UCODE
; 5537
; 5538 ;ENTER FROM A-READ CODE WITH:
; 5539 ;FE/ EXP
; 5540 ;SC/ EXP
; 5541 ;AR/ C(E) SHIFT RIGHT 2 PLACES
; 5542 ;ARX/ C(E+1) SHIFTED RIGHT 1 PLACE
; 5543 1635:
; 5544 DFSB: [ARX]_[ARX] ;NEGATE LOW WORD
; 5545 [AR]_[AR]-.25, MULTI PREC/1
; 5546 1637:
; 5547 DFAD: [BRX]_(AC[1].AND.[MAG])* .5, 3T ;GET LOW WORD
; 5548 [BR]_AC*.5, 3T, ;GET AC AND START TO SHIFT
; 5549 SC_SC-EXP-1, ;NUMBER OF PLACES TO SHIFT
; 5550 SKIP DPO ;SEE WHAT SIGN
; 5551 =0 [BR]_+SIGN*.5, 3T, ;SIGN SMEAR
; 5552 AC, SKIP/SC, J/DFAS1 ;SEE WHICH IS BIGGER
; 5553 [BR]_-SIGN*.5, 3T, ;SIGN SMEAR
; 5554 AC, SKIP/SC, J/DFAS1 ;SEE WHICH IS BIGGER
; 5555 =0
; 5556 DFAS1: Q [BRX], ;AR IS BIGGER
; 5557 J/DFAS2 ;ADJUST BR!Q
; 5558 [TO]_AC, ;BR IS BIGGER OR EQUAL
; 5559 SC_EXP, 2T, J/DFAS3 ;SET SC TO THAT EXPONENT
; 5560
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

U 2026, 0153,3441,0516,4174,4007,0700,0010,0000,0000
U 2027, 3230,3441,1605,4174,4007,0700,0000,0000,0000
U 3230, 3234,0002,0400,4174,4007,0700,0000,0000,0000

U 3231, 3232,3442,0400,4174,4007,0700,2000,0020,0000
U 3232, 2030,3333,0016,4174,4007,0700,1000,0041,2000
U 2030, 0153,3441,0316,4174,4007,0700,0010,0000,0000
U 2031, 3233,3441,1603,4174,4007,0700,0000,0000,0000
U 3233, 3234,0002,0600,4174,4007,0700,0000,0000,0000

U 3234, 3235,0116,0503,4174,4047,0700,0040,0000,0000

U 3235, 0433,3444,0303,4174,4046,2700,0000,0000,0000

U 0433, 3236,5111,1217,4174,4007,0700,0000,0000,0000
U 0437, 0433,0222,0000,4174,4007,0700,4000,0000,0000

U 3236, 3237,3444,0303,4174,4047,0700,1000,0041,1777

U 3237, 3240,3444,0303,4174,4047,0700,1000,0041,1777
U 3240, 2043,3002,1700,4170,4007,0700,0000,0000,0000

```
; 5561 ;HERE IF AR!ARX IS GREATER THAN BR!BRX
; 5562 =0
; 5563 DFAS2: [TO]_[BR], CALL [DFADJ] ;ADJUST BR!Q
; 5564 [BR]_[TO] ;PUT ANSWER BACK
; 5565 Q_Q+[ARX], J/DFAS5 ;ADD LOW WORDS
; 5566
; 5567 ;HERE IS BR!BRX IF GREATER THAN OR EQUAL TO AR!ARX
; 5568 DFAS3: Q_[ARX], ;SETUP TO SHIFT AR!ARX
; 5569 SC_SC-FE-1 ;COMPUTE # OF PLACES
; 5570 READ [TO], FE_EXP ;EXPONENT OF ANSWER
; 5571 =0 [TO]_[AR], CALL [DFADJ] ;ADJUST AR!Q
; 5572 [AR]_[TO] ;PUT ANSWER BACK
; 5573 Q_Q+[BRX], J/DFAS5 ;ADD LOW WORDS
; 5574
; 5575 ;BIT DIDDLE TO GET THE ANSWER (INCLUDING 2 GUARD BITS) INTO
; 5576 ; AR!Q
; 5577 DFAS5: [AR]_([AR]+[BR])*5 LONG, ;ADD HIGH WORDS
; 5578 MULTI PREC/1, ASHC ;INJECT SAVED CRY2
; 5579 [AR]_[AR]*2 LONG, ;SHIFT BACK LEFT
; 5580 ASHC, MUL DISP ;SEE IF WE LOST A 1
; 5581 =1011
; 5582 DFAS6: [T1]_[T1].AND.NOT.[MASK], J/DFAS7
; 5583 Q_Q+.25, J/DFAS6
; 5584 DFAS7: [AR]_[AR]*2 LONG, ASHC, ;PUT IN GUARD BITS
; 5585 FE_FE-1
; 5586 [AR]_[AR]*2 LONG, ASHC,
; 5587 FE_FE-1
; 5588 Q_[T1].OR.Q, HOLD LEFT, J/DNORMO
; 5589
```


; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

U 0153, 2071,3444,1616,4174,4067,0700,0010,0000,0000
U 0173, 3241,3444,1616,4174,4067,0700,0000,0000,0000
U 3241, 3242,3444,1616,4174,4067,0700,0000,0000,0000

U 3242, 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, 2032,3551,1313,4374,0007,0700,0000,0000,2000
U 0477, 2033,3551,1313,4374,0007,0700,0000,0000,2000

U 2032, 2032,3446,1616,4174,4047,0630,2000,0060,0000
U 2033, 0453,3221,0017,4174,4007,0700,0000,0000,0000

U 0453, 3243,3446,1616,4174,4047,0700,0000,0000,0000
U 0457, 0453,3551,1313,4374,0007,0700,0000,0000,2000
U 3243, 0001,3446,1616,4174,4044,1700,0000,0000,0000

```
; 5590 ;SUBROUTINE TO ADJUST NUMBER IN TO!Q
; 5591 ;RETURNS 1 WITH
; 5592 ; TO!Q ADJUSTED
; 5593 ; FLG.SN=1 IF WE SHIFTED OUT ANY 1 BITS (STICKY BIT)
; 5594 ; T1 HAS Q TWO STEPS PRIOR TO BEING DONE
; 5595 DFADJ "STEP SC, ASHC, MUL DISP"
; 5596
; 5597 =0**11
; 5598 DFADJ: [TO]_[TO]*2 LONG, DIV, ;MOVE EVERYTHING 2 PLACES
; 5599 CALL [CLRSN]
; 5600 [TO]_[TO]*2 LONG, DIV
; 5601 [TO]_[TO]*2 LONG, DIV
; 5602 [TO]_[TO]*.5 LONG, ASHC, ;SHIFT AT LEAST 1 PLACE
; 5603 STEP SC
; 5604 =1010
; 5605 DFADJ1: [TO]_[TO]*.5 LONG, ;UNNORMALIZE TO!Q
; 5606 DFADJ, J/DFADJ1 ;LOOP TILL DONE
; 5607 DFADJ2: [T1]_Q, ;SAVE GUARD BITS
; 5608 MUL DISP, J/DFADJ5 ;LOOK AT LAST BIT
; 5609 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ3
; 5610 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ4
; 5611
; 5612 =0
; 5613 DFADJ3: [TO]_[TO]*.5 LONG, ASHC, STEP SC, J/DFADJ3
; 5614 DFADJ4: [T1]_Q ;SAVE 2 GUARD BITS
; 5615 =1011
; 5616 DFADJ5: [TO]_[TO]*.5 LONG, ASHC, J/DFADJ6
; 5617 [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, J/DFADJ5
; 5618 DFADJ6: [TO]_[TO]*.5 LONG, ASHC, RETURN [1]
; 5619
```

; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
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, 2034, 3442, 0400, 4174, 4007, 0700, 2000, 0071, 0006

U 2034, 2034, 3444, 0303, 4174, 4047, 0630, 2000, 0060, 0000
U 2035, 3244, 3446, 1200, 4174, 4007, 0700, 0000, 0000, 0000
U 3244, 3245, 4662, 0000, 4374, 0007, 0700, 0000, 0007, 7777
U 3245, 3246, 3221, 0005, 4174, 4007, 0700, 0000, 0000, 0000

U 3246, 0623, 4557, 0006, 1274, 4007, 0700, 0000, 0000, 1441
U 0623, 2774, 3447, 0606, 4174, 4007, 0700, 2010, 0071, 0043

U 0627, 0730, 3442, 0300, 4174, 4007, 0700, 2000, 0071, 0043
U 0730, 2775, 4443, 0000, 4174, 4007, 0700, 0010, 0000, 0000
U 0734, 3247, 3441, 0416, 4174, 4007, 0700, 0000, 0000, 0000
U 3247, 3250, 3227, 0004, 4174, 4007, 0700, 2000, 0011, 0000

U 3250, 2036, 3777, 0006, 0274, 4007, 0521, 1000, 0040, 2000
U 2036, 0732, 5547, 0606, 4374, 4007, 0701, 0000, 0077, 7400
U 2037, 0732, 3547, 0606, 4374, 4007, 0701, 0000, 0077, 7400

U 0732, 2775, 3442, 0500, 4174, 4007, 0700, 2010, 0071, 0043
U 0736, 3251, 3221, 0017, 4174, 4007, 0700, 0000, 0000, 0000
U 3251, 1012, 0111, 1604, 4174, 4007, 0700, 0000, 0000, 0000

U 1012, 2775, 3442, 0300, 4174, 4007, 0700, 2010, 0071, 0043

```
; 5620 .TOC "FLOATING POINT -- DFMP"
; 5621
; 5622 .DCODE
; 5623 112: DBL FL-R, DAC, J/DFMP
; 5624 .UCODE
; 5625
; 5626 ;SAME ENTRY CONDITIONS AS DFAD/DFSB
; 5627 1631:
; 5628 DFMP: Q_[ARX], SC_6 ;SHIFT MEM OP 8 PLACES
; 5629 =0
; 5630 DFMP1: [AR]_[AR]*2 LONG, ASHC, ;SHIFT
; 5631 STEP SC, J/DFMP1
; 5632 Q_Q*.5
; 5633 Q_Q.AND.#, #/077777, HOLD RIGHT
; 5634 [BR]_Q ;COPY LOW WORD
; 5635 ;
; 5636 ; BRX * BR ==> C(E+1) * C(AC+1)
; 5637 ;
; 5638 [BRX]_(AC[1].AND.[MAG])*0.5 ;GET LOW AC
; 5639 =0** [BRX]_[BRX]*0.5, SC_35., CALL [MULSB1]
; 5640 ;
; 5641 ; BRX * Q ==> C(E) * C(AC+1)
; 5642 ;
; 5643 Q_[AR], SC_35. ;GO MULT NEXT HUNK
; 5644 =0** CALL [MULTIPLY]
; 5645 [TO]_[ARX] ;SAVE PRODUCT
; 5646 [ARX]_Q*.5, SC_FE ;PUT IN NEXT STEP
; 5647 ;
; 5648 ; BRX * BR ==> C(AC) * C(E+1)
; 5649 ;
; 5650 [BRX]_AC*.5, ;PREPARE TO DO HIGH HALF
; 5651 FE_SC+EXP, ;EXPONENT ON ANSWER
; 5652 SKIP DPO, 3T
; 5653 =0 [BRX]_+SIGN*.5, 3T, J/DFMP2
; 5654 [BRX]_-SIGN*.5, 3T
; 5655 =0**
; 5656 DFMP2: Q_[BR], SC_35., CALL [MULTIPLY] ;GO MULTIPLY
; 5657 [T1]_Q ;SAVE FOR ROUNDING
; 5658 [ARX]_[ARX]+[TO] ;PREPARE FOR LAST MUL
; 5659 ;
; 5660 ; BRX * Q ==> C(AC) * C(E)
; 5661 ;
; 5662 =0** Q_[AR], SC_35., ;DO THE LAST MULTIPLY
; 5663 CALL [MULTIPLY] ;...
; 5664
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

U 1016, 0243, 3446, 0403, 4174, 4007, 0700, 1000, 0041, 1576
U 0243, 2070, 3333, 0017, 4174, 4007, 0621, 0010, 0000, 0000
U 0263, 3252, 3444, 0303, 4174, 4047, 0700, 0000, 0000, 0000
U 3252, 3253, 3447, 0705, 4174, 4007, 0700, 0000, 0000, 0000
U 3253, 2040, 4553, 1700, 4374, 4007, 0321, 0000, 0020, 0000
U 2040, 2041, 0002, 0500, 4174, 4007, 0700, 0000, 0000, 0000
U 2041, 3254, 3444, 0303, 4174, 4047, 0700, 0000, 0000, 0000
U 3254, 2042, 4553, 1700, 4374, 4007, 0321, 0000, 0010, 0000
U 2042, 2043, 0002, 0500, 4174, 4007, 0700, 0000, 0000, 0000
U 2043, 0520, 3333, 0003, 4174, 4003, 4701, 1000, 0041, 0002

```
; 5665 ;OK, WE NOW HAVE THE PRODUCT IN ARX!Q!T1. ALL WE NEED TO DO
; 5666 ; IS SOME BIT DIDDLES TO GET EVERYTHING IN THE RIGHT PLACE
; 5667 [AR]_[ARX]*.5 LONG, ;SHIFT THE ANSWER
; 5668 FE_FE+S#, S#/1576 ;CORRECT EXPONENT
; 5669 =O**11 READ [T1], SKIP AD.EQ.O, ;SEE IF LOW ORDER 1
; 5670 CALL [SETSN] ; BITS AROUND SOMEPLACE
; 5671 [AR]_[AR]*2 LONG, ASHC ;SHIFT LEFT
; 5672 [BR]_[ONE]*.5 ;PLACE TO INSTERT BITS
; 5673 TL [T1], #/200000 ;ANYTHING TO INJECT?
; 5674 =O Q_Q+[BR] ;YES--PUT IT IN
; 5675 [AR]_[AR]*2 LONG, ASHC ;MAKE ROOM FOR MORE
; 5676 TL [T1], #/100000 ;ANOTHER BIT NEEDED
; 5677 =O Q_Q+[BR] ;YES--PUT IN LAST BIT
; 5678 DNORMO: READ [AR], NORM DISP, ;SEE WHAT WE NEED TO DO
; 5679 FE_FE+S#, S#/2, J/DNORM ;ADJUST FOR INITIAL SHIFTS
; 5680
```

; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

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

D 0113, 1105, 1636, 1100

U 1636, 0132, 3441, 0406, 4174, 4007, 0700, 0000, 0000, 0000

U 0132, 2071, 4221, 0017, 4174, 4007, 0700, 0010, 0000, 0000

U 0133, 2044, 3441, 0305, 1174, 4007, 0421, 0000, 0000, 1441

U 2044, 3257, 4557, 0004, 1274, 4007, 0700, 0000, 0000, 1441

U 2045, 3255, 7441, 1717, 4174, 4007, 0700, 0000, 0000, 0000

U 3255, 3256, 2441, 0606, 4174, 4007, 0700, 4000, 0000, 0000

U 3256, 2044, 2331, 0005, 1174, 4007, 0521, 0040, 0000, 1441

U 3257, 2046, 3777, 0003, 0274, 4007, 0521, 1000, 0030, 2000

U 2046, 2050, 5547, 0303, 4374, 4007, 0701, 0000, 0077, 7400

U 2047, 3260, 7441, 1717, 4174, 4007, 0700, 0000, 0000, 0000

U 3260, 3261, 3547, 0303, 4374, 4007, 0701, 0000, 0077, 7400

U 3261, 3262, 2442, 0400, 4174, 4007, 0700, 4000, 0000, 0000

U 3262, 2051, 2446, 0303, 4174, 4047, 0700, 0040, 0000, 0000

U 2050, 3035, 3442, 0400, 4174, 4007, 0700, 0010, 0000, 0000

U 2051, 2052, 2113, 0305, 4174, 4007, 0521, 4000, 0000, 0000

U 2052, 0603, 4443, 0000, 4174, 4467, 0700, 0000, 0071, 1000

U 2053, 1054, 3221, 0004, 4174, 4007, 0700, 0000, 0000, 0000

U 1054, 1276, 4222, 0000, 4174, 4007, 0700, 2010, 0071, 0032

U 1056, 1064, 3221, 0016, 4174, 4007, 0700, 2000, 0071, 0043

U 1064, 1276, 5002, 0000, 4174, 4007, 0621, 0010, 0000, 0000

U 1066, 3263, 3446, 1200, 4174, 4007, 0700, 0000, 0000, 0000

U 3263, 0513, 3444, 1616, 4174, 4046, 2700, 0000, 0000, 0000

U 0513, 2054, 3333, 0017, 4174, 4007, 0520, 1000, 0031, 0202

U 0517, 0513, 0222, 0000, 4174, 4007, 0700, 4000, 0000, 0000

U 2054, 0520, 3441, 1603, 4174, 4003, 4701, 0000, 0000, 0000

U 2055, 0200, 3441, 1603, 4174, 4003, 4701, 0000, 0000, 0000

```
; 5681 .TOC "FLOATING POINT -- DFDV"
; 5682
; 5683 .DCODE
; 5684 113: DBL FL-R, DAC, J/DFDV
; 5685 .UCODE
; 5686 1636:
; 5687 DFDV: [BRX]_[ARX] ;COPY OPERAND (COULD SAVE TIME
; 5688 ; WITH SEPERATE A-READ FOR DFDV)
; 5689 =1**10 [T1]_O, CALL [CLRSN] ;CLEAR FLAG
; 5690 [BR]_[AR], SKIP AD.LE.O, ;SEE IF POSITIVE
; 5691 AC[1] ;WARM UP RAM
; 5692 =0
; 5693 DFDV1: [ARX]_(AC[1].AND.[MAG])* .5, ;POSITIVE--GET AC
; 5694 J/DFDV2 ; AND CONTINUE BELOW
; 5695 [T1]_.NOT.[T1] ;DV'SOR NEGATIVE (OR ZERO)
; 5696 [BRX]_- [BRX] ;NEGATE LOW WORD
; 5697 AD/-B-.25, B/BR, DEST/AD, ;NEGATE HIGH WORD
; 5698 MULTI PREC/1, 3T, ;ADDING IN CRYO2
; 5699 SKIP DPO, AC[1], ;SEE IF STILL NEGATIVE
; 5700 J/DFDV1 ; ..
; 5701 DFDV2: [AR]_AC*.5, ;GET AC AND SHIFT
; 5702 FE_SC-EXP, 3T, ;COMPUTE NEW EXPONENT
; 5703 SKIP DPO ;SEE IF NEGATIVE
; 5704 =0 [AR]_+SIGN*.5, 3T, J/DFDV3 ;POSITIVE
; 5705 [T1]_.NOT.[T1] ;NEGATIVE OR ZERO
; 5706 [AR]_-SIGN*.5, 3T ;SIGN SMEAR
; 5707 Q_- [ARX] ;NEGATE OPERAND
; 5708 [AR]_(-[AR]_.25)*.5 LONG, ;NEGATE HIGH WORD
; 5709 MULTI PREC/1, ;USE SAVED CARRY
; 5710 ASHC, J/DFDV4 ;CONTINUE BELOW
; 5711 =0
; 5712 DFDV3: Q_[ARX], ;COPY OPERAND
; 5713 CALL [DDIVS] ;SHIFT OVER
; 5714 DFDV4: [AR]_- [BR], 3T, SKIP DPO ;SEE IF OVERFLOW
; 5715 =0 FL NO DIVIDE
; 5716 [ARX]_Q ;START DIVISION
; 5717 =0* Q_O, SC_26., CALL [DBLDIV]
; 5718 [TO]_Q, SC_35.
; 5719 =0* Q_Q.AND.NOT.[MAG], ;SEE IF ODD
; 5720 SKIP AD.EQ.O, ;SKIP IF EVEN
; 5721 CALL [DBLDIV] ;GO DIVIDE
; 5722 Q_Q*.5 ;MOVE ANSWER OVER
; 5723 =
; 5724 [TO]_[TO]*2 LONG, ASHC, ;DO FIRST NORM STEP
; 5725 MUL DISP ; SEE IF A 1 FELL OUT
; 5726 =1011
; 5727 DFDV4A: READ [T1], SKIP DPO, ;SHOULD RESULT BE NEGATIVE
; 5728 FE_S#-FE, S#/202, ;CORRECT EXPONENT
; 5729 J/DFDV4B ;LOOK BELOW
; 5730 Q_Q+.25, J/DFDV4A ;PUT BACK THE BIT
; 5731 =0
; 5732 DFDV4B: [AR]_[TO], NORM DISP, J/DNORM ;PLUS
; 5733 [AR]_[TO], NORM DISP, J/DNNORM ;MINUS
; 5734
```

Produced on Advanced Information Services Electronic Laser Printer, PKC/LE6, DTN: 223-7881

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-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, 2060,4553,1300,4374,4007,0321,0000,0000,2000
U 0522, 0312,3333,0003,4174,4003,4701,0010,0000,0000
U 0523, 2060,4553,1300,4374,4007,0321,0000,0000,2000
U 0524, 0312,3446,0303,4174,4047,0700,1010,0041,0001
U 0525, 2060,4553,1300,4374,4007,0321,0000,0000,2000
U 0526, 0312,3446,0303,4174,4047,0700,1010,0041,0001
U 0527, 2060,4553,1300,4374,4007,0321,0000,0000,2000
U 0530, 3265,4002,0000,4174,0007,0700,0000,0000,0000
U 0536, 3264,4221,0013,4174,4007,0700,0000,0000,0000
U 3264, 1515,4113,0400,1174,4007,0700,0400,0000,1441
U 3265, 2056,3223,0000,4174,4007,0621,0000,0000,0000
U 2056, 0520,3444,0303,4174,4043,4701,1000,0041,1777
U 2057, 1515,3440,0303,1174,4007,0700,0400,0000,1441

```
; 5735 .TOC "FLOATING POINT -- DOUBLE PRECISION NORMALIZE"  
; 5736  
; 5737 ;NORMALIZE AR!Q  
; 5738 ;DNORMO: READ [AR], NORM DISP, ;SEE WHAT WE NEED TO DO  
; 5739 ; FE_FE+S#, S#/2, J/DNORM ;ADJUST FOR INITIAL SHIFTS  
; 5740 =O000  
; 5741 DNORM: [AR] [AR]*2 LONG, ;SHIFT LEFT  
; 5742 FE_FE-1, ASHC, ;ADJUST EXPONENT  
; 5743 NORM DISP, J/DNORM ;TRY AGAIN  
; 5744 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5745 READ [AR], NORM DISP, ;SEE IF WE WENT TOO FAR  
; 5746 CALL [DROUND] ; AND ROUND ANSWER  
; 5747 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5748 [AR] [AR]*.5 LONG, ASHC,  
; 5749 FE_FE+1, CALL [DROUND]  
; 5750 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5751 [AR] [AR]*.5 LONG, ASHC,  
; 5752 FE_FE+1, CALL [DROUND]  
; 5753 TL [FLG], FLG.SN/1, J/DNEG ;RESULT IS NEGATIVE  
; 5754 Q [MAG].AND.Q, ;HIGH WORD IS ZERO  
; 5755 HOLD RIGHT, J/DNORM1 ;GO TEST LOW WORD  
; 5756 =1110 [FLG]_O ;[122] CLEAR FLAG WORD  
; 5757 =  
; 5758 AC[1] [ARX].AND.[MAG], ;STORE LOW WORD  
; 5759 J/STAC ;GO DO HIGH WORD  
; 5760  
; 5761  
; 5762 DNORM1: READ Q, SKIP AD.EQ.O ;TEST LOW WORD  
; 5763 =0 [AR] [AR]*2 LONG, ;LOW WORD IS NON-ZERO  
; 5764 FE_FE-1, ASHC, ;ADJUST EXPONENT  
; 5765 NORM DISP, J/DNORM ;KEEP LOOKING  
; 5766 AC[1] [AR], J/STAC ;WHOLE ANSWER IS ZERO  
; 5767
```

; T1OKL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 157
FLOATING POINT -- DOUBLE PRECISION NORMALIZE

U 2060, 3266,7222,0000,4174,4007,0700,0000,0000,0000
U 2061, 2062,2222,0000,4174,4007,0511,4000,0000,0000
U 3266, 2062,4221,0013,4174,4007,0700,0000,0000,0000

U 2062, 0200,7441,0303,4174,4003,4701,0000,0000,0000
U 2063, 0200,2441,0303,4174,4003,4701,4000,0000,0000

U 0200, 0200,3444,0303,4174,4043,4701,1000,0041,1777
U 0202, 0312,3333,0003,4174,4003,4701,0010,0000,0000
U 0204, 0312,3446,0303,4174,4047,0700,1010,0041,0001
U 0206, 0312,3446,0303,4174,4047,0700,1010,0041,0001
U 0210, 3271,4002,0000,4174,0007,0700,0000,0000,0000
U 0216, 0630,4111,1204,4174,4007,0700,0000,0000,0000

U 0630, 3272,4111,0004,4174,4007,0700,0010,0000,0000
U 0632, 3267,6111,0004,4174,4007,0700,0000,0000,0000
U 0633, 2064,2441,0404,4174,4007,0561,4000,0000,0000
U 3267, 2064,4221,0013,4174,4007,0700,0000,0000,0000
U 2064, 3270,7333,0003,0174,4007,0700,0400,0000,0000
U 2065, 3270,2443,0300,0174,4007,0701,4400,0000,0000
U 3270, 0100,4113,0400,1174,4156,4700,0400,0000,1441
U 3271, 2066,3223,0000,4174,4007,0621,0000,0000,0000
U 2066, 0200,3444,0303,4174,4043,4701,1000,0041,1777
U 2067, 1515,3440,0303,1174,4007,0700,0400,0000,1441

```
; 5768 ;HERE TO NORMALIZE NEGATIVE D.P. RESULTS
; 5769 =0
; 5770 DNEG: Q_.NOT.Q, J/DNEG1 ;ONES COMP
; 5771 Q_-Q, SKIP CRY2, J/DNEG2
; 5772 DNEG1: [FLG]_O
; 5773 =0
; 5774 DNEG2: [AR]_.NOT.[AR], ;NO CARRY
; 5775 NORM DISP, J/DNNORM ;GO NORMALIZE
; 5776 [AR]_-[AR], ;CARRY
; 5777 NORM DISP, J/DNNORM ;NORMALIZE
; 5778
; 5779 =000*
; 5780 DNNORM: [AR]_[AR]*2 LONG, ;SHIFT 1 PLACE
; 5781 FE FE-1, ASHC, ;ADJUST EXPONENT
; 5782 NORM DISP, J/DNNORM ;LOOP TILL DONE
; 5783 =001* READ [AR], NORM DISP, ;SEE IF WE WENT TOO FAR
; 5784 CALL [DROUND] ; AND ROUND ANSWER
; 5785 =010* [AR]_[AR]*.5 LONG, ASHC,
; 5786 FE FE+1, CALL [DROUND]
; 5787 =011* [AR]_[AR]*.5 LONG, ASHC,
; 5788 FE FE+1, CALL [DROUND]
; 5789 =100* Q_[MAG].AND.Q, ;HIGH WORD IS ZERO
; 5790 HOLD RIGHT, J/DNNRM1 ;GO TEST LOW WORD
; 5791 =111* [ARX]_[ARX].AND.[MASK] ;REMOVE ROUNDING BIT
; 5792 =
; 5793 =00 [ARX]_[ARX].AND.[MAG], ;ALSO CLEAR SIGN
; 5794 CALL [CHKSN] ;ONES COMP?
; 5795 =10 [ARX]_[ARX].XOR.[MAG], ;YES--ONES COMP
; 5796 J/DNN1 ;CONTINUE BELOW
; 5797 =11 [ARX]_-[ARX], 3T, ;NEGATE RESULT
; 5798 SKIP CRY1, J/DNN2
; 5799 =
; 5800 DNN1: [FLG]_O ;CLEAR FLAG
; 5801 =0
; 5802 DNN2: AC_.NOT.[AR], J/DNORM2
; 5803 AC_-[AR], 3T
; 5804 DNORM2: AC[1]_[ARX].AND.[MAG], ;STORE LOW WORD
; 5805 NEXT INST ;ALL DONE
; 5806
; 5807 DNNRM1: READ Q, SKIP AD.EQ.O ;TEST LOW WORD
; 5808 =0 [AR]_[AR]*2 LONG, ;LOW WORD IS NON-ZERO
; 5809 FE FE-1, ASHC, ;ADJUST EXPONENT
; 5810 NORM DISP, J/DNNORM ;KEEP LOOKING
; 5811 AC[1]_[AR], J/STAC ;WHOLE ANSWER IS ZERO
; 5812
```

; T10KL.MCR[10,1141]
; FLT.MIC[10,1141]

15:34 27-JULY-1984
01:46 20-MAR-1981

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 158
FLOATING POINT -- DOUBLE PRECISION NORMALIZE

U 3272, 0002,4553,1300,4374,4004,1321,0000,0000,2000

; 5813
; 5814
; 5815
; 5816
; 5817
; 5818
; 5819

CHKSN: TL [FLG], FLG.SN/1, RETURN [2]
;SUBROUTINE TO SET/CLEAR FLG.SN
;CALL WITH:
CALL [SETSN], SKIP IF WE SHOULD CLEAR
;RETURNS 23
=0

U 2070, 0023,3551,1313,4374,0004,1700,0000,0000,2000

; 5820
; 5821
; 5822
; 5823

SETSN: [FLG]_[FLG].OR.#, FLG.SN/1, HOLD RIGHT, RETURN [23]
CLRSN: [FLG]_[FLG].AND.NOT.#, FLG.SN/1, HOLD RIGHT, RETURN [23]

U 2071, 0023,5551,1313,4374,0004,1700,0000,0000,2000

; 5824
; 5825
; 5826
; 5827
; 5828
; 5829

;SUBROUTINE TO ROUND A FLOATING POINT NUMBER
;CALL WITH:
NUMBER IN AR!Q AND NORM DISP
;RETURNS 16 WITH ROUNDED NUMBER IN AR!ARX

U 0312, 0462,0007,0704,4174,4007,0511,0000,0000,0000

; 5830
; 5831
; 5832
; 5833
; 5834
; 5835

DROUND: [ARX]_(Q+1)*.5, ;ROUND AND SHIFT
SKIP CRY2, ;SEE IF OVERFLOW
J/DRND1 ;COMPLETE ROUNDING
[AR]_[AR]*.5 LONG, ;WE WENT TOO FAR
FE_FE+1, ASHC, J/DROUND ;SHIFT BACK AND ROUND

U 0316, 0312,3446,0303,4174,4047,0700,1000,0041,0001

; 5836
; 5837
; 5838
; 5839

=*010
DRND1: [AR]_EXP, RETURN [16] ;NO OVERFLOW
=011 [AR]_[AR]+.25, ;ADD CARRY (BITS 36 AND 37
; ARE COPIES OF Q BITS)
;SEE IF OVERFLOW

U 0463, 0462,0441,0303,4174,4003,4701,4000,0000,0000

; 5840
; 5841
; 5842
; 5843

NORM DISP,
J/DRND1 ; ..
=110 [AR]_[AR]*.5, ;SHIFT RIGHT
FE_FE+1, ;KEEP EXP RIGHT
J/DRND1 ;ALL SET NOW

U 0466, 0462,3447,0303,4174,4007,0700,1000,0041,0001

; 5844
; 5845

=

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 159
EXTEND -- DISPATCH ROM ENTRIES

```

; 5846 .TOC "EXTEND -- DISPATCH ROM ENTRIES"
; 5847
; 5848 .DCODE
D 0001, 0001, 1740, 2100 ; 5849 001: I, SJCL, J/L-CMS
D 0002, 0002, 1740, 2100 ; 5850 I, SJCE, J/L-CMS
D 0003, 0003, 1740, 2100 ; 5851 I, SJCLE, J/L-CMS
D 0004, 0002, 1741, 2100 ; 5852 I, B/2, J/L-EDIT
D 0005, 0005, 1740, 2100 ; 5853 I, SJCGE, J/L-CMS
D 0006, 0006, 1740, 2100 ; 5854 I, SJCN, J/L-CMS
D 0007, 0007, 1740, 2100 ; 5855 I, SJCG, J/L-CMS
; 5856
D 0010, 0001, 1742, 2100 ; 5857 010: I, B/1, J/L-DBIN ;CVTDBO
D 0011, 0004, 1742, 2100 ; 5858 I, B/4, J/L-DBIN ;CVTDBT
D 0012, 0001, 1743, 2100 ; 5859 I, B/1, J/L-BDEC ;CVTBDO
D 0013, 0000, 1743, 2100 ; 5860 I, B/O, J/L-BDEC ;CVTBDT
; 5861
D 0014, 0001, 1744, 2100 ; 5862 014: I, B/1, J/L-MVS ;MOVSO
D 0015, 0000, 1744, 2100 ; 5863 I, B/O, J/L-MVS ;MOVST
D 0016, 0002, 1744, 2100 ; 5864 I, B/2, J/L-MVS ;MOVSLJ
D 0017, 0003, 1744, 2100 ; 5865 I, B/3, J/L-MVS ;MOVSRJ
; 5866
D 0020, 0000, 1746, 2100 ; 5867 020: I, J/L-XBLT ;XBLT
D 0021, 0000, 1747, 2100 ; 5868 I, J/L-SPARE-A ;GSNGL
D 0022, 0000, 1750, 2100 ; 5869 I, J/L-SPARE-B ;GDBLE
D 0023, 0000, 1751, 2100 ; 5870 I, B/O, J/L-SPARE-C ;GDFIX
D 0024, 0001, 1751, 2100 ; 5871 I, B/1, J/L-SPARE-C ;GFIX
D 0025, 0002, 1751, 2100 ; 5872 I, B/2, J/L-SPARE-C ;GDFIXR
D 0026, 0004, 1751, 2100 ; 5873 I, B/4, J/L-SPARE-C ;GFIXR
D 0027, 0010, 1751, 2100 ; 5874 I, B/10, J/L-SPARE-C ;DGFLTR
; 5875 ;30: ;GFLTR
; 5876 ;GFSC
; 5877 .UCODE
; 5878
```


; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
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
U 1741, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1742, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1743, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1744, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1746, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1747, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1750, 0400,4751,1203,4374,4007,0700,0000,0000,0040
U 1751, 0400,4751,1203,4374,4007,0700,0000,0000,0040

; 5879 1740:
; 5880 L-CMS: LUUD
; 5881 1741:
; 5882 L-EDIT: LUUD
; 5883 1742:
; 5884 L-DBIN: LUUD
; 5885 1743:
; 5886 L-BDEC: LUUD
; 5887 1744:
; 5888 L-MVS: LUUD
; 5889 1746:
; 5890 L-XBLT: LUUD
; 5891 1747:
; 5892 L-SPARE-A: LUUD
; 5893 1750:
; 5894 L-SPARE-B: LUUD
; 5895 1751:
; 5896 L-SPARE-C: LUUD
; 5897
; 5898 ;NOTE: WE DO NOT NEED TO RESERVE 3746 TO 3751 BECAUSE THE CODE
; 5899 ; AT EXTEND DOES A RANGE CHECK.
; 5900

; T1OKL.MCR[10,1141] 15:34 27-JULY-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, 1022, 3771, 0005, 4365, 5007, 0700, 0200, 0000, 0002

U 1022, 3556, 4553, 0500, 4374, 4007, 0321, 0010, 0076, 0740

U 1026, 3273, 4521, 0206, 4374, 4007, 0700, 0000, 0000, 0740

U 3273, 3274, 3111, 0605, 4174, 0417, 0700, 0000, 0000, 0000

U 3274, 3275, 3333, 0005, 4174, 4217, 0700, 0000, 0000, 0500

U 3275, 3276, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0240

U 3276, 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, 3277, 0551, 0505, 2270, 4007, 0700, 0200, 0004, 0512

U 0176, 3277, 3443, 0500, 4174, 4007, 0700, 0200, 0004, 0512

U 3277, 3276, 3771, 0005, 4361, 5217, 0700, 0200, 0000, 0502

U 0556, 3400, 3333, 0005, 7174, 4001, 2700, 0400, 0000, 0241

U 0557, 2072, 3333, 0005, 4174, 4007, 0530, 0000, 0000, 0000

U 2072, 3400, 3333, 0005, 7174, 4001, 2700, 0400, 0000, 0241

U 2073, 0556, 3771, 0005, 4374, 0007, 0700, 0000, 0077, 7777

```
; 5901 .TOC "EXTEND -- INSTRUCTION SET DECODING"
; 5902
; 5903 ;EACH INSTRUCTION IN THE RANGE 1-23 GOES TO 1 OF 2 PLACES
; 5904 ; 1740-1747 IF NOT UNDER EXTEND
; 5905 ; 3740-3747 IF UNDER EXTEND
; 5906
; 5907 .DCODE
; 5908 123: I,READ/1, J/EXTEND
; 5909 .UCODE
; 5910
; 5911 1467:
; 5912 EXTEND: MEM READ, [BR] MEM ;FETCH INSTRUCTION
; 5913 =0** TL [BR], #/760740, ;IN RANGE 0-17 (AND AC#=0)
; 5914 CALL [BITCHK] ;TRAP IF NON-ZERO BITS FOUND
; 5915 [BRX] [HR].AND.# CLR RH, ;SPLIT OUT AC NUMBER
; 5916 #/000740 ; FROM EXTEND INSTRUCTION
; 5917 [BR] [BR].OR.[BRX], ;LOAD IR AND AC #
; 5918 HOLD RIGHT, LOAD IR ;
; 5919 READ [BR], LOAD BYTE EA, ;LOAD XR #
; 5920 J/EXTEAO ;COMPUTE E1
; 5921
; 5922 EXTEAO: WORK[EO]_[AR]
; 5923 EXTEA1: EA MODE DISP
; 5924 =100*
; 5925 EXTEA: [BR]_[BR]+XR
; 5926 EXTDSP: [BR]_EA FROM [BR], LOAD VMA,
; 5927 B DISP, J/EXTEXT
; 5928 [BR] [BR]+XR, START READ, PXCT EXTEND EA, LOAD VMA, J/EXTIND
; 5929 VMA_[BR], START READ, PXCT EXTEND EA
; 5930
; 5931 EXTIND: MEM READ, [BR]_MEM, HOLD LEFT, LOAD BYTE EA, J/EXTEA1
; 5932
; 5933 ;HERE TO EXTEND SIGN FOR OFFSET MODES
; 5934 =1110
; 5935 EXTEXT: WORK[E1]_[BR], ;SAVE E1
; 5936 DISP/DROM, J/3400 ;GO TO EXTENDED EXECUTE CODE
; 5937 READ [BR], SKIP DP18 ;NEED TO EXTEND SIGN
; 5938 =0 WORK[E1]_[BR], ;POSITIVE
; 5939 DISP/DROM, J/3400
; 5940 [BR] #, #/777777, HOLD RIGHT, ;NEGATIVE
; 5941 J/EXTEXT
; 5942
```

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 162
EXTEND -- MOVE STRING -- SETUP

U 3744, 3472,0111,0703,4174,4007,0700,0210,0004,0012
U 3754, 2100,3771,0005,1276,6007,0701,0000,0000,1443
U 2100, 3556,4553,0500,4374,4007,0321,0010,0077,7000
U 2104, 2074,3771,0003,0276,6007,0700,0000,0000,0000
U 2074, 3474,4521,0306,4374,4007,0700,0010,0077,7000
U 2075, 2076,2113,0305,1174,4007,0521,4400,0000,1443
U 2076, 3300,7441,0503,4174,4007,0700,0000,0000,0000
U 2077, 3300,7441,0303,4174,4007,0700,0000,0000,0000
U 3300, 0574,3333,0003,7174,4003,7700,0400,0000,0242
U 0574, 0500,3771,0013,4370,4007,0700,0000,0000,0003
U 0575, 3301,3771,0005,1276,6007,0701,0000,0000,1444
U 0576, 3321,3441,0304,4174,4007,0700,0000,0000,0000
U 0577, 0650,3771,0004,1276,6007,0522,0000,0000,1443
U 3301, 3302,3333,0005,4174,4007,0700,1000,0041,6020
U 3302, 2101,4222,0000,4174,4006,7701,1000,0041,1770
U 2101, 2101,4224,0003,4174,4026,7701,1000,0041,1770
U 2103, 3303,7221,0003,4174,4007,0700,0000,0000,0000
U 3303, 0507,4113,0312,7174,4007,0700,0400,0000,0243

```
; 5943 .TOC "EXTEND -- MOVE STRING -- SETUP"
; 5944
; 5945 ;HERE TO MOVE A STRING
; 5946 ;COME HERE WITH:
; 5947 ; AR/ EO
; 5948 ; BR/ E1
; 5949 ;
; 5950 3744:
; 5951 MVS: [AR]_[AR]+1, ;GO FETCH FILL
; 5952 LOAD VMA, ; BYTE
; 5953 START READ, ;
; 5954 CALL [GTFILL] ;SUBROUTINE TO COMPLETE
; 5955 [BR] AC[DLEN] ;GET DEST LENGTH AND FLAGS
; 5956 =0** TL [BR], #/777000, ;ANY FLAGS SET?
; 5957 CALL [BITCHK] ;SEE IF ILLEGAL
; 5958 [AR] AC ;GET SRC LENGTH AND FLAGS
; 5959 =0 [BRX]_[AR].AND.# CLR RH, ;COPY FLAGS TO BRX
; 5960 #/777000, ;
; 5961 CALL [CLRFLG] ;CLEAR FLAGS IN AR
; 5962 ;NEW DLEN IS <SRC LEN>-<DST LEN>
; 5963 AC[DLEN]_[AR]-[BR], 3T, ;COMPUTE DIFFERENCE
; 5964 SKIP DPO ;WHICH IS SHORTER?
; 5965 =0 [AR].NOT.[BR], ;DESTINATION
; 5966 J/MVS1 ;GET NEGATIVE LENGTH
; 5967 [AR].NOT.[AR] ;SOURCE
; 5968 MVS1: WORK[SLEN]_[AR], ;
; 5969 B DISP ;SEE WHAT TYPE OF MOVE
; 5970 ;SLEN NOW HAS -<LEN OF SHORTER STRING>-1
; 5971 =1100
; 5972 STATE_[SRC], J/MOVL P ;TRANSLATE--ALL SET
; 5973 [BR] AC[DSTP], J/MVSO ;OFFSET BUILD MASK
; 5974 [ARX]_[AR], ;LEFT JUSTIFY
; 5975 J/MOVSTO ;
; 5976 [ARX]_AC[DLEN], ;RIGHT JUSTIFY
; 5977 SKIP DPO, 4T, ;WHICH IS SHORTER?
; 5978 J/MOVRJ
; 5979
; 5980 MVSO: READ [BR], FE_S+2 ;GET DST BYTE SIZE
; 5981 Q_O, BYTE STEP ;BUILD AN S BIT MASK
; 5982 =0*
; 5983 MVSO1: GEN MSK [AR], BYTE STEP, J/MVS01
; 5984 [AR].NOT.Q ;BITS WHICH MUST NOT BE SET
; 5985 WORK[MSK]_[AR].AND.[MASK], ;SAVE FOR SRCMOD
; 5986 J/MOVLPO ;GO ENTER LOOP
; 5987
```

44 c100 1000
92/ 700 1000 S6
5 DL
7 + 2 (-2)

; T10KL.MCR[10,1141] 15:34 27-JULY-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

U 0500, 1114,0551,0703,7274,4007,0701,0010,0000,0242
U 0501, 1020,3441,0304,4174,4007,0520,0000,0000,0000
U 0504, 3304,1771,0003,7274,4007,0701,4000,0000,0242
U 0505, 3464,3771,0013,4370,4007,0700,0010,0000,0005
U 0507, 0500,3771,0013,4370,4007,0700,0000,0000,0003

U 3304, 2106,3771,0005,1276,6007,0522,0000,0000,1443
U 2106, 3305,3440,0303,1174,4007,0700,0400,0000,1443
U 2107, 2106,1111,0503,4174,4007,0700,4000,0000,0000
U 3305, 3306,7771,0003,7274,4007,0701,0000,0000,0242
U 3306, 2110,3333,0005,4174,4007,0520,0000,0000,0000
U 2110, 2111,0111,0503,4174,4007,0700,0000,0000,0000
U 2111, 3307,3111,0603,4174,4007,0700,0000,0000,0000
U 3307, 1515,4221,0013,4170,4007,0700,0000,0000,0000

```
; 5988 .TOC "EXTEND -- MOVE STRING -- OFFSET/TRANSLATE"  
; 5989  
; 5990 ;HERE IS THE LOOP FOR OFFSET AND TRANSLATED MOVES  
; 5991 =000  
; 5992 MOVLEP: [AR]_WORK[SLEN]+1, ;UPDATE STRING LENGTH  
; 5993 CALL [SRCMOD] ;GET A SOURCE BYTE  
; 5994 =001 [ARX]_[AR], SKIP DPO, ;(1) LENGTH EXHAUSTED  
; 5995 J/MOVST2 ; SEE IF FILL IS NEEDED  
; 5996 =100 [AR]_WORK[SLEN], ;(4) ABORT  
; 5997 J/MVABT ;  
; 5998 STATE [SRC+DST], ;(5) NORMAL--STORE DST BYTE  
; 5999 CALL [PUTDST] ;  
; 6000 =111  
; 6001 MOVLP0: STATE_[SRC], J/MOVLEP ;(7) DPB DONE  
; 6002 =  
; 6003  
; 6004 ;HERE TO ABORT A STRING MOVE DUE TO TRANSLATE OR OFFSET FAILURE  
; 6005  
; 6006 MVABT: [BR]_AC[DLEN], ;WHICH STRING IS LONGER  
; 6007 SKIP_DPO, 4T  
; 6008 =0  
; 6009 MVABT1: AC[DLEN]_[AR], J/MVABT2 ;PUT AWAY DEST LEN  
; 6010 [AR]_[AR]-[BR], ;DEST LEN WAS GREATER  
; 6011 J/MVABT1 ;STICK BACK IN AC  
; 6012  
; 6013 MVABT2: [AR]_.NOT.WORK[SLEN] ;GET UNDECREMENTED SLEN  
; 6014 READ [BR], SKIP DPO ;NEED TO FIXUP SRC?  
; 6015 =0 [AR]_[AR]+[BR] ;SRC LONGER BY (DLEN)  
; 6016 MVEND: [AR]_[AR].OR.[BRX] ;PUT BACK SRC FLAGS  
; 6017 END STATE, J/STAC ;ALL DONE  
; 6018
```

Produced on Advanced Information Services Electronic Laser Printer. PGO/IES6, DTN: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```

; 6019 .TOC "EXTEND -- MOVE STRING -- MOVSRJ"
; 6020
; 6021 =00
U 0650, 3310,3771,0003,1276,6007,0701,0000,0000,1441 ; 6022 MOVSRJ: [AR] AC[SRCP], J/MVSKP ;SRC LONGER, SKIP OVER SOME
; 6023 STATE [DSTF], ;DST LONGER, FILL IT
U 0651, 2301,3771,0013,4370,4007,0700,0010,0000,0006 ; 6024 CALL [MOVFIL] ;
; 6025 =11 [ARX] WORK[SLEN]+1, ;DONE FILLING
U 0653, 3322,0551,0704,7274,4007,0701,0000,0000,0242 ; 6026 J/MOVST1 ;GO MOVE STRING
; 6027
; 6028 ;HERE TO SKIP OVER EXTRA SOURCE BYTES
U 3310, 2112,3440,0303,1174,4007,0670,0400,0000,1441 ; 6029 MVSKP: AC[SRCP]_[AR], SKIP -1MS ;[121] Is there a timer interrupt?
U 2112, 3313,3333,0003,7174,4007,0700,0400,0000,0211 ; 6030 =0 WORK[SV.AR]_[AR], J/MVSK2 ;[121][123] Yes, save regs for int
rrupt.
; 6031 [ARX]_[ARX]-1, 3T. ;DONE SKIPPING?
U 2113, 2114,1111,0704,4174,4007,0521,4000,0000,0000 ; 6032 SKIP DPO
; 6033 =0 IBP DP, IBP SCAD, ;NO--START THE IBP
; 6034 SCAD DISP, SKIP IRPT, ;4-WAY DISPATCH
U 2114, 0750,3770,0305,4334,4016,7371,0000,0033,6000 ; 6035 3T, J/MVSKP1 ;GO BUMP POINTER
; 6036 AC[DLEN]_0, ;LENGTHS ARE NOW EQUAL
U 2115, 0546,4223,0000,1174,4007,0700,0400,0000,1443 ; 6037 J/MOVST4 ;GO MOVE STRING
; 6038
; 6039 =00
U 0750, 3310,3441,0503,4174,4007,0700,0000,0000,0000 ; 6040 MVSKP1: [AR]_[BR], J/MVSKP ;NO OVERFLOW.
; 6041 [AR]_.NOT.WORK[SLEN], ;INTERRUPT
U 0751, 3311,7771,0003,7274,4007,0701,0000,0000,0242 ; 6042 J/MVSK3 ;
; 6043 SET P TO 36-S, ;WORD OVERFLOW
U 0752, 3312,3770,0503,4334,4017,0700,0000,0032,6000 ; 6044 J/MVSKP2 ;FIXUP Y
U 0753, 3311,7771,0003,7274,4007,0701,0000,0000,0242 ; 6045 [AR]_.NOT.WORK[SLEN] ;[121] INTERRUPT or timer.
U 3311, 2116,3440,0303,1174,4007,0700,0400,0000,1443 ; 6046 MVSK3: AC[DLEN]_[AR] ;RESET DLEN
; 6047 =0 [AR]_[AR]+[ARX],
; 6048 CALL [INCAR] ;ADD 1 TO AR
U 2116, 3677,0111,0403,4174,4007,0700,0010,0000,0000 ; 6049 AC [AR].OR.[BRX], ;PUT BACK FLAGS
; 6050 J/ITRAP ;DO INTERRUPT TRAP
; 6051
U 2117, 3733,3113,0306,0174,4007,0700,0400,0000,0000 ; 6052 MVSKP2: [AR]_[AR]+1, HOLD LEFT, ;BUMP Y
; 6053 J/MVSKP ;KEEP GOING
; 6054
; 6055
U 3312, 3310,0111,0703,4170,4007,0700,0000,0000,0000 ; 6056 MVSK2: WORK[SV.BR] [BR] ;BEGIN EDIT [123]
; 6057 WORK[SV.ARX]_[ARX] ;SAVE ALL
; 6058 WORK[SV.BRX]_[BRX] ;THE REGISTERS
; 6059 =0* CALL [TICK] ;FOR THE TICK
; 6060 [AR] WORK[SV.AR] ;UPDATE CLOCK AND SET INTERUPT
; 6061 [BR] WORK[SV.BR] ;NOW PUT
; 6062 [ARX] WORK[SV.ARX] ;THEM ALL
; 6063 [BRX] WORK[SV.BRX], ;BACK SO WE
; 6064 J/MVSKP ;CAN CONTINUE
; 6065 ;END EDIT [123]
; 6066
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

U 3321, 3322,0111,0704,4174,4007,0700,0000,0000,0000
U 3322, 0540,3771,0013,4370,4007,0700,0000,0000,0003

U 0540, 2312,3333,0004,7174,4007,0520,0410,0000,0242
U 0541, 1020,3441,0304,4174,4007,0520,0000,0000,0000
U 0542, 3464,3771,0013,4370,4007,0700,0010,0000,0005

U 0546, 3322,0551,0704,7274,4007,0701,0000,0000,0242

U 1020, 3323,4223,0000,1174,4007,0700,0400,0000,1443
U 1021, 2301,3771,0013,4370,4007,0700,0010,0000,0004
U 1023, 2161,3440,0606,0174,4007,0700,0400,0000,0000

U 3323, 3324,3113,0406,0174,4007,0700,0400,0000,0000
U 3324, 0252,4221,0013,4170,4007,0700,0000,0000,0000

```
; 6067 .TOC "EXTEND -- MOVE STRING -- SIMPLE MOVE LOOP"  
; 6068  
; 6069 ;HERE FOR NO-MODIFICATION STRING MOVES  
; 6070 MOVST0: [ARX]_[ARX]+1 ;CANT DO [ARX]_[AR]+1  
; 6071 MOVST1: STATE_[SRC] ;PREPARE FOR PAGE FAIL  
; 6072 =000  
; 6073 WORK[SLEN]_[ARX], ;GO GET A SOURCE BYTE  
; 6074 SKIP DPO, CALL [GSRG] ;  
; 6075 MOVSTX: [ARX]_[AR], ;SHORT STRING RAN OUT  
; 6076 SKIP DPO, J/MOVST2 ;GO SEE IF FILL NEEDED  
; 6077 =010 STATE_[SRC+DST], ;WILL NEED TO BACK UP BOTH POINTERS  
; 6078 CALL [PUTDST] ;STORE BYTE  
; 6079 =110  
; 6080 MOVST4: [ARX]_WORK[SLEN]+1, ;COUNT DOWN LENGTH  
; 6081 J/MOVST1 ;LOOP OVER STRING  
; 6082 =  
; 6083 =00  
; 6084 MOVST2: AC[DLEN]_0, J/MOVST3 ;CLEAR DEST LEN, REBUILD SRC  
; 6085 STATE_[DST], CALL [MOVFIL] ;FILL OUT DEST  
; 6086 =11 AC_[BRX], J/ENDSKP ;ALL DONE  
; 6087  
; 6088 MOVST3: AC_[ARX].OR.[BRX] ;REBUILD SRC  
; 6089 END STATE, J/SKIPE ;  
; 6090
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 166
EXTEND -- COMPARE STRING

U 3740, 2121,3771,0004,1276,6007,0701,0000,0000,1443
U 2121, 3556,4553,0400,4374,4007,0321,0010,0077,7000
U 2125, 2123,3771,0006,0276,6007,0700,0000,0000,0000
U 2123, 3556,4553,0600,4374,4007,0321,0010,0077,7000
U 2127, 2130,2113,0604,4174,4007,0521,4000,0000,0000
U 2130, 2131,0111,0703,4174,4007,0700,0000,0000,0000
U 2131, 2132,0111,0703,4170,4007,0700,0200,0004,0012

U 2132, 3675,4221,0003,4174,4007,0700,0010,0000,0000

U 2133, 3334,3223,0000,7174,4007,0700,0400,0000,0244

```
; 6091 .TOC "EXTEND -- COMPARE STRING"  
; 6092  
; 6093 3740:  
; 6094 CMS: [ARX]_AC[DLEN] ;GET DEST LEN  
; 6095 =0** TL [ARX], #/777000, CALL [BITCHK]  
; 6096 [BRX]_AC ;GET SRC LEN  
; 6097 =0** TL [BRX], #/777000, CALL [BITCHK]  
; 6098 [BRX]-[ARX], 3T, SKIP DPO ;WHICH STRING IS LONGER?  
; 6099 =0 [AR]_[AR]+1 ;SRC STRING IS LONGER  
; 6100 VMA_[AR]+1, START READ ;DST STRING  
; 6101 =0 [AR]_O, ;FORCE FIRST COMPARE TO BE  
; 6102 ;EQUAL  
; 6103 CALL [LOADQ] ;PUT FILL INTO Q  
; 6104 WORK[FILL]_Q, ;SAVE FILLER  
; 6105 J/CMS2 ;ENTER LOOP  
; 6106
```

Produced on Advanced Information Services Electronic Laser Printer. PKO/SES, DTN: 221-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 167
EXTEND -- COMPARE STRING

U 2134, 0250,4221,0013,4170,4003,7700,0000,0000,0000
U 2135, 3325,3771,0003,1276,6007,0701,0000,0000,1441
U 3325, 1030,3333,0006,4174,4007,0520,0000,0000,0000

U 1030, 2313,3771,0013,4370,4007,0700,0010,0000,0011

U 1031, 2136,3333,0004,4174,4007,0520,0000,0000,0000
U 1032, 3326,3333,0003,7174,4007,0700,0400,0000,0245

U 3326, 3327,3440,0606,0174,4007,0700,0400,0000,0000
U 3327, 3330,3771,0013,4370,4007,0700,0000,0000,0010
U 3330, 1050,3333,0004,4174,4007,0520,0000,0000,0000

U 1050, 2140,4443,0000,4174,4007,0700,0010,0000,0000

U 1051, 3331,3771,0003,7274,4007,0701,0000,0000,0244
U 1052, 3331,3440,0404,1174,4007,0700,0400,0000,1443

U 3331, 3332,4111,1203,7174,4007,0700,0000,0000,0245
U 3332, 3333,4551,1205,7274,4007,0700,0000,0000,0245
U 3333, 3334,2111,0503,4174,4007,0700,4000,0000,0000
U 3334, 3335,1111,0704,4174,4007,0700,4000,0000,0000
U 3335, 3336,1111,0706,4174,4007,0700,4000,0000,0000
U 3336, 2134,3333,0003,4174,4007,0621,0000,0000,0000

U 2136, 3337,3772,0000,7274,4007,0701,0000,0000,0244
U 2137, 2134,4221,0003,4174,4007,0700,0000,0000,0000
U 3337, 3340,3771,0013,4370,4007,0700,0000,0000,0012
U 3340, 1050,3223,0000,7174,4007,0700,0400,0000,0245

U 2140, 3465,3771,0003,1276,6007,0701,0010,0000,1444

U 2141, 0340,3333,0003,4174,4006,5701,1000,0051,0770

; 6107 ;HERE IS THE COMPARE LOOP.
; 6108 ; ARX/ CONATINS REMAINING DEST LENGTH
; 6109 ; BRX/ CONTAINS REMAINING SOURCE LENGTH
; 6110 =0
; 6111 CMS3: ;BYTES ARE NOT EQUAL
; 6112 END STATE, ;NO MORE SPECIAL PAGE FAIL ACTION
; 6113 SKIP-COMP DISP ;SEE SKIP-COMP-TABLE
; 6114 CMS4: [AR]_AC[SRCP] ;GET BYTE POINTER
; 6115 READ [BRX], SKIP DPO ;MORE IN SOURCE STRING?
; 6116 =00 STATE [EDIT-SRC], ;PREPARE FOR PAGE FAIL
; 6117 CALL [GETSRC] ; GO GET BYTE
; 6118 READ [ARX], SKIP DPO, ;NO MORE SRC--SEE IF MORE DEST
; 6119 J/CMS5 ;
; 6120 WORK[CMS]_[AR] ;SAVE SRC BYTE
; 6121 =
; 6122 AC [BRX] ;PUT BACK SRC LEN
; 6123 STATE [COMP-DST] ;HAVE TO BACK UP IF DST FAILS
; 6124 READ [ARX], SKIP DPO ;ANY MORE DEST?
; 6125 =00
; 6126 CMS6: CALL [CMPDST] ;MORE DEST BYTES
; 6127 [AR]_WORK[FILL], ;OUT OF DEST BYTES
; 6128 J/CMS7 ;GO DO COMPARE
; 6129 AC[DLEN]_[ARX] ;GOT A BYTE, UPDATE LENGTH
; 6130 =
; 6131 CMS7: [AR]_[AR].AND.[MASK], ;MAKE MAGNITUDES
; 6132 WORK[CMS] ;WARM UP RAM
; 6133 [BR]_[MASK].AND.WORK[CMS], 2T ;GET SRC MAGNITUDE
; 6134 [AR]_[BR]-[AR] REV ;UNSIGNED COMPARE
; 6135 CMS2: [ARX]_[ARX]-1 ;UPDATE LENGTHS
; 6136 [BRX]_[BRX]-1 ;
; 6137 READ [AR], SKIP AD.EQ.O, J/CMS3 ;SEE IF EQUAL
; 6138
; 6139 =0
; 6140 CMS5: Q_WORK[FILL], J/CMS8 ;MORE DST--GET SRC FILL
; 6141 [AR]_O, J/CMS3 ;STRINGS ARE EQUAL
; 6142 CMS8: STATE [EDIT-DST] ;JUST DST POINTER ON PAGE FAIL
; 6143 WORK[CMS]_Q, J/CMS6 ;MORE DST--SAVE SRC FILL
; 6144
; 6145 =0
; 6146 CMPDST: [AR]_AC[DSTP], ;GET DEST POINTER
; 6147 CALL [IDST] ;UPDATE IT
; 6148 READ [AR], ;LOOK AT BYTE POINTER
; 6149 FE FE.AND.S#, S#/0770, ;MASK OUT BIT 6
; 6150 BYTE DISP, J/LDB1 ;GO LOAD BYTE
; 6151

; T1OKL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 168
EXTEND -- DECIMAL TO BINARY CONVERSION

```
; 6152 .TOC "EXTEND -- DECIMAL TO BINARY CONVERSION"  
; 6153  
; 6154 3742:  
U 3742, 3341,4571,1203,4374,4007,0700,0000,0077,7777 ; IF WE ARE IN OFFSET MODE  
U 3341, 3342,3333,0003,7174,4007,0700,0400,0000,0243 ; ONLY ALLOW 18 BITS  
; 6155 DBIN: [AR] [777777] XWD 0 ; RANGE CHECKED (0-10) LATER  
; 6156 WORK[MASK]_[AR] ; GET SRC LENGTH  
; 6157 ; SPLIT OUT FLAGS  
U 3342, 3343,3771,0003,0276,6007,0700,0000,0000,0000 ;  
; 6158 [AR] AC ;  
; 6159 [BRX]_[AR].AND.# CLR RH, ;  
; 6160 #/777000 ;  
; 6161 =0* [AR]_AC[BIN1], ; GET LOW WORD  
; 6162 CALL [CLRAXO] ; CLEAR BIT 0 OF ARX  
; 6163 AC[BIN1]_[ARX] ; STORE BACK  
; 6164 =0 READ [BRX], SKIP DPO, ; IS S ALREADY SET?  
; 6165 CALL [CLRBIN] ; GO CLEAR BIN IF NOT  
; 6166 [AR]_[AR].AND.#, ; CLEAR FLAGS FROM LENGTH  
; 6167 #/000777, HOLD RIGHT, ;  
; 6168 B DISP ; SEE IF OFFSET OR TRANSLATE  
; 6169 =1110  
U 3343, 2124,4521,0306,4374,4007,0700,0000,0077,7000 ;  
; 6170 DBIN1: STATE [CVTDB], J/DBIN2 ; TRANSLATE--LEAVE S ALONE  
; 6171 [BRX]_[BRX].OR.#, ; OFFSET--FORCE S TO 1  
; 6172 #/400000, HOLD RIGHT, ;  
; 6173 J/DBIN1 ;  
U 2124, 2236,3771,0004,1276,6007,0701,0010,0000,1444 ;  
U 2126, 2142,3440,0404,1174,4007,0700,0400,0000,1444 ;  
; 6174 DBIN2: WORK[SLEN]_.NOT.[AR] ; STORE -SLEN-1  
; 6175 ;  
; 6176 ; HERE IS THE MAIN LOOP  
; 6177 =0*0  
U 2142, 2166,3333,0006,4174,4007,0520,0010,0000,0000 ;  
; 6178 DBINLP: [AR]_WORK[SLEN]+1, CALL [SRCMOD] ; (0) GET MODIFIED SRC BYTE  
; 6179 TL [BRX], #/100000, ; (1) DONE, IS M SET?  
; 6180 J/DBXIT ;  
; 6181 [AR]_.NOT.WORK[SLEN], ; (4) ABORT  
; 6182 J/DBABT ;  
; 6183 [AR]-#, #/10., ; (5) NORMAL--SEE IF 0-9  
; 6184 4T, SKIP DP18 ;  
; 6185 =0 [AR]_.NOT.WORK[SLEN], ; DIGIT TOO BIG  
; 6186 J/DBABT ; GO ABORT CVT  
; 6187
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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 2145, 1074,3771,0005,1276,6007,0622,0000,0000,1443

U 1074, 0560,3771,0004,1276,6007,0701,0010,0000,1444

U 1075, 3345,3771,0005,1276,6007,0701,0000,0000,1444

U 1076, 0460,4443,0000,4174,4007,0700,0000,0000,0000

U 3345, 2146,4553,0500,4374,4007,0321,0000,0076,0000

U 2146, 1074,4443,0000,4174,4007,0700,0000,0000,0000

U 2147, 3346,3775,0005,1276,6007,0701,0000,0000,1444

U 3346, 2150,3445,0505,1174,4007,0700,0000,0000,1444

U 2150, 3700,0551,0505,1274,4007,0700,0010,0000,1444

U 2151, 0460,0113,0305,1174,4007,0701,0400,0000,1444

```
; 6188 ;HERE TO ADD IN A DIGIT
; 6189 [BR]_AC[BINO], 4T, ;GET HIGH BINARY
; 6190 SKIP AD.EQ.O ;SEE IF SMALL
; 6191 =0
; 6192 DBSLO: [ARX]_AC[BIN1], ;TOO BIG
; 6193 CALL [DBSLOW] ;GO USE DOUBLE PRECISION PATHS
; 6194 [BR]_AC[BIN1], ;GET LOW WORD
; 6195 J/DBFAST ;MIGHT FIT IN 1 WORD
; 6196 J/DBINLP ;RETURN FROM DBSLOW
; 6197 ;GO DO NEXT DIGIT
; 6198 =
; 6199 DBFAST: TL [BR], #/760000 ;WILL RESULT FIT IN 36 BITS?
; 6200 =0 J/DBSLO ;MAY NOT FIT--USE DOUBLE WORD
; 6201 [BR]_AC[BIN1]*2 ;COMPUTE AC*2
; 6202 [BR]_[BR]*2, AC[BIN1] ;COMPUTE AC*4
; 6203 =0 [BR]_[BR]+AC[BIN1], 2T, ;COMPUTE AC*5
; 6204 CALL [SBRL] ;COMPUTE AC*10
; 6205 AC[BIN1]_[AR]+[BR], 3T, ;NEW BINARY RESULT
; 6206 J/DBINLP ;DO NEXT DIGIT
; 6207
```

; T10KLMICR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS: MICROCODE V124, 27-JUL-84 Page 170
EXTEND -- DECIMAL TO BINARY CONVERSION

U 0560, 0620,3771,0005,1276,6007,0701,0010,0000,1443

U 0561, 2152,0551,0404,1274,4007,0562,0010,0000,1444

U 0565, 0600,0551,0505,1274,4007,0701,0000,0000,1443

U 0600, 0621,4443,0000,4174,4007,0700,0010,0000,0000

U 0601, 2152,0111,0304,4174,4007,0561,0010,0000,0000

U 0605, 3347,3440,0404,1174,4007,0700,0400,0000,1444

U 3347, 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, 2152,0111,0404,4174,4007,0561,0010,0000,0000

U 0626, 0001,4443,0000,4174,4004,1700,0000,0000,0000

U 2152, 0004,4443,0000,4174,4004,1700,0000,0000,0000

U 2153, 3350,4551,0404,4374,0007,0700,0000,0037,7777

U 3350, 0004,0111,0705,4174,4004,1700,0000,0000,0000

```
; 6208 ;HERE IF NUMBER DOES NOT FIT IN ONE WORD
; 6209
; 6210 =000
; 6211 DBSLOW: [BR]_AC[BINO], ;FETCH HIGH WORD
; 6212 CALL [MULBY4] ;MULTIPLY BY 4
; 6213 [ARX]_[ARX]+AC[BIN1], ;COMPUTE VALUE * 5
; 6214 SKIP CRY1, 4T, ;SEE IF OVERFLOW
; 6215 CALL [ADDCRY] ;GO ADD CARRY
; 6216 =101 [BR]_[BR]+AC[BINO] ;ADD IN HIGH WORD
; 6217 =
; 6218 =000 CALL [DBLDBL] ;MAKE * 10
; 6219 [ARX]_[ARX]+[AR], 3T. ;ADD IN NEW DIGIT
; 6220 SKIP CRY1, ;SEE IF OVERFLOW
; 6221 CALL [ADDCRY] ;ADD IN THE CARRY
; 6222 =101 AC[BIN1]_[ARX] ;PUT BACK ANSWER
; 6223 =
; 6224 AC[BINO]_[BR], ;...
; 6225 RETURN [2] ;GO DO NEXT BYTE
; 6226
; 6227 ;HERE TO DOUBLE BR!ARX
; 6228 =000
; 6229 MULBY4: CALL [DBLDBL] ;DOUBLE TWICE
; 6230 DBLDBL: [BR]_[BR]+[BR] ;DOUBLE HIGH WORD FIRST
; 6231 ;(SO WE DON'T DOUBLE CARRY)
; 6232 [ARX]_[ARX]+[ARX], ;DOUBLE LOW WORD
; 6233 SKIP CRY1, 3T, ;SEE IF CARRY
; 6234 CALL [ADDCRY] ;ADD IN CARRY
; 6235 =110 RETURN [1] ;ALL DONE
; 6236 =
; 6237
; 6238 ;HERE TO ADD THE CARRY
; 6239 =0
; 6240 ADDCRY: RETURN [4] ;NO CARRY
; 6241 CLEAR [ARX]0 ;KEEP LOW WORD POSITIVE
; 6242 [BR]_[BR]+1, ;ADD CARRY
; 6243 RETURN [4] ;ALL DONE
; 6244
```

Produced on Advanced Information Services Electronic Laser Printer, PGO/ISE6, DTH: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-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 3351, 3352,3111,0306,4174,4007,0700,0000,0000,0000

U 3352, 2155,1111,0701,4170,4007,0700,4000,0000,0000

U 2154, 3355,3771,0004,1276,6007,0701,0000,0000,1444

U 2155, 3353,3771,0003,1276,6007,0701,0000,0000,1444

U 3353, 2156,3771,0005,1276,6007,0522,0000,0000,1443

U 2156, 3354,4551,0303,4374,0007,0700,0000,0037,7777

U 2157, 3354,3551,0303,4374,0007,0700,0000,0040,0000

U 3354, 2160,3440,0303,1174,4007,0700,0400,0000,1444

U 2160, 2166,3770,0606,0174,4007,0520,0410,0000,0000

U 2161, 0260,4221,0013,4170,4007,0700,0000,0000,0000

U 3355, 3356,4551,0404,4374,0007,0700,0000,0037,7777

U 3356, 2162,2441,0404,1174,4007,0621,4000,0000,1443

U 2162, 2165,7771,0003,1274,4007,0700,0000,0000,1443

U 2163, 2164,1771,0003,1274,4007,0621,4000,0000,1443

U 2164, 2165,4571,1204,4374,4007,0700,0000,0040,0000

U 2165, 3357,3440,0303,1174,4007,0700,0400,0000,1443

U 3357, 2155,3440,0404,1174,4007,0700,0400,0000,1444

U 2166, 3360,4223,0000,1174,4007,0700,0400,0000,1443

U 2167, 0001,4443,0000,4174,4004,1700,0000,0000,0000

U 3360, 0001,4223,0000,1174,4004,1700,0400,0000,1444

```
; 6245 ;HERE TO ABORT CONVERSION
; 6246 DBABT: [BRX] [BRX].OR.[AR] ;PUT BACK UNUSED LENGTH
; 6247 [PC] [PC]-1, HOLD LEFT, ;DO NOT SKIP
; 6248 J/DBDONE ;GO FIX UP SIGN COPY
; 6249
; 6250 ;HERE AT END
; 6251 =0
; 6252 DBXIT: [ARX]_AC[BIN1], ;GET LOW WORD
; 6253 J/DBNEG ;GO NEGATE
; 6254 DBDONE: [AR]_AC[BIN1] ;FETCH LOW WORD
; 6255 [AR]_AC[BINO], 4T, ;GET HIGH WORD
; 6256 SKIP DPO ;WHAT SIGN
; 6257 =0 CLEAR [AR]0, J/DBDN1 ;POSITIVE
; 6258 [AR] [AR].OR.#, #/400000, HOLD RIGHT
; 6259 DBDN1: AC[BIN1]_[AR] ;STORE AC BACK
; 6260 =0 AC [BRX] TEST, ;RETURN FLAGS
; 6261 SKIP DPO, CALL [CLRBIN] ;CLEAR BIN IS S=0
; 6262 ENDSKP: END STATE, J/SKIP ;NO--ALL DONE
; 6263
; 6264 DBNEG: CLEAR ARXO ;CLEAR EXTRA SIGN BIT
; 6265 [ARX]_- [ARX], 3T, ;NEGATE AND SEE IF
; 6266 SKIP AD.EQ.O, AC[BINO] ; ANY CARRY
; 6267 =0 [AR]_.NOT.AC[BINO], 2T, J/STAC34 ;NO CARRY
; 6268 [AR]_-AC[BINO], 3T, ;CARRY
; 6269 SKIP AD.EQ.O ;SEE IF ALL ZERO
; 6270 =0 [ARX]_[400000] XWD 0 ;MAKE COPY OF SIGN
; 6271 ; UNLESS HIGH WORD IS ZERO
; 6272 STAC34: AC[BINO]_[AR] ;PUT BACK ANSWER
; 6273 AC[BIN1]_[ARX], J/DBDONE ; ..
; 6274
; 6275 ;HELPER SUBROUTINE TO CLEAR AC[BINO] AND AC[BIN1] IF S=0
; 6276 ;CALL WITH:
; 6277 ; READ [BRX], SKIP DPO, CALL [CLRBIN]
; 6278 ;RETURNS 1 ALWAYS
; 6279 =0
; 6280 CLRBIN: AC[BINO]_0, J/CLRB1
; 6281 RETURN [1]
; 6282 CLRB1: AC[BIN1]_0, RETURN [1]
; 6283
```

; T1OKL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 172
EXTEND -- BINARY TO DECIMAL CONVERSION

```

; 6284 .TOC "EXTEND -- BINARY TO DECIMAL CONVERSION"
; 6285
; 6286 3743:
; 6287 BDEC: [BRX]_AC[DLEN], ;GET LENGTH AND FLAGS
; 6288 SKIP FPD ;CONTINUE FROM INTERRUPT?
; 6289 =0 [BRX]_[BRX].AND.#, ;JUST KEEP THE FLAGS
; 6290 #/777000, ;
; 6291 J/BDECO ;COMPUTE NEW FLAGS
; 6292 DOCVT: [AR] AC, J/DOCVT1 ;ALL SET PRIOR TO TRAP
; 6293 BDEC: [ARX]_AC[1] ;GET LOW BINARY
; 6294 [AR] AC, SC 20. ;GET HIGH WORD, SET STEP COUNT
; 6295 =0* WORK[BDL]_[ARX], ;SAVE IN CASE OF ABORT
; 6296 CALL [CLARXO] ;MAKE SURE BIT 0 IS OFF
; 6297 WORK[BDH]_[AR], ;SAVE HIGH WORD AND
; 6298 SKIP DPO ; TEST SIGN
; 6299 =0
; 6300 BDEC1: [BRX]_O, HOLD LEFT, ;POSITIVE, CLEAR RH OF BRX
; 6301 J/BDEC3 ;COMPUTE # OF DIGITS REQUIRED
; 6302 [BRX]_[BRX].OR.#, ;NEGATIVE, SET M
; 6303 #/100000, HOLD RIGHT ;
; 6304 =0*
; 6305 BDEC2: CLEAR ARXO, CALL [DBLNG1] ;NEGATE ARIARX
; 6306 AC [AR] TEST, ;PUT BACK ANSWER
; 6307 SKIP DPO ;IF STILL MINUS WE HAVE
; 6308 ; 1BO, AND NO OTHER BITS
; 6309 =0 AC[1]_[ARX], J/BDEC1 ;POSITIVE NOW
; 6310 [ARX]_[ARX]+1 ;JUST 1BO--ADD 1
; 6311 [BRX]_[BRX].OR.#, ;AND REMEMBER THAT WE DID
; 6312 #/040000, HOLD RIGHT, ; IN LEFT HALF OF AC+3
; 6313 J/BDEC2 ; NEGATE IT AGAIN
; 6314 =0
; 6315 BDEC3: [AR] AC, J/BDEC4 ;GET HIGH AC
; 6316 [BRX]_[BRX].OR.#, ;NO LARGER POWER OF 10 FITS
; 6317 #/200000, ;SET N FLAG (CLEARLY NOT 0)
; 6318 HOLD RIGHT, J/BDEC5 ;SETUP TO FILL, ETC.
; 6319 =001
; 6320 BDEC4: [ARX]_AC[1], ;GET HIGH WORD
; 6321 CALL [BDSUB] ;SEE IF 10**C(BRX) FITS
; 6322 =011 [BRX]_[BRX]+1, ;NUMBER FITS--TRY A LARGER ONE
; 6323 STEP SC, J/BDEC3 ;UNLESS WE ARE OUT OF NUMBERS
; 6324 =111 TR [BRX], #/777777 ;ANY DIGITS REQUIRED?
; 6325 =
; 6326 =0 [BRX]_[BRX].OR.#, ;SOME DIGITS NEEDED,
; 6327 #/200000, HOLD RIGHT, ; SET N FLAG
; 6328 J/BDEC5 ;CONTINUE BELOW
; 6329 [BRX]_[BRX]+1 ;ZERO--FORCE AT LEAST 1 DIGIT
; 6330
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 6331 =0
; 6332 BDEC5: [AR]_AC[DLEN], ;GET LENGTH
; 6333 CALL [CLRFLG] ;REMOVE FLAGS FROM AR
; 6334 [BR]_O
; 6335 [BR]_[BRX], HOLD LEFT ;GET # OF DIGITS NEEDED
; 6336 [BR]_[BR]-[AR], ;NUMBER OF FILLS NEEDED
; 6337 SKIP AD.LE.O ;SEE IF ENOUGH ROOM
; 6338 =0 [ARX]_WORK[BDL], ;DOES NOT FIT IN SPACE ALLOWED
; 6339 J/BDA BT ; DO NOT DO CONVERT
; 6340 READ [BRX], SKIP DPO ;IS L ALREADY SET
; 6341 =0 AC[DLEN]_[BRX], ;NO--NO FILLERS
; 6342 J/DOCVT ;GO CHURN OUT THE NUMBER
; 6343
; 6344
; 6345 ;HERE TO STORE LEADING FILLERS
; 6346 [AR]_[BRX], HOLD RIGHT ;MAKE SURE THE FLAGS GET SET
; 6347 AC[DLEN]_[AR] ; BEFORE WE PAGE FAIL
; 6348 [AR]_WORK[EO] ;ADDRESS OF FILL (-1)
; 6349 [AR]_[AR]+1, LOAD VMA, ;FETCH FILLER
; 6350 START READ
; 6351 MEM READ, [TO]_MEM ;GET FILLER INTO AR
; 6352 STATE [EDIT-DST] ;PAGE FAILS BACKUP DST
; 6353 WORK[SLEN]_[BR]-1, 3T ;SAVE # OF FILLERS
; 6354 BDFILL: [AR]_[TO], WORK[SLEN] ;RESTORE FILL BYTE AND
; 6355 ; WARM UP RAM FILE
; 6356 [BR]_WORK[SLEN]+1, 3T, ;MORE FILLERS NEEDED?
; 6357 SKIP DPO
; 6358 =000 AC[DLEN]_[BRX], J/DOCVT ;ALL DONE FIX FLAGS AND CONVERT
; 6359 =001 WORK[SLEN]_[BR], ;SAVE UPDATED LENGTH
; 6360 CALL [PUTDST] ; AND STORE FILLER
; 6361 =111 [BR]_AC[DLEN]-1 ;COUNT DOWN STRING LENGTH
; 6362 =
; 6363 AC[DLEN]_[BR], J/BDFILL ;KEEP FILLING
; 6364
```

; T10KLMCR[10,1141] 15:34 27-JULY-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

U 3377, 3406,3771,0004,1276,6007,0701,0000,0000,1441

U 0562, 2226,0111,0705,4174,4007,0700,0010,0000,0000
U 0566, 3400,3333,0003,7174,4007,0700,0400,0000,0247

U 3400, 0636,0551,0503,7274,4003,7701,0000,0000,0241

U 0636, 2220,3333,0003,4174,4007,0700,0200,0004,0012
U 0637, 0510,3333,0004,7174,4007,0700,0400,0000,0250
U 0510, 3464,3771,0013,4370,4007,0700,0010,0000,0012
U 0516, 3401,2551,0705,1274,4007,0701,4000,0000,1443
U 3401, 3402,3771,0003,7274,4007,0701,0000,0000,0247
U 3402, 3403,3771,0004,7274,4007,0701,0000,0000,0250
U 3403, 2214,4553,0500,4374,4007,0321,0000,0004,0000
U 2214, 3410,0111,0704,4174,4007,0700,0000,0000,0000
U 2215, 3404,3440,0303,0174,4007,0700,0400,0000,0000
U 3404, 3405,3440,0404,1174,4007,0700,0400,0000,1441
U 3405, 3406,3440,0505,1174,4007,0700,0400,0000,1443
U 3406, 2216,1111,0706,4174,4007,0531,4000,0000,0000
U 2216, 0562,2441,0705,4174,4467,0701,4000,0003,0000
U 2217, 0260,4221,0013,4170,4467,0700,0000,0005,0000

U 2220, 3673,4221,0013,4170,4007,0700,0010,0000,0000
U 2221, 2222,4553,0600,4374,4007,0331,0000,0077,7777
U 2222, 0637,4221,0003,4174,0007,0700,0000,0000,0000
U 2223, 2224,4553,0600,4374,4007,0321,0000,0010,0000
U 2224, 2225,3770,0303,4344,4007,0700,0000,0000,0000
U 2225, 0637,4221,0003,4174,0007,0700,0000,0000,0000

U 3407, 1505,3771,0003,7274,4007,0701,0000,0000,0247

U 3410, 2215,5551,0505,4374,0007,0700,0000,0004,0000

; 6365 ;HERE TO STORE THE ANSWER
; 6366
; 6367 DOCVT1: [ARX]_AC[1], ;GET LOW WORD
; 6368 J/DOCVT2 ;ENTER LOOP FROM BOTTOM
; 6369 =010
; 6370 BDECLP: [BR]_[BR]+1, ;COUNT DIGITS
; 6371 CALL [BDSUB] ;KEEP SUBTRACTING 10**C(BRX)
; 6372 =110 WORK[BDH]_[AR] ;SAVE BINARY
; 6373 =
; 6374 [AR]_[BR]+WORK[E1], ;OFFSET DIGIT
; 6375 B DISP ;SEE WHICH MODE
; 6376 =1110 READ [AR], LOAD VMA, ;TRANSLATE, START READING TABLE
; 6377 START READ, J/BDTBL ; GO GET ENTRY FROM TABLE
; 6378 BDSET: WORK[BDL]_[ARX] ;SAVE LOW BINARY
; 6379 =00* STATE_[EDIT-DST], CALL [PUTDST]
; 6380 =11* [BR]_AC[DLEN]-1 ;UPDATE STRING LENGTH
; 6381 [AR]_WORK[BDH]
; 6382 [ARX]_WORK[BDL]
; 6383 TL [BR], #/040000 ;ARE WE CONVERTING 1BO?
; 6384 =0 [ARX]_[ARX]+1, J/BDCFLG ;YES--FIX THE NUMBER AND CLEAR FLAG
; 6385 DOCVT3: AC [AR]
; 6386 AC[1]_[ARX]
; 6387 AC[DLEN]_[BR] ;STORE BACK NEW STRING LENGTH
; 6388 DOCVT2: [BRX]_[BRX]-1, 3T, SKIP DP18
; 6389 =0 [BR]_-1, SET FPD, 3T, J/BDECLP
; 6390 END STATE, CLR FPD, J/SKIP
; 6391
; 6392 ;HERE TO TRANSLATE 1 DIGIT
; 6393 =0
; 6394 BDTBL: END STATE, ;DON'T CHANGE BYTE POINTER IF
; 6395 ; THIS PAGE FAILS
; 6396 CALL [LOADAR] ;GO PUT WORD IN AR
; 6397 TR [BRX], #/777777 ;LAST DIGIT
; 6398 =0 [AR]_0, HOLD RIGHT, J/BDSET
; 6399 TL [BRX], #/100000 ;AND NEGATIVE
; 6400 =0 [AR]_[AR] SWAP ;LAST AND MINUS, USE LH
; 6401 [AR]_0, HOLD RIGHT, J/BDSET
; 6402
; 6403 BDABT: [AR]_WORK[BDH], J/DAC
; 6404
; 6405 BDCFLG: [BR]_[BR].AND.NOT.#, ;CLEAR FLAG THAT TELLS US
; 6406 #/040000, HOLD RIGHT, ; TO SUBTRACT 1 AND
; 6407 J/DOCVT3 ; CONTINUE CONVERTING
; 6408

; T1OKL.MCR[10,1141] 15:34 27-JULY-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 2226, 2230,0551,0616,4374,4007,0701,0000,0000,0344
U 2227, 2675,4443,0000,4174,4007,0700,0000,0000,0000

U 2230, 2236,3441,1617,4174,4007,0700,0210,0000,0010

U 2232, 2234,1551,0404,6274,4007,0561,4000,0000,0000
U 2234, 2235,1111,0703,4174,4007,0700,4000,0000,0000
U 2235, 3411,0551,0616,4374,4007,0701,0000,0000,0373
U 3411, 3412,3333,0016,4174,4007,0700,0200,0000,0010

U 3412, 2236,1551,0303,6274,4007,0522,4000,0000,0000

U 2236, 0002,4551,0404,4374,0004,1700,0000,0037,7777
U 2237, 3413,0551,0303,6274,4007,0700,0000,0000,0000
U 3413, 3414,3333,0017,4174,4007,0700,0200,0000,0010
U 3414, 2240,0551,0404,6274,4007,0561,0000,0000,0000

U 2240, 0006,4551,0404,4374,0004,1700,0000,0037,7777

U 2241, 2240,0111,0703,4174,4007,0700,0000,0000,0000

; 6409 ;SUBROUTINE TO SUBTRACT A POWER OF 10 FROM AR!ARX
; 6410 ;CALL WITH:
; 6411 ; AR!ARX/ NUMBER TO BE CONVERTED
; 6412 ; BRX(RIGHT)/ POWER OF 10
; 6413 ;RETURNS:
; 6414 ; 2 RESULT IS STILL POSITIVE
; 6415 ; 6 RESULT WOULD HAVE BEEN NEGATIVE (RESTORE DONE)
; 6416 =0
; 6417 BDSUB: [TO]_[BRX]+#, 3T, WORK/DECLO, ;ADDRESS OF LOW WORD
; 6418 J/BDSUB1 ;NO INTERRUPT
; 6419 J/FIXPC ;INTERRUPT
; 6420 =0*
; 6421 BDSUB1: [T1]_[TO], LOAD VMA, ;PUT IN VMA,
; 6422 CALL [CLARXO] ;FIX UP SIGN OF LOW WORD
; 6423 [ARX]_[ARX]-RAM, 3T, ;SUBTRACT
; 6424 SKIP CRY1 ;SEE IF OVERFLOW
; 6425 =0 [AR]_[AR]-1 ;PROCESS CARRY
; 6426 [TO]_[BRX]+#, 3T, WORK/DECHI ;ADDRESS OF HIGH WORD
; 6427 READ [TO], LOAD VMA ;PLACE IN VMA
; 6428 [AR]_[AR]-RAM, 4T, ;SUBTRACT
; 6429 SKIP DPO ;SEE IF IT FIT
; 6430 =0
; 6431 CLARXO: CLEAR ARXO, ;IT FIT, KEEP LOW WORD +
; 6432 RETURN [2] ; AND RETURN
; 6433 [AR]_[AR]+RAM ;RESTORE
; 6434 READ [T1], LOAD VMA
; 6435 [ARX]_[ARX]+RAM, 3T, SKIP CRY1
; 6436 =0
; 6437 BDSUB2: CLEAR ARXO, ;KEEP LOW WORD +
; 6438 RETURN [6] ;RETURN OVERFLOW
; 6439 [AR]_[AR]+1, ;ADD BACK THE CARRY
; 6440 J/BDSUB2 ;COMPLETE SUBTRACT
; 6441

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 176
EXTEND -- EDIT -- MAIN LOOP

U 3741, 3472,0111,0703,4170,4007,0700,0210,0004,0012
U 3751, 2242,3771,0006,0276,6007,0700,0000,0000,0000

U 2242, 3556,4553,0600,4374,4007,0321,0010,0004,7777
U 2246, 3415,3443,0600,4174,4007,0700,0200,0004,0012
U 3415, 3416,4221,0013,4170,4007,0700,0000,0000,0000
U 3416, 2244,3770,0605,4344,4007,0700,0000,0000,0000

U 2244, 3673,0115,0505,4174,4007,0700,0010,0000,0000
U 2245, 0654,3333,0005,4174,4003,1701,0000,0000,0000

U 0654, 2250,3770,0303,4344,4007,0700,2000,0071,0007
U 0655, 2251,3770,0303,4344,4007,0700,0000,0000,0000
U 0656, 2250,3447,0303,4174,4007,0700,2000,0071,0006
U 0657, 3417,4551,0303,4374,4007,0700,0000,0000,0777

U 2250, 2250,3447,0303,4174,4007,0630,2000,0060,0000
U 2251, 3417,4551,0303,4374,4007,0700,0000,0000,0777

```
; 6442 .TOC      "EXTEND -- EDIT -- MAIN LOOP"
; 6443
; 6444 ;HERE FOR EDIT INSTRUCTION
; 6445 ;CALL WITH:
; 6446 ;   AR/      EO      ADDRESS OF FILL, FLOAT, AND MESSAGE TABLE
; 6447 ;   BR/      E1      TRANSLATE TABLE
; 6448 ;
; 6449 3741:
; 6450 EDIT:  VMA [AR]+1, START READ, ;FIRST GET FILL BYTE
; 6451      CALL [GTFILL]           ;GO GET IT
; 6452 3751:  [BRX]_AC             ;GET PATTERN POINTER
; 6453      =0** TL [BRX], #/047777, ;MAKE SURE SECTION 0
; 6454      CALL [BITCHK]           ;...
; 6455 EDITLP: VMA [BRX], START READ ;FETCH PATTERN WORD
; 6456      END STATE               ;NO SPECIAL PAGE FAIL ACTION
; 6457      [BR]_[BRX] SWAP        ;GET PBN IN BITS 20 & 21
; 6458      =0 [BR]_[BR]*4,        ;...
; 6459      CALL [LOADAR]           ;GET PATTERN WORD
; 6460      READ [BR], 3T, DISP/DP LEFT
; 6461      =1100
; 6462      [AR]_[AR] SWAP, SC_7, J/MOVPAT ;(0) BITS 0-8
; 6463      [AR]_[AR] SWAP, J/MSKPAT   ;(1) BITS 9-17
; 6464      [AR]_[AR]*.5, SC_6, J/MOVPAT ;(2) BITS 18-27
; 6465      [AR]_[AR].AND.#, #/777, J/EDISP ;(3) BITS 28-35
; 6466      =0
; 6467      MOVPAT: [AR]_[AR]*.5, STEP SC, J/MOVPAT ;SHIFT OVER
; 6468      MSKPAT: [AR]_[AR].AND.#, #/777
; 6469
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/ES6, DTN: 223-7881

; T1OKL.MCR[10,1141] 15:34 27-JULY-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 3417, 2252,3447,0305,4174,4007,0700,2000,0071,0002 ; 6470 ;HERE WITH PATTERN BYTE RIGHT ADJUSTED IN AR
; 6471 EDISP: [BR]_[AR]*.5, SC_2 ;SHIFT OVER
; 6472 =0
U 2252, 2252,3447,0505,4174,4007,0630,2000,0060,0000 ; 6473 EDISP1: [BR]_[BR]*.5, STEP SC, J/EDISP1
U 2253, 0661,3333,0005,4174,4003,5701,0000,0000,0000 ; 6474 READ [BR], 3T, DISP/DP ;LOOK AT HIGH 3 BITS
; 6475 =0001 ;(0) OPERATE GROUP
; 6476 [AR]-#, #/5, 4T, ; SEE IF O-4
; 6477 SKIP DP18, J/EDOPR ;
; 6478 ;(1) MESSAGE BYTE
; 6479 READ [BRX], SKIP DPO,
; 6480 J/EDMSG ;
; 6481 ;(2) UNDEFINED
U 0663, 2266,3333,0006,4174,4007,0520,0000,0000,0000 ; 6482 J/EDNOP ;(3) UNDEFINED
; 6483 ;(3) UNDEFINED
U 0665, 3437,4443,0000,4174,4007,0700,0000,0000,0000 ; 6484 J/EDNOP ;(4) UNDEFINED
; 6485 ;(4) UNDEFINED
U 0667, 3437,4443,0000,4174,4007,0700,0000,0000,0000 ; 6486 J/EDNOP ;(5) SKIP IF M SET
; 6487 ;(5) SKIP IF M SET
; 6488 TL [BRX], #/100000,
; 6489 J/EDSKP ;(6) SKIP IF N SET
; 6490 ;(6) SKIP IF N SET
U 0673, 2272,4553,0600,4374,4007,0321,0000,0010,0000 ; 6491 TL [BRX], #/200000,
; 6492 J/EDSKP ;(7) SKIP ALWAYS
; 6493 ;(7) SKIP ALWAYS
U 0675, 2272,4553,0600,4374,4007,0321,0000,0020,0000 ; 6494 J/EDSKP
; 6495
; 6496 .TOC "EXTEND -- EDIT -- DECODE OPERATE GROUP"
; 6497
; 6498 ;HERE FOR OPERATE GROUP. SKIP IF IN RANGE
; 6499 =0
U 2254, 3437,4443,0000,4174,4007,0700,0000,0000,0000 ; 6500 EDOPR: J/EDNOP ;OUT OF RANGE
U 2255, 0710,3333,0003,4174,4003,5701,0000,0000,0000 ; 6501 READ [AR], 3T, DISP/DP ;DISPATCH ON TYPE
U 0710, 3420,0111,0701,4174,4007,0700,0000,0000,0000 ; 6502 =1000 [PC]_[PC]+1, J/EDSTOP ;(0) STOP EDIT
; 6503 STATE [EDIT-SRC], ;(1) SELECT SOURCE BYTE
; 6504 J/EDSEL
; 6505 READ [BRX], SKIP DPO, ;(2) START SIGNIFICANCE
; 6506 J/EDSSIG
; 6507 [BRX]_[BRX].AND.#, ;(3) FIELD SEPERATOR
; 6508 #/77777, HOLD RIGHT,
; 6509 J/EDNOP
U 0711, 2231,3771,0013,4370,4007,0700,0000,0000,0011 ; 6510 [BR]_AC[MARK] ;(4) EXCHANGE MARK AND DEST
U 0712, 0246,3333,0006,4174,4007,0520,0000,0000,0000 ; 6511 VMA_[BR], START READ,
; 6512 J/EDEXMD
; 6513 =
; 6514
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES6, DTN: 223-7881

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 178
EXTEND -- EDIT -- STOP EDIT

U 3420, 3421,7441,0605,4174,4007,0700,1000,0071,0010
U 3421, 3422,3441,0603,4174,4007,0701,1000,0043,0000
U 3422, 2256,4553,0500,4374,4007,0321,0000,0003,0000
U 2256, 1515,3770,0303,4334,4017,0700,0000,0041,0000
U 2257, 2256,0111,0703,4174,4007,0700,1000,0051,0700

U 0246, 3426,4443,0000,4174,4007,0700,0010,0000,0000
U 0247, 3437,4443,0000,4174,4007,0700,0000,0000,0000

U 2260, 3673,3772,0000,1275,5007,0701,0010,0000,1444
U 2261, 3423,4443,0000,4174,4007,0700,0200,0003,0002

U 3423, 3424,3223,0000,4174,4007,0701,0200,0000,0002
U 3424, 3437,3440,0303,1174,4007,0700,0400,0000,1444

```
; 6515 .TOC "EXTEND -- EDIT -- STOP EDIT"  
; 6516  
; 6517 ;HERE TO END AN EDIT OPERATION. PC IS SET TO SKIP IF NORMAL END  
; 6518 ; OR NON-SKIP IF ABORT  
; 6519 EDSTOP: [BR]_NOT.[BRX], ;AD WILL NOT DO D.AND.NOT.A  
; 6520 FE_S#, S#/10 ;PRESET FE  
; 6521 [AR]_[BRX], 3T, FE_FE+P ;MOVE POINTER, UPBATE PBN  
; 6522 [BR]_AND.#, 3T, ;WAS OLD NUMBER 3?  
; 6523 #/030000, SKIP ADL.EQ.O ; ..  
; 6524 =0  
; 6525 EDSTP1: [AR]_P, J/STAC ;NO--ALL DONE  
; 6526 [AR]_[AR]+1, ;YES--BUMP WORD #  
; 6527 FE_FE.AND.S#, S#/0700, ;KEEP ONLY FLAG BITS  
; 6528 J/EDSTP1 ;GO STOP EDIT  
; 6529  
; 6530 .TOC "EXTEND -- EDIT -- START SIGNIFICANCE"  
; 6531  
; 6532 ;HERE WITH DST POINTER IN AR  
; 6533 =110  
; 6534 EDSSIG: CALL [EDFLT] ;STORE FLT CHAR  
; 6535 J/EDNOP ;DO NEXT PATTERN BYTE  
; 6536  
; 6537 .TOC "EXTEND -- EDIT -- EXCHANGE MARK AND DESTINATION"  
; 6538  
; 6539 ;HERE WITH ADDRESS OF MARK POINTER IN BR  
; 6540 =0  
; 6541 EDEXMD: Q_AC[DSTP], ;GET DEST POINTER  
; 6542 CALL [LOADAR] ;GO PUT MARK IN AR  
; 6543 START WRITE ;START WRITE. SEPERATE STEP TO AVOID  
; 6544 ; PROBLEM ON DPM5  
; 6545 MEM WRITE, MEM_Q ;PUT OLD DEST IN MARK  
; 6546 AC[DSTP]_[AR], J/EDNOP ;PUT BACK DEST POINTER  
; 6547
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 6548 .TOC "EXTEND -- EDIT -- PROCESS SOURCE BYTE"
; 6549
; 6550 =O*
; 6551 EDSEL: [AR]_AC[SRCP], ;PICK UP SRC POINTER
; 6552 CALL [GETSRC] ;GET SOURCE BYTE
U 2231, 2313,3771,0003,1276,6007,0701,0010,0000,1441 ; 6553 [AR]_[AR]*.5, WORK[E1] ;PREPARE TO TRANSLATE
U 2233, 0700,3447,0303,7174,4007,0700,0000,0000,0241 ; 6554 =O00 [AR]_[AR]+WORK[E1], ;GO TRANSLATE BY HALFWORDS
; 6555 2T, CALL [TRNAR] ; ..
U 0700, 3457,0551,0303,7274,4007,0700,0010,0000,0241 ; 6556 =O10
; 6557 EDFILL: READ [AR], ;(2) NO SIGNIFICANCE, GO FILL
; 6558 SKIP AD.EQ.O, ; SEE IF ANY FILLER
; 6559 J/EDFIL1 ; GO TO IT
U 0702, 2262,3333,0003,4174,4007,0621,0000,0000,0000 ; 6560 STATE [EDIT-SRC], ;(3) SIG START, DO FLOAT CHAR
; 6561 J/EDSFLT
U 0703, 0606,3771,0013,4370,4007,0700,0000,0000,0011 ; 6562 =100 ;(4) ABORT
U 0704, 3420,4443,0000,4174,4007,0700,0000,0000,0000 ; 6563 =101
; 6564 EDSPUT: STATE [EDIT-S+D], ;(5) NORMAL, STORE AT DST
; 6565 CALL [PUTDST] ; ..
U 0705, 3464,3771,0013,4370,4007,0700,0010,0000,0013 ; 6566 =111
; 6567 J/EDNOP ;(7) BYTE STORED
; 6568 =
; 6569
; 6570 ;HERE TO COMPLETE STORING FILL
; 6571 =O
U 2262, 0705,4443,0000,4174,4007,0700,0000,0000,0000 ; 6572 EDFIL1: J/EDSPUT ;STORE FILLER
U 2263, 3437,4443,0000,4174,4007,0700,0000,0000,0000 ; 6573 J/EDNOP ;NO FILLER TO STORE
; 6574
; 6575 ;HERE TO DO FLOAT BYTE
; 6576 =110
; 6577 EDSFLT: WORK[FSIG] [ARX], ;SAVE SIG CHAR
; 6578 CALL [EDFLT] ;STORE FLOAT CHAR
; 6579 [AR]_WORK[FSIG] ;RESTORE CHAR
U 0606, 3426,3333,0004,7174,4007,0700,0410,0000,0246 ; 6580 [AR]_[AR].AND.# CLR LH, ;JUST KEEP THE BYTE IN CASE
U 0607, 3425,3771,0003,7274,4007,0701,0000,0000,0246 ; 6581 #/77777, ; DEST BYTE .GT. 15 BITS
; 6582 J/EDSPUT ;GO STORE CHAR WHICH STARTED THIS AL
; 6583
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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 3426, 3427,3771,0005,1276,6007,0701,0000,0000,1443
U 3427, 3430,3443,0500,4174,4007,0700,0200,0003,0012
U 3430, 3431,3771,0005,1276,6007,0701,0000,0000,1444
U 3431, 2264,3333,0005,4175,5007,0701,0200,0000,0002

U 2264, 3433,4751,1203,4374,4007,0700,0010,0000,0002
U 2265, 0740,3771,0003,4365,5007,0621,0200,0000,0002

U 0740, 3464,3551,1313,4370,4007,0700,0010,0000,0012

U 0741, 3432,3551,0606,4374,0007,0700,0000,0040,0000

U 0746, 3432,3551,0606,4374,0007,0700,0000,0040,0000

U 3432, 0007,3440,0606,0174,4004,1700,0400,0000,0000

U 2266, 0760,3771,0003,7274,4007,0622,0000,0000,0244

U 2267, 2270,4251,0303,4374,4007,0700,0000,0000,0077

U 2270, 3433,0111,0703,7174,4007,0700,0010,0000,0240

U 2271, 0760,3771,0003,4365,5007,0700,0200,0000,0002

U 0760, 3464,3771,0013,4370,4007,0700,0010,0000,0012

U 0761, 3437,4443,0000,4174,4007,0700,0000,0000,0000

U 0766, 3437,4443,0000,4174,4007,0700,0000,0000,0000

U 3433, 3434,0551,0303,7274,4007,0701,0000,0000,0240

U 3434, 0001,3443,0300,4174,4004,1700,0200,0004,0012

```
; 6584 ;SUBROUTINE TO PROCESS FLOAT CHAR
; 6585 ;CALL WITH:
; 6586 ; AR/ POINTER TO STORE @ MARK
; 6587 ;RETURN 7 WITH FLOAT STORED
; 6588 EDFLT: [BR]_AC[MARK] ;ADDRESS OF MARK POINTER
; 6589 VMA [BR], START WRITE ;READY TO STORE
; 6590 [BR]_AC[DSTP] ;GET DST POINTER
; 6591 MEM WRITE, MEM [BR] ;STORE POINTER
; 6592 =0 [AR]_O XWD [2], ;FETCH FLOAT CHAR
; 6593 CALL [EDBYTE] ;GET TBL BYTE
; 6594 MEM READ, [AR]_MEM, ;GET FLOAT CHAR
; 6595 SKIP AD.EQ.O ;SEE IF NULL
; 6596 =000
; 6597 [FLG]_[FLG].OR.#, ;REMEMBER TO BACKUP DST POINTER
; 6598 STATE/EDIT-DST, ; WILL ALSO BACKUP SRC IF CALLED
; 6599 HOLD LEFT, ; FROM SELECT
; 6600 CALL [PUTDST] ; STORE FLOAT
; 6601 =001 [BRX]_[BRX].OR.#, #/400000,
; 6602 HOLD RIGHT, J/EDFLT1 ;NULL
; 6603 =110 [BRX]_[BRX].OR.#, #/400000,
; 6604 HOLD RIGHT, J/EDFLT1 ;MARK STORED
; 6605 =
; 6606 EDFLT1: AC_[BRX], ;SAVE FLAGS SO WE DON'T
; 6607 ;TRY TO DO THIS AGAIN IF
; 6608 ;NEXT STORE PAGE FAILS
; 6609 RETURN [7] ;AND RETURN
; 6610
; 6611 .TOC "EXTEND -- EDIT -- MESSAGE BYTE"
; 6612
; 6613 ;HERE WITH SKIP ON S
; 6614 =0
; 6615 EDMSG: [AR]_WORK[FILL], ;GET FILL BYTE
; 6616 SKIP AD.EQ.O, 4T, ;SEE IF NULL
; 6617 J/EDMSG1 ;GO STORE
; 6618 [AR]_[AR].AND.# CLR LH, ;GET OFFSET INTO TABLE
; 6619 #/77
; 6620 =0 [AR]_[AR]+1, WORK[EO], ;PLUS 1
; 6621 CALL [EDBYTE] ;GET TBL BYTE
; 6622 MEM READ, [AR]_MEM ;FROM MEMORY
; 6623 =000
; 6624 EDMSG1: STATE [EDIT-DST], ;WHAT TO DO ON PAGE FAILS
; 6625 CALL [PUTDST] ;STORE MESSAGE BYTE
; 6626 =001 J/EDNOP ;NULL FILLER
; 6627 =110 J/EDNOP ;NEXT BYTE
; 6628 =
; 6629
; 6630 EDBYTE: [AR]_[AR]+WORK[EO] ;GET OFFSET INTO TABLE
; 6631 VMA [AR], START READ, ;START MEMORY CYCLE
; 6632 RETURN [1] ;RETURN TO CALLER
; 6633
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 181
EXTEND -- EDIT -- SKIP

U 2272, 3435,4551,0303,4374,4007,0700,0000,0000,0077

U 2273, 3437,4443,0000,4174,4007,0700,0000,0000,0000
U 3435, 3436,0115,0703,4174,4007,0700,0000,0000,0000

U 3436, 3440,3333,0003,4174,4007,0701,2000,0007,0000

U 3437, 3440,4443,0000,4174,4007,0700,2000,0071,0000

U 3440, 3441,3333,0006,4174,4007,0701,1000,0073,0000

U 3441, 3442,4443,0000,4174,4007,0700,1000,0051,0030

U 3442, 3443,4443,0000,4174,4007,0700,1000,0040,0000

U 3443, 3444,4443,0000,4174,4007,0700,1000,0041,0010

U 3444, 3445,3777,0003,4334,4057,0700,2000,0041,0000

U 3445, 2274,4251,0303,4374,4007,0630,0000,0000,0170

U 2274, 2276,3447,0303,4174,4007,0700,2000,0071,0000

U 2275, 2274,3551,0303,4370,4007,0700,0000,0000,0200

U 2276, 2276,3447,0303,4174,4007,0630,2000,0060,0000

U 2277, 3446,0111,0306,4170,4007,0700,0000,0000,0000

U 3446, 3447,3770,0303,4334,4017,0700,0000,0041,0000

U 3447, 3450,4551,0606,4374,0007,0700,0000,0070,0000

U 3450, 3451,4551,0303,4374,4007,0700,0000,0003,0000

U 3451, 3452,3111,0306,4174,0007,0700,0000,0000,0000

U 3452, 2246,3440,0606,0174,4007,0700,0400,0000,0000

```
; 6634 .TOC "EXTEND -- EDIT -- SKIP"
; 6635
; 6636 =0
; 6637 ;HERE TO SKIP ALWAYS
; 6638 EDSKP: [AR][AR].AND.#, #/77, ;JUST KEEP SKIP DISTANCE
; 6639 J/EDSKP1 ;CONTINUE BELOW
; 6640 ;HERE IF WE DO NOT WANT TO SKIP
; 6641 J/EDNOP
; 6642 EDSKP1: [AR]_([AR]+1)*2 ;GIVE 1 EXTRA SKIP
; 6643 READ [AR], SCAD/A*2, ;PUT THE ADJUSTMENT
; 6644 SCADA/BYTE5, 3T, LOAD SC, ; THE SC
; 6645 J/EDNOP1 ;JOIN MAIN LOOP
; 6646
; 6647
; 6648 .TOC "EXTEND -- EDIT -- ADVANCE PATTERN POINTER"
; 6649
; 6650 EDNOP: SC_O ;NO SKIP
; 6651 EDNOP1: READ [BRX], 3T, FE_P ;PUT PBN IN FE
; 6652 FE_FE.AND.S#, S#/30 ;JUST BYTE #
; 6653 FE_FE+SC ;ADD IN ANY SKIP DISTANCE
; 6654 FE_FE+S#, S#/10 ;BUMP PBN
; 6655 [AR]_FE, ;GET NUMBER OF WORDS
; 6656 LOAD SC ;PUT MSB WHERE IT CAN BE TESTED
; 6657 ; QUICKLY
; 6658 [AR][AR].AND.# CLR LH, ;KEEP ONLY 1 COPY
; 6659 #/170, SKIP/SC ; ..
; 6660 =0
; 6661 EDN1A: [AR][AR]*.5, SC_O,
; 6662 J/EDNOP2 ;READY TO SHIFT OFF BYTE WITHIN
; 6663 ; WORD
; 6664 [AR][AR].OR.#, #/200, ;GET THE SIGN BIT OF THE FE
; 6665 HOLD LEFT, ; INTO THE AR. ONLY HAPPENS ON
; 6666 J/EDN1A ; SKP 76 OR SKP 77
; 6667 =0
; 6668 EDNOP2: [AR][AR]*.5, STEP SC, J/EDNOP2
; 6669 [BRX]_[BRX]+[AR], ;UPDATE WORD ADDRESS
; 6670 HOLD LEFT
; 6671 [AR] P ;PUT PBN BACK IN BRX
; 6672 [BRX]_[BRX].AND.#, ;JUST KEEP FLAGS
; 6673 #/700000, ; ..
; 6674 HOLD RIGHT
; 6675 [AR][AR].AND.#, ;JUST KEEP PBN
; 6676 #/030000
; 6677 [BRX]_[BRX].OR.[AR], ;FINAL ANSWER
; 6678 HOLD RIGHT
; 6679 AC_[BRX], J/EDITLP ;DO NEXT FUNCTION
; 6680
```

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 182
EXTEND SUBROUTINES -- FILL OUT DESTINATION

```
; 6681 .TOC "EXTEND SUBROUTINES -- FILL OUT DESTINATION"  
; 6682  
; 6683 ;CALL WITH  
; 6684 ; AC[DLEN]/ NEGATIVE NUMBER OF BYTES LEFT IN DEST  
; 6685 ; FILL/ FILL BYTE  
; 6686 ; RETURN [2] WITH FILLERS STORED  
; 6687 ;  
; 6688 ;NOTE: THIS ROUTINE NEED NOT TEST FOR INTERRUPTS ON EACH BYTE  
; 6689 ; BECAUSE EVERY BYTE STORE DOES A MEMORY READ.  
; 6690 ;  
; 6691 =01*  
; 6692 MOVF1: [AR]_WORK[FILL], 2T, ;GET FILL BYTE  
; 6693 CALL [PUTDST] ;PLACE IN DEST  
; 6694 [AR]_AC[DLEN] ;AMOUNT LEFT  
; 6695 AC[DLEN]_[AR]+1, 3T, ;STORE UPDATED LEN  
; 6696 SKIP DPO ; AND SEE IF DONE  
; 6697 =0 RETURN [2] ;DONE  
; 6698 MOVFIL: WORK[FILL], J/MOVF1 ;DO ANOTHER BYTE  
; 6699 ;ENTERING HERE SAVES 15ONS  
; 6700 ; PER BYTE BUT COSTS 30ONS  
; 6701 ; PER FIELD MOVED. I ASSUME (BUT DO  
; 6702 ; NOT KNOW) THAT THIS SPEEDS  
; 6703 ; THINGS UP.  
; 6704
```

U 0322, 3464,3771,0003,7274,4007,0700,0010,0000,0244
U 0326, 3453,3771,0003,1276,6007,0701,0000,0000,1443
U 3453, 2300,0113,0703,1174,4007,0521,0400,0000,1443
U 2300, 0002,4443,0000,4174,4004,1700,0000,0000,0000
U 2301, 0322,4443,0000,7174,4007,0700,0000,0000,0244

; T10KL.MCR[10,1141] 15:34 27-JULY-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

```
; 6705 .TOC"EXTEND SUBROUTINES -- GET MODIFIED SOURCE BYTE"  
; 6706  
; 6707 ;CALL WITH:  
; 6708 ;SLEN = MINUS LENGTH OF STRING  
; 6709 ;MSK = MASK FOR BYTE SIZE (1 IF BIT MUST BE ZERO)  
; 6710 ;E1 = EFFECTIVE ADDRESS OF OPERATION WORD (SIGN EXTENDED IF OFFSET)  
; 6711 ; [AR]_WORK[SLEN]+1, CALL [SRCMOD]  
; 6712 ;RETURNS:  
; 6713 ; 1 LENGTH EXHAUSTED  
; 6714 ; 2 (EDIT ONLY) NO SIGNIFICANCE  
; 6715 ; 3 (EDIT ONLY) SIGNIFICANCE START:  
; 6716 ; 4 ABORT: OUT OF RANGE OR TRANSLATE FAILURE  
; 6717 ; 5 NORMAL: BYTE IN AR  
; 6718 ;  
; 6719 ;DROM B SET AS FOLLOWS:  
; 6720 ; 0 TRANSLATE  
; 6721 ; 1 OFFSET  
; 6722 ; 2 EDIT  
; 6723 ; 4 CVTDBT  
; 6724 =00  
; 6725 SRCMOD: WORK[SLEN]_[AR], ;PUT BACK SOURCE LENGTH  
; 6726 SKIP DPO, ;SEE IF DONE  
; 6727 CALL [GSRC] ;GET A SOURCE BYTE  
; 6728 END STATE, RETURN [1] ;DONE  
; 6729 WORK[E1], B DISP ;OFFSET OR TRANSLATE?  
; 6730 =  
; 6731 =1110 [AR]_[AR]*.5, J/XLATE ;TRANSLATE  
; 6732 FIX [AR] SIGN, WORK[E1] ;IF WE ARE PROCESSING FULL WORD  
; 6733 ; BYTES, AND THEY ARE NEGATIVE,  
; 6734 ; AND THE OFFSET IS POSITIVE THEN  
; 6735 ; WE HAVE TO MAKE BITS -1 AND -2  
; 6736 ; COPIES OF THE SIGN BIT.  
; 6737 [AR]_[AR]+WORK[E1], 2T ;OFFSET  
; 6738 [AR].AND.WORK[MSK], ;VALID BYTE?  
; 6739 SKIP AD.EQ.O, 4T, ;SKIP IF OK  
; 6740 RETURN [4] ;RETURN 4 IF BAD, 5 IF OK  
; 6741
```

```
U 1114, 2312,3333,0003,7174,4007,0520,0410,0000,0242  
U 1115, 0001,4221,0013,4170,4004,1700,0000,0000,0000  
U 1116, 0716,4443,0000,7174,4003,7700,0000,0000,0241  
  
U 0716, 3456,3447,0303,4174,4007,0700,0000,0000,0000  
U 0717, 3454,3770,0303,7174,0007,0700,0000,0000,0241  
  
U 3454, 3455,0551,0303,7274,4007,0700,0000,0000,0241  
  
U 3455, 0004,4553,0300,7274,4004,1622,0000,0000,0243
```


; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 184
EXTEND SUBROUTINES -- TRANSLATE

U 3456, 3457,0551,0303,7274,4007,0701,0000,0000,0241

U 3457, 2302,3333,0003,4174,4007,0700,0200,0004,0012

U 2302, 3674,3445,0303,4174,4007,0700,0010,0000,0000

U 2303, 2304,4553,0300,4374,4007,0331,0000,0000,0001

U 2304, 0721,3441,0403,4174,4003,1701,0000,0000,0000

U 2305, 2304,3770,0404,4344,4007,0700,0000,0000,0000

```
; 6742 .TOC "EXTEND SUBROUTINES -- TRANSLATE"  
; 6743  
; 6744 ;HERE WITH BYTE IN AR 1-36. FETCH TABLE ENTRY.  
; 6745 XLATE: [AR]_[AR]+WORK[E1] ;COMPUTE ADDRESS  
; 6746 TRNAR: READ [AR], LOAD VMA, ;FETCH WORD  
; 6747 START READ ;  
; 6748 =0 [AR]_[AR]*2, ;GET BACK LSB  
; 6749 ;BIT 36 IS NOT PRESERVED  
; 6750 ; BY PAGE FAILS  
; 6751 CALL [LOADARX] ;PUT ENTRY IN ARX  
; 6752 TR [AR], #/1 ;WHICH HALF?  
; 6753 =0  
; 6754 XLATE1: [AR]_[ARX], 3T, ;RH -- COPY TO AR  
; 6755 DISP/DP LEFT, ;DISPATCH ON CODE  
; 6756 J/TRNFNC ;DISPATCH TABLE  
; 6757 [ARX]_[ARX] SWAP, ;LH -- FLIP AROUND  
; 6758 J/XLATE1 ;START SHIFT  
; 6759
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 185
EXTEND SUBROUTINES -- TRANSLATE

```

; 6760 ; HERE ON TRANSLATE OPERATION TO PERFORM FUNCTIONS REQUIRED BY
; 6761 ; THE 3 HIGH ORDER BITS OF THE TRANSLATE FUNCTION HALFWORD. WE
; 6762 ; DISPATCH ON FUNCTION AND HAVE:
; 6763 ; BRX/   FLAGS
; 6764 ; ARX/   TABLE ENTRY IN RH
; 6765 ;
; 6766 =0001
; 6767
; 6768 TRNFNC: READ [BRX], SKIP DPO, ;(0) NOP
; 6769 J/TRNRET ;S FLAG ALREADY SET?
; 6770 ;
; 6771 RETURN [4] ;(1) ABORT
; 6772
; 6773 [BRX]_[BRX].AND.NOT.#, ;(2) CLEAR M FLAG
; 6774 #/100000, HOLD RIGHT,
; 6775 J/TRNFNC
; 6776 ;(3) SET M FLAG
; 6777 [BRX]_[BRX].OR.#,
; 6778 #/100000, HOLD RIGHT,
; 6779 J/TRNFNC ;(4) SET N FLAG
; 6780
; 6781 TRNSIG: [BRX]_[BRX].OR.#,
; 6782 #/200000, HOLD RIGHT,
; 6783 J/TRNFNC ;(5) SET N FLAG THEN ABORT
; 6784
; 6785 [BRX]_[BRX].OR.#,
; 6786 #/200000, HOLD RIGHT,
; 6787 RETURN [4] ;(6) CLEAR M THEN SET N
; 6788
; 6789 [BRX]_[BRX].AND.NOT.#,
; 6790 #/100000, HOLD RIGHT,
; 6791 J/TRNSIG ;(7) SET N AND M
; 6792
; 6793 [BRX]_[BRX].OR.#,
; 6794 #/300000, HOLD RIGHT,
; 6795 J/TRNFNC
; 6796
```

U 0721, 2306,3333,0006,4174,4007,0520,0000,0000,0000

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

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; EXTEND.MIC[10,1141] 11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 186
EXTEND SUBROUTINES -- TRANSLATE

U 2306, 0754, 3333, 0004, 4174, 4003, 7530, 0000, 0000, 0000
U 2307, 0005, 4251, 0403, 4374, 4004, 1700, 0000, 0007, 7777

U 0754, 0533, 3771, 0003, 1276, 6003, 7701, 0000, 0000, 1443

U 0755, 2307, 3551, 0606, 4374, 0007, 0700, 0000, 0040, 0000
U 0756, 0002, 3771, 0003, 7274, 4004, 1701, 0000, 0000, 0244
U 0757, 0003, 3771, 0003, 1276, 6004, 1701, 0000, 0000, 1444

U 0533, 3460, 1111, 0703, 4174, 4007, 0700, 4000, 0000, 0000
U 0537, 1114, 0551, 0703, 7274, 4007, 0701, 0000, 0000, 0242
U 3460, 2310, 3770, 0303, 1174, 4007, 0520, 0400, 0000, 1443
U 2310, 1114, 3771, 0003, 7274, 4007, 0701, 0000, 0000, 0242
U 2311, 1114, 0551, 0703, 7274, 4007, 0701, 0000, 0000, 0242

```
; 6797 ;HERE TO COMPLETE A TRANSLATE
; 6798
; 6799 =0
; 6800 TRNRET: READ [ARX], SKIP DP18, ;S-FLAG IS ZERO
; 6801 B DISP, SKIP DP18, ;SEE IF EDIT OR SIG START
; 6802 J/TRNSS ; ..
; 6803 TRNSS1: [AR]_[ARX].AND.# CLR LH, ;S IS SET, JUST RETURN BYTE
; 6804 ;/77777, RETURN [5] ; ..
; 6805
; 6806 =1100
; 6807 TRNSS: [AR]_AC[DLEN], ;NO SIG ON MOVE OR D2B
; 6808 B DISP, J/TRNSS1 ;SEE IF D2B
; 6809 [BRX]_[BRX].OR.#, ;SIG START ON MOVE OR D2B
; 6810 ;/400000, HOLD RIGHT,
; 6811 J/TRNSS1 ;RETURN BYTE
; 6812 [AR]_WORK[FILL], ;EDIT--NO SIG RETURN FILL
; 6813 RETURN [2] ; ..
; 6814 [AR]_AC[DSTP], ;EDIT--START OF SIG
; 6815 RETURN [3] ; ..
; 6816
; 6817 =1011
; 6818 TRNNS1: [AR]_[AR]-1, J/TRNNS2 ;COMPENSATE FOR IGNORING SRC
; 6819 [AR]_WORK[SLEN]+1, ;DEC TO BIN HAS NO DEST LENGTH
; 6820 J/SRCMOD ;JUST UPDATE SRC LENTH
; 6821 TRNNS2: AC[DLEN]_[AR] TEST, ;PUT BACK DLEN AND
; 6822 SKIP DPO ; SEE WHICH IS NOW SHORTER
; 6823 =0 [AR]_WORK[SLEN], ;DEST IS SHORTER. DO NOT CHANGE
; 6824 J/SRCMOD ; AMOUNT LEFT
; 6825 [AR]_WORK[SLEN]+1, ;GO LOOK AT NEXT BYTE
; 6826 J/SRCMOD
; 6827
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 6828 .TOC "EXTEND SUBROUTINES -- GET UNMODIFIED SOURCE BYTE"  
; 6829  
; 6830 ;CALL:  
; 6831 ; GSRC WITH SKIP ON SOURCE LENGTH  
; 6832 ; GETSRC IF LENGHT IS OK  
; 6833 ;WITH:  
; 6834 ; AC1/ SOURCE BYTE POINTER  
; 6835 ;RETURNS:  
; 6836 ; 1 IF LENGTH RAN OUT  
; 6837 ; 2 IF OK (BYTE IN AR)  
; 6838 ;  
; 6839 =0  
; 6840 GSRC: [AR]_AC[DLEN], ;LENGTH RAN OUT  
; 6841 RETURN [1] ;RESTORE AR AND RETURN  
; 6842 GETSRC: [AR]_AC[SRCP] ;GET SRC PTR  
; 6843 IBP DP, IBP SCAD, ;UPDATE BYTE POINTER  
; 6844 SCAD DISP, 3T ;SEE IF OFLOW  
; 6845 =01 [AR]_[BR], J/GSRC1 ;NO OFLOW  
; 6846 SET P TO 36-S ;RESET P  
; 6847 [AR]_[AR]+1, HOLD LEFT ;BUMP Y  
; 6848  
; 6849 GSRC1: AC[SRCP]_[AR] ;STORE UPDATED POINTER  
; 6850 =0 READ [AR], LOAD BYTE EA, ;SETUP TO FIGURE OUT  
; 6851 FE_P, 3T, CALL [BYTEAS] ; EFFECTIVE ADDRESS  
; 6852 READ [AR], ;LOOK AT POINTER  
; 6853 BYTE DISP, ;SEE IF 7 BIT  
; 6854 FE FE.AND.S#, S#/0770, ;MASK OUT P FIELD  
; 6855 J/_LDB1 ;GO GET THE BYTE  
; 6856
```

```
U 2312, 0001,3771,0003,1276,6004,1701,0000,0000,1443  
U 2313, 3461,3771,0003,1276,6007,0701,0000,0000,1441  
  
U 3461, 0231,3770,0305,4334,4016,7701,0000,0033,6000  
U 0231, 3463,3441,0503,4174,4007,0700,0000,0000,0000  
U 0233, 3462,3770,0503,4334,4017,0700,0000,0032,6000  
U 3462, 3463,0111,0703,4170,4007,0700,0000,0000,0000  
  
U 3463, 2314,3440,0303,1174,4007,0700,0400,0000,1441  
  
U 2314, 3071,3333,0003,4174,4217,0701,1010,0073,0500  
  
U 2315, 0340,3333,0003,4174,4006,5701,1000,0051,0770
```

; T10KL.MCR[10,1141]
; EXTEND.MIC[10,1141]

15:34 27-JULY-1984
11:35 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 188
EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING

```
; 6857 .TOC "EXTEND SUBROUTINES -- STORE BYTE IN DESTINATION STRING"  
; 6858  
; 6859 ;CALL WITH:  
; 6860 ; AR/ BYTE TO STORE  
; 6861 ; AC4/ DESTINATION BYTE POINTER  
; 6862 ;RETURNS:  
; 6863 ; AR & AC4/ UPDATED BYTE POINTER  
; 6864 ; ARX/ BYTE TO STORE  
; 6865 ; BR/ WORD TO MERGE WITH  
; 6866 ; 6 ALWAYS  
; 6867 ;  
; 6868 PUTDST: [AR]_[AR] ;SAVE BYTE  
; 6869 =0 [AR]_AC[DSTP], ;GET DEST POINTER  
; 6870 CALL [IDST] ;BUMP DEST POINTER  
; 6871 AD/A+B, A/ARX, B/ARX, ;SHIFT 7-BIT BYTE TO  
; 6872 SCAD/A, 3T, ; NATURAL PLACE, AND PUT  
; 6873 SCADA/BYTES, LOAD FE ; INTO FE  
; 6874 =0* READ [AR], BYTE DISP, ;GO PUT BYTE IN MEMORY  
; 6875 CALL [DPB1] ; ..  
; 6876 RETURN [6] ;ALL DONE  
; 6877
```

U 3464, 2316,3441,0304,4174,4007,0700,0000,0000,0000

U 2316, 3465,3771,0003,1276,6007,0701,0010,0000,1444

U 2317, 2320,0113,0404,4174,4007,0701,1000,0077,0000

U 2320, 0360,3333,0003,4174,4006,5701,0010,0000,0000

U 2322, 0006,4443,0000,4174,4004,1700,0000,0000,0000

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES6, DTN: 223-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-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

U 3465, 2321,3770,0305,4334,4016,7701,0000,0033,6000
U 2321, 3467,3441,0503,4174,4217,0700,0000,0000,0600
U 2323, 3466,3770,0503,4334,4017,0700,0000,0032,6000
U 3466, 3467,0111,0703,4170,4217,0700,0000,0000,0600

U 3467, 0230,3440,0303,1174,4006,6701,1400,0073,1444

U 0230, 3074,0553,0300,2274,4007,0701,0200,0004,0712
U 0232, 3074,3443,0300,4174,4007,0700,0200,0004,0712
U 0234, 3470,0553,0300,2274,4007,0701,0200,0004,0612
U 0236, 3470,3443,0300,4174,4007,0700,0200,0004,0612

U 3470, 3471,3771,0003,4361,5217,0700,0200,0000,0602
U 3471, 0230,4443,0000,2174,4006,6700,0000,0000,0000

U 3556, 2732,4551,0202,4374,0007,0700,0000,0077,7740
U 3557, 0004,4443,0000,4174,4004,1700,0000,0000,0000

U 3472, 3473,3771,0003,4365,5007,0700,0200,0000,0002

U 3473, 0010,3333,0003,7174,4004,1700,0400,0000,0244

U 3474, 0001,4551,0303,4374,0004,1700,0000,0000,0777

```
; 6878 .TOC "EXTEND SUBROUTINES -- UPDATE DEST STRING POINTERS"
; 6879
; 6880
; 6881 ;SUBROUTINE TO BUMP DST POINTERS
; 6882 ;CALL WITH:
; 6883 ; AR/ AC[DSTP]
; 6884 ; RETURN 1 WITH UPDATED POINTER STORED
; 6885 ;
; 6886 IDST: IBP DP, IBP SCAD, SCAD DISP, 3T
; 6887 =0* [AR][BR], LOAD DST EA, J/IDSTX
; 6888 SET P TO 36-S
; 6889 [AR][AR]+1, HOLD LEFT, LOAD DST EA
; 6890 IDSTX: AC[DSTP][AR], 3T, ;STORE PTR BACK
; 6891 FE_P, DISP/EAMODE ;SAVE P FOR CMPDST
; 6892 =100*
; 6893 DSTEА: VMA[AR]+XR, START READ, PXCT BYTE DATA, 3T, J/BYTFET
; 6894 VMA[AR], START READ, PXCT BYTE DATA, J/BYTFET
; 6895 VMA[AR]+XR, START READ, PXCT/BIS-DST-EA, 3T, J/DSTIND
; 6896 VMA[AR], START READ, PXCT/BIS-DST-EA, J/DSTIND
; 6897
; 6898 DSTIND: MEM READ, [AR]_MEM, HOLD LEFT, LOAD DST EA
; 6899 EA MODE DISP, J/DSTEА
; 6900
; 6901
; 6902 ;HERE TO TEST ILLEGAL BITS SET
; 6903 ;CALL WITH:
; 6904 ; SKIP IF ALL BITS LEGAL
; 6905 ; RETURN [4] IF OK, ELSE DO UUU
; 6906 ;
; 6907 3556: ;EXTEND OF 0 COMES HERE
; 6908 BITCHK: UUU
; 6909 3557: RETURN [4]
; 6910
; 6911 ;HERE TO PUT FILL IN [AR] AND WORK[FILL]
; 6912 GTFILL: MEM READ, ;WAIT FOR DATA
; 6913 [AR]_MEM ;PLACE IN AR
; 6914 WORK[FILL][AR], ;SAVE FOR LATER
; 6915 RETURN [10] ;RETURN TO CALLER
; 6916
; 6917 ;SUBROUTINE TO CLEAR FLAGS IN AR
; 6918 CLRFLG: [AR][AR].AND.#, ;CLEAR FLAGS IN AR
; 6919 #/000777. ;
; 6920 HOLD RIGHT, RETURN [1]
; 6921
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-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

```
; 6922 .TOC "EXTEND -- PAGE FAIL CLEANUP"
; 6923
; 6924 ;BACK UP SOURCE POINTER
; 6925 =0
; 6926 BACKS: [AR]_AC[SRCP],
; 6927 CALL [BACKBP] ;BACKUP BP
; 6928 AC[SRCP]_[BR], J/CLDISP
; 6929
; 6930 CMSDST: [AR]_WORK[SV.BRX] ;GET OLD SRC LEN
; 6931 AC_[AR]+1, 3T ;BACK UP
; 6932 ;BACK UP DESTINATION POINTER
; 6933 =0
; 6934 BACKD: [AR]_AC[DSTP],
; 6935 CALL [BACKBP]
; 6936 AC[DSTP]_[BR], J/CLDISP
; 6937
; 6938 ;FAILURES DURING MOVE STRING (BACKUP LENGTHS)
; 6939 STRPF: [AR]_-WORK[SLEN] ;GET AMOUNT LEFT
; 6940 STRPFO: [BR]_AC[DLEN], 4T, ;WHICH STRING IS LONGER?
; 6941 SKIP_DPO
; 6942 =0
; 6943 STRPF1: AC[DLEN]_[AR], J/STPF1A ;SRC LONGER
; 6944 [ARX]_[AR] ;COPY SRC LENGTH
; 6945 =0 [ARX]_[ARX].OR.WORK[SV.BRX], ;REBUILD FLAGS
; 6946 CALL [AC_ARX] ;RESET AC[SLEN]
; 6947 [AR]_[AR]-[BR] ;MAKE DEST LEN
; 6948 STRPF3: AC[DLEN]_[AR], ;PUT BACK DEST LEN
; 6949 J/CLDISP ;DO NEXT CLEANUP
; 6950
; 6951 STPF1A: [AR]_[AR]+[BR], J/STPF2
; 6952
; 6953 PFDBIN: [AR]_-WORK[SLEN] ;RESTORE LENGTH
; 6954 STRPF2: [AR]_[AR].OR.WORK[SV.BRX]
; 6955 PFGACO: AC_[AR], J/CLDISP ;PUT BACK SRC LEN AND FLAGS
; 6956
; 6957 STRPF4: [AR]_.NOT.WORK[SLEN], J/STRPFO
; 6958
; 6959 BACKBP: IBP DP, SCAD/A+B, SCADA/BYTE1, SCADB/SIZE, ;P_P+S
; 6960 RETURN [1]
; 6961

U 2324, 3507,3771,0003,1276,6007,0701,0010,0000,1441
U 2325, 2676,3440,0505,1174,4007,0700,0400,0000,1441

U 3475, 3476,3771,0003,7274,4007,0701,0000,0000,0214
U 3476, 2326,0113,0703,0174,4007,0701,0400,0000,0000

U 2326, 3507,3771,0003,1276,6007,0701,0010,0000,1444
U 2327, 2676,3440,0505,1174,4007,0700,0400,0000,1444

U 3477, 3500,1771,0003,7274,4007,0701,4000,0000,0242

U 3500, 2330,3771,0005,1276,6007,0522,0000,0000,1443

U 2330, 3502,3440,0303,1174,4007,0700,0400,0000,1443
U 2331, 2332,3441,0304,4174,4007,0700,0000,0000,0000

U 2332, 3704,3551,0404,7274,4007,0701,0010,0000,0214
U 2333, 3501,1111,0503,4174,4007,0700,4000,0000,0000

U 3501, 2676,3440,0303,1174,4007,0700,0400,0000,1443

U 3502, 3504,0111,0503,4174,4007,0700,0000,0000,0000

U 3503, 3504,1771,0003,7274,4007,0701,4000,0000,0242
U 3504, 3505,3551,0303,7274,4007,0701,0000,0000,0214
U 3505, 2676,3440,0303,0174,4007,0700,0400,0000,0000

U 3506, 3500,7771,0003,7274,4007,0701,0000,0000,0242

U 3507, 0001,3770,0305,4334,4014,1700,0000,0043,6000
```

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 191
TRAPS

```

; 6962          .TOC      "TRAPS"
; 6963
U 3510, 3511,3741,0104,4074,4007,0700,0000,0000,0000 ; 6964 TRAP:  [ARX]_PC WITH FLAGS      ;SAVE THE PC WHICH CAUSED THE
; 6965          WORK[TRAPPC]_[ARX],      ; TRAP
U 3511, 2334,3333,0004,7174,4007,0340,0400,0000,0425 ; 6966          SKIP KERNEL          ;SEE IF UBR OR EBR
; 6967          =0      [AR]_[AR]+[UBR],      ;ADDRESS OF INSTRUCTION
; 6968          MEM READ,          ;WAIT FOR PREFETCH TO GET INTO
; 6969          ; THE CACHE. MAY PAGE FAIL BUT
; 6970          ; THAT IS OK
; 6971          START READ,          ;START FETCH
; 6972          VMA PHYSICAL,        ;ABSOLUTE ADDRESSING
U 2334, 3512,0111,1103,4364,4007,0700,0200,0024,1016 ; 6973          J/TRP1          ;JOIN COMMON CODE
; 6974
; 6975          [AR]_[AR]+[EBR],      ;WE COME HERE IN EXEC MODE
; 6976          MEM READ,          ;WAIT FOR PREFETCH TO GET INTO
; 6977          ; THE CACHE. MAY PAGE FAIL BUT
; 6978          ; THAT IS OK
; 6979          START READ,          ;START FETCH
; 6980          VMA PHYSICAL,        ;ABSOLUTE ADDRESSING
U 2335, 3512,0111,1003,4364,4007,0700,0200,0024,1016 ; 6981          J/TRP1          ;JOIN COMMON CODE
; 6982
; 6983          TRP1:  MEM READ, [HR]_MEM,      ;PLACE INSTRUCTION IN HR
; 6984          LOAD INST          ;LOAD IR, XR, @
; 6985          [HR].AND.#,          ;TEST TO SEE IF THIS
; 6986          #/700000, 3T,          ; IS A UO
; 6987          SKIP ADL.EQ.O
; 6988          =0      CHANGE FLAGS,          ;NOT A UO
; 6989          HOLD USER/1,          ;CLEAR TRAP FLAGS
; 6990          J/XCT1          ;DO THE INSTRUCTION
U 2336, 2711,4443,0000,4174,4467,0700,0000,0001,0000 ; 6991          UO          ;DO THE UO
U 2337, 2732,4551,0202,4374,0007,0700,0000,0077,7740 ; 6992
; 6992
```


; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KSIO MICROCODE V124, 27-JUL-84 Page 192
IO -- INTERNAL DEVICES

```

; 6993          .TOC      "IO -- INTERNAL DEVICES"
; 6994
; 6995          .DCODE
; 6996 700:      IOT,AC DISP,      J/GRP700
; 6997          IOT,AC DISP,      J/GRP701
; 6998          .UCODE
; 6999
; 7000 1701:    UUU                ;DATAI APR,
; 7001 1702:    UUU                ;BLKO APR,
; 7002 1703:    UUU                ;DATAO APR,
; 7003 1706:    [BR]_APR, J/APRSZ ;CONSZ APR,
; 7004 1707:    [BR]_APR, J/APRSO ;CONSO APR,
; 7005 1710:
; 7006 RDERA:    UUU                ;BLKI PI,
; 7007 1711:    UUU                ;DATAI PI,
; 7008 1712:    UUU                ;BLKO PI,
; 7009 1713:    UUU                ;DATAO PI,
; 7010 1716:    [BR]_[PI], J/CONSZ ;CONSZ PI,
; 7011 1717:    [BR]_[PI], J/CONSO ;CONSO PI,
; 7012
; 7013 1720:
; 7014 GRP701:  UUU                ;BLKI PAG,
; 7015 1726:    UUU                ;CONSZ PAG,
; 7016 1727:    UUU                ;CONSO PAG,
; 7017
; 7018 ;680I AND CACHE SWEEP STUFF
; 7019 1730:    UUU
; 7020 1731:    UUU
; 7021 1732:    UUU
; 7022 1733:    UUU
; 7023 1734:    UUU
; 7024 1735:    UUU
; 7025 1736:    UUU
; 7026 1737:    UUU
; 7027
; 7028 APRSO:    [BR]_[BR].AND.# CLR LH, #/7770
; 7029 CONSO:    [BR].AND.[AR], SKIP ADR.EQ.O, J/SKIP
; 7030
; 7031 APRSZ:    [BR]_[BR].AND.# CLR LH, #/7770
; 7032 CONSZ:    [BR].AND.[AR], SKIP ADR.EQ.O, J/DONE,
; 7033
```

D 0700, 1200, 1700, 4100
D 0701, 1200, 1720, 4100

U 1701, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1702, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1703, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1706, 3516, 3771, 0005, 4304, 4007, 0701, 0000, 0000, 0000
U 1707, 3514, 3771, 0005, 4304, 4007, 0701, 0000, 0000, 0000

U 1710, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1711, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1712, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1713, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1716, 3517, 3441, 1405, 4174, 4007, 0700, 0000, 0000, 0000
U 1717, 3515, 3441, 1405, 4174, 4007, 0700, 0000, 0000, 0000

U 1720, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1726, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1727, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1730, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1731, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1732, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1733, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1734, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1735, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1736, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740
U 1737, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 3514, 3515, 4251, 0505, 4374, 4007, 0700, 0000, 0000, 7770
U 3515, 0260, 4113, 0305, 4174, 4007, 0330, 0000, 0000, 0000

U 3516, 3517, 4251, 0505, 4374, 4007, 0700, 0000, 0000, 7770
U 3517, 1400, 4113, 0305, 4174, 4007, 0330, 0000, 0000, 0000

MICROCODE RELEASE (MINOR)/UCR,
MICROCODE RELEASE (MINOR)/UCR

; FIXED by HWE STD
; NOT PART of wave
; for ID only

Produced on Advanced Information Services Electronic Laser Printer. PK01E56, DTN: 221-7181

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 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

```
; 7034 1700:  
; 7035 GRP700:  
; 7036 APRID: [BR]_#,  
; 7037 #/4097.  
; 7038 137: [BR]_#,  
; 7039 MICROCODE OPTION(INHCST)/OPT,  
; 7040 MICROCODE OPTION(NOCST)/OPT,  
; 7041 MICROCODE OPTION(NONSTD)/OPT,  
; 7042 MICROCODE OPTION(UBABL)/OPT,  
; 7043 MICROCODE OPTION(KIPAGE)/OPT,  
; 7044 MICROCODE OPTION(KLPAGE)/OPT,  
; 7045 MICROCODE VERSION/UCV,  
; 7046 HOLD RIGHT,  
; 7047 J/RTNREG  
; 7048  
; 7049
```

U 0137, 3626,3771,0005,4374,0007,0700,0000,0055,0124

U 1704, 3520,3771,0005,7274,4007,0701,0000,0000,0230

```
1704:  
7050 WRAPR: [BR]_WORK[APR]  
; 7051 [BR]_[BR].AND.NOT.#, ;CLEAR THE OLD PIA  
; 7052 #/7, HOLD LEFT ;  
; 7053 [ARX]_[AR].AND.#, #/7 ;PUT NEW PIA IN ARX  
; 7054 [BR]_[BR].OR.[ARX] ;PUT NEW PIA IN BR  
; 7055 [ARX]_[AR].AND.#, ;MASK THE DATA BITS  
; 7056 #/007760 ; DOWN TO ENABLES  
; 7057 TR [AR], #/100000 ;WANT TO ENABLE ANY?  
; 7058 =0 [BR]_[BR].OR.[ARX] ;YES--SET THEM  
; 7059 TR [AR], #/40000 ;WANT TO DISABLE ANY?  
; 7060 =0 [BR]_[BR].AND.NOT.[ARX] ;YES--CLEAR THEM  
; 7061 [BRX]_APR ;GET CURRENT STATUS  
; 7062 TR [AR], #/20000 ;WANT TO CLEAR FLAGS?  
; 7063 =0 [BRX]_[BRX].AND.NOT.[ARX] ;YES--CLEAR BITS  
; 7064 TR [AR], #/10000 ;WANT TO SET ANY FLAGS?  
; 7065 =0 [BRX]_[BRX].OR.[ARX] ;YES--SET FLAGS  
; 7066 TR [AR], #/30000 ;ANY CHANGE AT ALL?  
; 7067 =0 READ [BRX], ;YES--LOAD NEW FLAGS  
; 7068 J/WRAPR2 ;TURN OFF INTERRUPT 8080  
; 7069 WRAPR1: READ [BR], ;FIX DPM TIMING BUG  
; 7070 READ [BR], ;ENABLE CONDITIONS  
; 7071 SET APR ENABLES  
; 7072 WORK[APR]_[BR], ;SAVE FOR RDAPR  
; 7073 J/DONE ;ALL DONE  
; 7074  
; 7075 WRAPR2: READ [BRX], ;LOAD NEW FLAGS  
; 7076 SPEC/APR FLAGS ;  
; 7077 [BRX]_[BRX].AND.NOT.#, ;CLEAR INTERRUPT THE 8080  
; 7078 #/002000, HOLD LEFT ; FLAG  
; 7079 READ [BRX], ;LOAD NEW FLAGS  
; 7080 SPEC/APR FLAGS, ;  
; 7081 J/WRAPR1 ;LOOP BACK  
; 7082
```

U 3520, 3521,5551,0505,4370,4007,0700,0000,0000,0007

U 3521, 3522,4551,0304,4374,4007,0700,0000,0000,0007

U 3522, 3523,3111,0405,4174,4007,0700,0000,0000,0000

U 3523, 3524,4551,0304,4374,4007,0700,0000,0000,7760

U 3524, 2340,4553,0300,4374,4007,0331,0000,0010,0000

U 2340, 2341,3111,0405,4174,4007,0700,0000,0000,0000

U 2341, 2342,4553,0300,4374,4007,0331,0000,0004,0000

U 2342, 2343,5111,0405,4174,4007,0700,0000,0000,0000

U 2343, 3525,3771,0006,4304,4007,0701,0000,0000,0000

U 3525, 2344,4553,0300,4374,4007,0331,0000,0002,0000

U 2344, 2345,5111,0406,4174,4007,0700,0000,0000,0000

U 2345, 2346,4553,0300,4374,4007,0331,0000,0001,0000

U 2346, 2347,3111,0406,4174,4007,0700,0000,0000,0000

U 2347, 2350,4553,0300,4374,4007,0331,0000,0003,0000

U 2350, 3530,3333,0006,4174,4007,0700,0000,0000,0000

U 2351, 3526,3333,0005,4174,4007,0700,0000,0000,0000

U 3526, 3527,3333,0005,4174,4257,0700,0000,0000,0000

U 3527, 1400,3333,0005,7174,4007,0700,0400,0000,0230

U 3530, 3531,3333,0006,4174,4237,0700,0000,0000,0000

U 3531, 3532,5551,0606,4370,4007,0700,0000,0000,2000

U 3532, 2351,3333,0006,4174,4237,0700,0000,0000,0000

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 194
IO -- INTERNAL DEVICES

```
U 1705, 3533,3771,0005,7274,4007,0701,0000,0000,0230 ; 7083 1705:
; 7084 RDAPR: [BR]_WORK[APR]
; 7085 [BR]_[BR] SWAP, ;PUT ENABLES IN BOTH
; 7086 HOLD RIGHT ; HALVES
U 3533, 3534,3770,0505,4344,0007,0700,0000,0000,0000 ; 7087 [BR]_[BR].AND.#, ;SAVE ENABLES IN LH
; 7088 #/7760, ;
; 7089 HOLD RIGHT
U 3534, 3535,4551,0505,4374,0007,0700,0000,0000,7760 ; 7090 [BR]_[BR].AND.#, ;SAVE PIA IN RH
; 7091 #/7;
; 7092 HOLD LEFT
U 3535, 3536,4551,0505,4370,4007,0700,0000,0000,0007 ; 7093 [ARX]_APR ;READ THE APR FLAGS
U 3536, 3537,3771,0004,4304,4007,0701,0000,0000,0000 ; 7094 [ARX]_[ARX].AND.# CLR LH, ;MASK OUT JUNK
; 7095 #/007770 ;KEEP 8 FLAGS
U 3537, 3540,4251,0404,4374,4007,0700,0000,0000,7770 ; 7096 [BR]_[BR].OR.[ARX], ;MASH THE STUFF TOGETHER
; 7097 J/RTNREG ;RETURN
; 7098
```

Produced on Advanced Information Services Electronic Laser Printer, PKC1/ES5, DTN: 233-7381

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 195
IO -- INTERNAL DEVICES -- EBR & UBR

```

; 7099 .TOC "IO -- INTERNAL DEVICES -- EBR & UBR"
; 7100
; 7101 1723:
U 1723, 3541,3443,0300,4174,4007,0700,0200,0004,0012 ; 7102 WRUBR: VMA [AR], ;LOAD E INTO VMA
; 7103 START READ ;START MEMORY
; 7104 MEM READ, ;WAIT FOR DATA
; 7105 [AR]_MEM, 3T, ;PUT IT INTO THE AR
; 7106 SKIP_DPO ;SEE IF WE WANT TO LOAD
; 7107 ; AC BLOCK NUMBERS
; 7108 =0 [AR]_[AR].AND.#, ;NO--CLEAR JUNK IN AR
; 7109 #/100000, ; LEAVE ONLY LOAD UBR
; 7110 HOLD RIGHT, ; IN LEFT HALF
; 7111 SKIP ADL.EQ.O, 3T, ;SEE IF WE WANT TO LOAD
; 7112 J/ACBSET ;SKIP AROUND UBR LOAD
; 7113 [UBR]_[UBR].AND.#, ;MASK OUT THE OLD
; 7114 #/770077, ; AC BLOCK NUMBERS
; 7115 HOLD RIGHT ;IN THE LEFT HALF
; 7116 [AR].AND.#, ;SEE IF WE WANT TO LOAD
; 7117 #/100000, 3T, ; UBR ALSO
; 7118 SKIP ADL.EQ.O
; 7119 =0
; 7120 ACBSET: [BR]_[AR].AND.#, ;COPY UBR PAGE NUMBER
; 7121 #/17777, ; INTO BR
; 7122 J/SETUBR ;GO LOAD UBR
; 7123 [UBR]_[UBR].OR.[AR], ;DO NOT LOAD UBR
; 7124 ; PUT AC BLOCK # IN
; 7125 HOLD RIGHT, ; THE LEFT HALF
; 7126 LOAD AC BLOCKS, ;LOAD HARDWARE
; 7127 J/DONE ;ALL DONE
; 7128 SETUBR: [BR]_O, ;CLEAR BR LEFT
; 7129 SC_7, ;PUT THE COUNT IN SC
; 7130 HOLD RIGHT
; 7131 =0
; 7132 STUBRS: [BR]_[BR]*2, ;SHIFT BR OVER
; 7133 STEP SC, ; 9 PLACES
; 7134 J/STUBRS
; 7135 [UBR]_[UBR].AND.#, ;MASK OUT OLD UBR
; 7136 #/777774, ; BITS IN
; 7137 HOLD RIGHT ; LEFT HALF
; 7138 [UBR]_O, ;CLEAR RIGHT HALF
; 7139 HOLD LEFT
; 7140 [UBR]_[UBR].OR.[BR] ;PUT IN PAGE TABLE ADDRESS
; 7141 [UBR]_[UBR].OR.[AR], ;PUT IN AC BLOCK #
; 7142 HOLD RIGHT, ; IN LEFT HALF
; 7143 LOAD AC BLOCKS, ;TELL HARDWARE
; 7144 J/SWEEP ;CLEAR CACHE
; 7145
```

; T10KL.MCR[10,1141] 15:34 27-JULY-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, 2360, 3445, 0303, 4174, 4007, 0700, 2000, 0071, 0006
U 2360, 2360, 3445, 0303, 4174, 4007, 0630, 2000, 0060, 0000
U 2361, 3547, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0230
U 3547, 3550, 4551, 0505, 4370, 4007, 0700, 0000, 0074, 7777
U 3550, 2362, 4553, 0300, 4374, 4007, 0321, 0000, 0000, 0020
U 2362, 2363, 3551, 0505, 4370, 4007, 0700, 0000, 0003, 0000
HAPPEN
U 2363, 3551, 3333, 0005, 4174, 4257, 0700, 0000, 0000, 0000
U 3551, 3552, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0230
NABLE
ERNAL FLAG
BIT 0
U 3552, 2426, 3441, 0310, 4174, 4007, 0700, 0000, 0000, 0000
U 1725, 2364, 3447, 1005, 4174, 4007, 0700, 2000, 0071, 0006
U 2364, 2364, 3447, 0505, 4174, 4007, 0630, 2000, 0060, 0000
U 2365, 3553, 4551, 0505, 4374, 4007, 0700, 0000, 0006, 3777
U 3553, 3626, 4221, 0005, 4174, 0007, 0700, 0000, 0000, 0000

; 7146 1724:
; 7147 WREBR: [AR]_[AR]*2, SC_6 ;DO A SHIFT OVER 8 MORE
; 7148 =0
; 7149 WREBR1: [AR]_[AR]*2, STEP SC, J/WREBR1 ;SKIP WHEN = -1
; 7150 .IF/FULL ;DO NOT ENABLE PAGING IN SMALL
; 7151 ; MICROCODE.
; 7152 [BR]_WORK[APR]
; 7153 [BR]_[BR].AND.#, #/747777, HOLD LEFT
; 7154 [AR].AND.#, #/20, 3T, SKIP ADL.EQ.0 ;BIT 22 - TRAP ENABL
; 7155 =0 [BR]_[BR].OR.#, #/030000, HOLD LEFT ;SET - ALLOW TRAPS T
; 7156 READ [BR], SET APR ENABLES
; 7157 WORK[APR]_[BR]
; 7158 .ENDIF/FULL
; 7159
;:7160 .IF/KIPAGE
;:7161 .IF/KLPAGE
;:7162 [EBR]_[AR] ;NOTE: SHIFTED LEFT 9 BITS
;:7163 [EBR].AND.#, #/40, 3T, SKIP ADL.EQ.0 ;BIT 21 - KL PAGING
;:7164 =0 [EBR]_[EBR].OR.#, #/400000, HOLD RIGHT, J/SWEEP ;YES, SET IN
;:7165 [EBR]_[EBR].AND.NOT.#, #/400000, HOLD RIGHT, J/SWEEP ;NO, CL
;:7166 .ENDIF/KLPAGE
; 7167 .ENDIF/KIPAGE
; 7168
;:7169 .IFNOT/KLPAGE ;MUST BE KI ONLY
;:7170 [EBR]_[AR].J/SWEEP ;SO INTERNAL FLAG ISN'T USED
; 7171 .ENDIF/KLPAGE
; 7172
; 7173 .IFNOT/KIPAGE ;MUST BE KL ONLY
; 7174 [EBR]_[AR].J/SWEEP ;SO INTERNAL FLAG ISN'T USED
; 7175 .ENDIF/KIPAGE
; 7176
; 7177 1725:
; 7178 RDEBR: [BR]_[EBR]*.5, SC_6
; 7179 =0
; 7180 RDEBR1: [BR]_[BR]*.5, STEP SC, J/RDEBR1
; 7181 [BR]_[BR].AND.#, #/63777 ;MASK TO JUST EBR
; 7182 [BR]_0, ;CLEAR LEFT HALF
; 7183 HOLD RIGHT, ;BITS
; 7184 J/RTNREG ;RETURN ANSWER
; 7185

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 197
IO -- INTERNAL DEVICES -- EBR & UBR

U 1721, 2366,3441,1105,4174,4007,0700,0000,0000,0000
U 2366, 2370,3447,0506,4174,4007,0700,2010,0071,0006

U 2367, 3626,4551,0505,4374,0007,0700,0000,0050,7700

U 3554, 3555,3441,1105,4174,4007,0700,0000,0000,0000
U 3555, 2370,3447,0506,4174,4007,0700,2000,0071,0006

U 2370, 2370,3447,0606,4174,4007,0630,2000,0060,0000
U 2371, 3560,4551,0606,4374,4007,0700,0000,0001,7777
U 3560, 0001,3441,0605,4170,4004,1700,0000,0000,0000

```
; 7186 1721:  
; 7187 RDUBR: [BR]_[UBR]  
; 7188 =0 [BRX]_[BR]*.5, SC_6, CALL [GTPCW1]  
; 7189 [BR]_[BR].AND.#, ;JUST RETURN USEFUL  
; 7190 #/507700, HOLD RIGHT, ; BITS  
; 7191 J/RTNREG  
; 7192  
; 7193  
; 7194 GETPCW: [BR]_[UBR]  
; 7195 [BRX]_[BR]*.5, SC_6  
; 7196 =0  
; 7197 GTPCW1: [BRX]_[BRX]*.5, STEP SC, J/GTPCW1  
; 7198 [BRX]_[BRX].AND.#, #/17777  
; 7199 [BR]_[BRX], HOLD LEFT, RETURN [1]  
; 7200
```

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-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

U 1760, 3626, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0215

U 1761, 3626, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0216

U 1762, 3626, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0220

U 1763, 3626, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0217

U 1766, 3626, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0227

U 1767, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

U 1770, 3561, 4443, 0000, 4174, 4007, 0700, 0200, 0004, 0002

U 3561, 3562, 3771, 0003, 4365, 5007, 0700, 0200, 0000, 0002

U 3562, 1400, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0215

U 1771, 3563, 4443, 0000, 4174, 4007, 0700, 0200, 0004, 0002

U 3563, 3564, 3771, 0003, 4365, 5007, 0700, 0200, 0000, 0002

U 3564, 1400, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0216

U 1772, 3565, 4443, 0000, 4174, 4007, 0700, 0200, 0004, 0002

U 3565, 3566, 3771, 0003, 4365, 5007, 0700, 0200, 0000, 0002

U 3566, 1400, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0220

U 1773, 3567, 4443, 0000, 4174, 4007, 0700, 0200, 0004, 0002

U 3567, 3570, 3771, 0003, 4365, 5007, 0700, 0200, 0000, 0002

U 3570, 1400, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0217

U 1776, 3571, 4443, 0000, 4174, 4007, 0700, 0200, 0004, 0002

U 3571, 3572, 3771, 0003, 4365, 5007, 0700, 0200, 0000, 0002

U 3572, 1400, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0227

U 1777, 2732, 4551, 0202, 4374, 0007, 0700, 0000, 0077, 7740

; 7201 .TOC "IO -- INTERNAL DEVICES -- KL PAGING REGISTERS"
; 7202
; 7203 .DCODE
; 7204 702: IOT, AC DISP, M, J/GRP702
; 7205 .UCODE
; 7206
; 7207 1760:
; 7208 GRP702:
; 7209 RDSPB: [BR]_WORK[SBR], J/RTNREG
; 7210 1761:
; 7211 RDCSB: [BR]_WORK[CBR], J/RTNREG
; 7212 1762:
; 7213 RDPUR: [BR]_WORK[PUR], J/RTNREG
; 7214 1763:
; 7215 RDCSTM: [BR]_WORK[CSTM], J/RTNREG
; 7216 1766:
; 7217 RDHSB: [BR]_WORK[HSBADR], J/RTNREG
; 7218 1767: UUU
; 7219
; 7220 1770:
; 7221 WRSPB: START READ
; 7222 MEM READ, [AR]_MEM
; 7223 WORK[SBR]_[AR], J/DONE
; 7224 1771:
; 7225 WRCSB: START READ
; 7226 MEM READ, [AR]_MEM
; 7227 WORK[CBR]_[AR], J/DONE
; 7228 1772:
; 7229 WRPUR: START READ
; 7230 MEM READ, [AR]_MEM
; 7231 WORK[PUR]_[AR], J/DONE
; 7232 1773:
; 7233 WRCSTM: START READ
; 7234 MEM READ, [AR]_MEM
; 7235 WORK[CSTM]_[AR], J/DONE
; 7236 1776:
; 7237 WRHSB: START READ
; 7238 MEM READ, [AR]_MEM
; 7239 WORK[HSBADR]_[AR], J/DONE
; 7240 1777: UUU
; 7241

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 199
IO -- INTERNAL DEVICES -- TIMER CONTROL

```

; 7242 .TOC "IO -- INTERNAL DEVICES -- TIMER CONTROL"
; 7243
; 7244
; 7245 ;BEGIN [123]
; 7246 TICK: [AR]_WORK[TIME1], ;GET LOW WORD
; 7247 SPEC/CLRCLK ;CLEAR CLOCK FLAG
; 7248 ;END [123]
U 3573, 3574,3771,0003,7274,4117,0701,0000,0000,0301 ; 7249 TOCK: [BR]_O XWD [10000] ;2^12 UNITS PER MS
; 7250 [AR]_[AR]+[BR] ;INCREMENT THE TIMER
U 3574, 3575,4751,1205,4374,4007,0700,0000,0001,0000 ; 7251 FIX [AR] SIGN, SKIP DPO ;SEE IF IT OVERFLOWED
U 3575, 3576,0111,0503,4174,4007,0700,0000,0000,0000 ; 7252 =0
U 3576, 2372,3770,0303,4174,0007,0520,0000,0000,0000 ; 7253 TOCK1: WORK[TIME1]_[AR], ;STORE THE NEW TIME
; 7254 J/TOCK2 ;SKIP OVER THE OVERFLOW CODE
U 2372, 3577,3333,0003,7174,4007,0700,0400,0000,0301 ; 7255 [AR]_WORK[TIME0] ;GET HIGH WORD
U 2373, 2374,3771,0003,7274,4007,0701,0000,0000,0300 ; 7256 =0* [AR]_[AR]+1. ;BUMP IT
; 7257 CALL [WRTIM1] ;STORE BACK IN RAM
U 2374, 3605,0111,0703,4174,4007,0700,0010,0000,0000 ; 7258 [AR]_O, ;CAUSE LOW WORD WORD
; 7259 J/TOCK1 ; TO GET STORED
U 2376, 2372,4221,0003,4174,4007,0700,0000,0000,0000 ; 7260 TOCK2: [AR]_WORK[TTG]
U 3577, 3600,3771,0003,7274,4007,0701,0000,0000,0303 ; 7261 [AR]_[AR]-[BR], ;COUNT DOWN TIME TO GO
; 7262 SKIP AD.LE.0 ;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]
U 2400, 0002,3333,0003,7174,4004,1700,0400,0000,0303 ; 7267 [BR]_APR ;GET CURRENT FLAGS
U 2401, 3601,3771,0003,7274,4007,0701,0000,0000,0302 ; 7268 [BR]_[BR].OR.#, #/40 ;SET TIMER INTERRUPT FLAG
U 3601, 3602,3771,0005,4304,4007,0701,0000,0000,0000 ; 7269 READ [BR], ;PLACE ON DP AND
U 3602, 3603,3551,0505,4374,4007,0700,0000,0000,0040 ; 7270 SPEC/APR FLAGS, ;LOAD INTO HARDWARE
; 7271 J/TOCK3 ;ALL DONE
; 7272
```


; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 200
IO -- INTERNAL DEVICES -- WRTIME & RDTIME

U 1774, 3604,4443,0000,4174,4007,0700,0200,0004,0002
U 3604, 1120,3771,0003,4365,5007,0700,0200,0000,0002
U 1120, 3674,0111,0702,4170,4007,0700,0210,0004,0012
U 1121, 3605,4551,0404,4370,4007,0700,0010,0077,0000
U 1123, 1400,3333,0004,7174,4007,0700,0400,0000,0301
U 3605, 0002,3333,0003,7174,4004,1700,0400,0000,0300
U 1764, 3606,4451,1205,4324,4007,0700,0000,0000,0000
U 3606, 3607,4451,1204,4324,4007,0700,0000,0000,0000
U 3607, 3610,4451,1206,4324,4007,0700,0000,0000,0000
U 3610, 2402,6113,0405,4174,4007,0621,0000,0000,0000
U 2402, 2403,3441,0604,4174,4007,0700,0000,0000,0000
U 2403, 3611,3771,0005,7274,4007,0701,0000,0000,0300
U 3611, 1124,0551,0404,7274,4007,0671,0000,0000,0301
U 1124, 3574,3771,0003,7274,4117,0700,0010,0000,0301
U 1125, 3612,3333,0002,4174,4007,0700,0200,0003,0012
U 1126, 1764,4443,0000,4174,4007,0700,0000,0000,0000
U 3612, 3613,3333,0005,4175,5007,0701,0200,0000,0002
U 3613, 3614,0111,0702,4170,4007,0700,0200,0003,0012
U 3614, 1400,3333,0004,4175,5007,0701,0200,0000,0002

```
; 7273 .TOC "IO -- INTERNAL DEVICES -- WRTIME & RDTIME"  
; 7274  
; 7275 1774:  
; 7276 WRTIME: START READ ;FETCH WORD AT E  
; 7277 MEM READ, ;WAIT FOR DATA  
; 7278 [AR]_MEM ;PUT WORD IN AR  
; 7279 =00 VMA [HR]+1, ;BUMP E  
; 7280 START READ, ;START MEMORY  
; 7281 CALL [LOADARX] ;PUT DATA IN ARX  
; 7282 [ARX]_[ARX].AND.#, ;CLEAR PART HELD IN  
; 7283 #/770000, ; HARDWARE COUNTER  
; 7284 HOLD LEFT, CALL [WRTIM1]  
; 7285 =11 WORK[TIME1]_[ARX], ;IN WORK SPACE  
; 7286 J/DONE ;NEXT INSTRUCTION  
; 7287 =  
; 7288 WRTIM1: WORK[TIME0]_[AR], ;SAVE THE NEW VALUE  
; 7289 RETURN [2]  
; 7290  
; 7291 1764:  
; 7292 RDTIME: [BR]_TIME ;READ THE TIME  
; 7293 [ARX]_TIME ; AGAIN  
; 7294 [BRX]_TIME ; AGAIN  
; 7295 [BR].XOR.[ARX], ;SEE IF STABLE  
; 7296 SKIP AD.EQ.O ; ..  
; 7297 =0 [ARX]_[BRX] ;NO THEN NEXT TRY MUST BE OK  
; 7298 [BR]_WORK[TIME0]  
; 7299 [ARX]_[ARX]+WORK[TIME1], ;COMBINE PARTS  
; 7300 SKIP/-1 MS ;SEE IF OVERFLOW HAPPENED  
; 7301 =00 SPEC/CLRCLK, ;CLEAR CLOCK FLAG  
; 7302 [AR]_WORK[TIME1], 2T, ;GET LOW WORD FOR TOCK  
; 7303 CALL [TOCK] ;UPDATE CLOCKS  
; 7304 READ [HR], LOAD VMA, ;DID NOT OVERFLOW  
; 7305 START WRITE, J/RDTIM1 ;STORE ANSWER  
; 7306 J/RDTIME ;TRY AGAIN  
; 7307 =  
; 7308 RDTIM1: MEM WRITE, MEM_[BR]  
; 7309 VMA [HR]+1, LOAD VMA, START WRITE  
; 7310 MEM WRITE, MEM_[ARX], J/DONE  
; 7311
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 201
IO -- INTERNAL DEVICES -- WRINT & RDINT

; 7312 .TOC "IO -- INTERNAL DEVICES -- WRINT & RDINT"
; 7313
; 7314
; 7315 1775:
; 7316 WRINT: START READ
; 7317 MEM READ, [AR]_MEM
; 7318 WORK[PERIOD]_[AR]
; 7319 WORK[TTG]_[AR],
; 7320 J/DONE
; 7321
; 7322 1765:
; 7323 RDINT: [BR]_WORK[PERIOD],
; 7324 J/RTNREG
; 7325

U 1775, 3615,4443,0000,4174,4007,0700,0200,0004,0002
U 3615, 3616,3771,0003,4365,5007,0700,0200,0000,0002
U 3616, 3617,3333,0003,7174,4007,0700,0400,0000,0302
U 3617, 1400,3333,0003,7174,4007,0700,0400,0000,0303
U 1765, 3626,3771,0005,7274,4007,0701,0000,0000,0302

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 202
IO -- INTERNAL DEVICES -- RDPI & WRPI

```

; 7326 .TOC "IO -- INTERNAL DEVICES -- RDPI & WRPI"
; 7327
; 7328 1715:
; 7329 RDPI: [BR]_[PI], J/RTNREG
; 7330
; 7331 1714:
; 7332 WRPI: TR [AR], PI.CLR/1
; 7333 =0 [PI]_O
; 7334 TR [AR], PI.MBZ/17
; 7335 =0 UUU
; 7336 [BR]_[AR].AND.#,#/177
; 7337 [BR]_[BR] SWAP, HOLD RIGHT
; 7338 TR [AR], PI.DIR/1
; 7339 =0 [PI]_[PI].AND.NOT.[BR], HOLD RIGHT
; 7340 TR [AR], PI.REQ/1
; 7341 =0 [PI]_[PI].OR.[BR], HOLD RIGHT
; 7342 TR [AR], PI.TSN/1
; 7343 =0 [PI]_[PI].OR.#,PI.ON/1, HOLD LEFT
; 7344 TR [AR], PI.TSF/1
; 7345 =0 [PI]_[PI].AND.NOT.#,PI.ON/1, HOLD LEFT
; 7346 TR [AR], PI.TCN/1
; 7347 =0 [PI]_[PI].OR.[BR], HOLD LEFT
; 7348 TR [AR], PI.TCF/1
; 7349 =0**0 [PI]_[PI].AND.NOT.[BR], HOLD LEFT
; 7350 PIEXIT: CALL LOAD PI
; 7351 =1**1
; 7352 DONE
; 7353 =
; 7354
; 7355 ;SUBROUTINE TO LOAD PI HARDWARE
; 7356 ;CALL WITH:
; 7357 ; CALL LOAD PI
; 7358 ;RETURNS IO WITH PI HARDWARE LOADED
; 7359 ;
; 7360 LOADPI: [TO]_[PI] SWAP ;PUT ACTIVE CHANS IN LH
; 7361 LDPI2: [TO]_-1, HOLD LEFT ;DONT MASK RH
; 7362 [TO]_[TO].AND.[PI] ;ONLY REQUEST CHANS THAT ARE ON
; 7363 .NOT.[TO], LOAD PI, ;RELOAD HARDWARE
; 7364 RETURN [IO] ;RETURN TO CALLER
; 7365

U 1715, 3626,3441,1405,4174,4007,0700,0000,0000,0000
U 1714, 2404,4553,0300,4374,4007,0331,0000,0001,0000
U 2404, 2405,4221,0014,4174,4007,0700,0000,0000,0000
U 2405, 2406,4553,0300,4374,4007,0331,0000,0074,0000
U 2406, 2732,4551,0202,4374,0007,0700,0000,0077,7740
U 2407, 3620,4551,0305,4374,4007,0700,0000,0000,0177
U 3620, 3621,3770,0505,4344,0007,0700,0000,0000,0000
U 3621, 2410,4553,0300,4374,4007,0331,0000,0002,0000
U 2410, 2411,5111,0514,4174,0007,0700,0000,0000,0000
U 2411, 2412,4553,0300,4374,4007,0331,0000,0000,4000
U 2412, 2413,3111,0514,4174,0007,0700,0000,0000,0000
U 2413, 2414,4553,0300,4374,4007,0331,0000,0000,0200
U 2414, 2415,3551,1414,4370,4007,0700,0000,0000,0200
U 2415, 2416,4553,0300,4374,4007,0331,0000,0000,0400
U 2416, 2417,5551,1414,4370,4007,0700,0000,0000,0200
U 2417, 2420,4553,0300,4374,4007,0331,0000,0000,2000
U 2420, 2421,3111,0514,4170,4007,0700,0000,0000,0000
U 2421, 0404,4553,0300,4374,4007,0331,0000,0000,1000
U 0404, 0405,5111,0514,4170,4007,0700,0000,0000,0000
U 0405, 3623,3770,1416,4344,4007,0700,0010,0000,0000

U 0415, 0110,3443,0100,4174,4156,4700,0200,0014,0012

U 3622, 3623,3770,1416,4344,4007,0700,0000,0000,0000
U 3623, 3624,2441,0716,4170,4007,0700,4000,0000,0000
U 3624, 3625,4111,1416,4174,4007,0700,0000,0000,0000

U 3625, 0010,7443,1600,4174,4434,1700,0000,0000,0000
```

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 203
IO -- INTERNAL DEVICES -- SUBROUTINES

; 7366 .TOC "IO -- INTERNAL DEVICES -- SUBROUTINES"
; 7367
; 7368
; 7369 ;HERE WITH SOMETHING IN BR STORE IT @AR
; 7370 RTNREG: VMA_[AR], START WRITE
; 7371 MEM WRITE, MEM_[BR], J/DONE
; 7372

U 3626, 3627, 3443, 0300, 4174, 4007, 0700, 0200, 0003, 0012
U 3627, 1400, 3333, 0005, 4175, 5007, 0701, 0200, 0000, 0002

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 204
IO -- INTERNAL DEVICES -- SUBROUTINES

```
U 1722, 3630,3443,0300,4174,4147,0700,0200,0000,0010
U 3630, 2422,4221,0003,4174,4007,0700,0000,0000,0000
U 2422, 3634,3771,0003,4374,4007,0700,0010,0037,7377
U 2423, 3631,3771,0005,4374,4247,0700,0000,0000,1001
U 3631, 2424,3333,0003,4174,4247,0700,0000,0000,1000

U 2424, 2424,1111,0503,4174,4247,0630,6000,0060,1000
U 2425, 2431,3333,0003,4174,4007,0700,0000,0000,0000

U 2426, 3634,3771,0003,4374,4007,0700,0010,0037,7377
U 2427, 3632,3771,0005,4374,4347,0700,0000,0000,1001
U 3632, 2430,3333,0003,4174,4347,0700,0000,0000,1000

U 2430, 2430,1111,0503,4174,4347,0630,6000,0060,1000
U 2431, 3633,4223,0000,7174,4007,0700,0400,0000,0424
U 3633, 1400,4223,0000,7174,4007,0700,0400,0000,0423

U 3634, 0001,4443,0000,4174,4004,1700,2000,0071,0375

; 7373 ;CACHE SWEEP
; 7374
; 7375 1722:
; 7376 CLRPT: VMA [AR], ;PUT CORRECT ADDRESS IN VMA
; 7377 LOAD PAGE TABLE ;GET SET TO WRITE PAGE TABLE
; 7378 [AR]_O ;CLEAR ENTRY
; 7379 =0 [AR]_#,#/377377, ;INITIAL VMA VALUE
; 7380 CALL [SSWEEP] ;LOAD THE SC
; 7381 [BR]_#; #/1001, ;CONSTANT TO KEEP ADDING
; 7382 CLRC5H ;START TO CLEAR CACHE
; 7383 READ [AR], CLRC5H ;FIRST THING TO CLEAR
; 7384 =0
; 7385 CLRPTL: [AR]_[AR]-[BR], ;UPDATE AR (AND PUT ON DP)
; 7386 CLRC5H, ;SWEEP ON NEXT STEP
; 7387 STEP SC, ;SKIP IF WE ARE DONE
; 7388 J/CLRPTL ;LOOP FOR ALL ENTRIES
; 7389 READ [AR], J/ZAPPTA ;CLEAR LAST ENTRY
; 7390
; 7391 =0
; 7392 SWEEP: [AR]_#,#/377377, ;INITIAL VMA VALUE
; 7393 CALL [SSWEEP] ;LOAD NUMBER OF STEPS INTO SC
; 7394 [BR]_#; #/1001, ;CONSTANT TO KEEP ADDING
; 7395 SWEEP ;START SWEEP
; 7396 READ [AR], SWEEP ;FIRST THING TO CLEAR
; 7397 =0
; 7398 SWEEPL: [AR]_[AR]-[BR], ;UPDATE AR (AND PUT ON DP)
; 7399 SWEEP, ;SWEEP ON NEXT STEP
; 7400 STEP SC, ;SKIP IF WE ARE DONE
; 7401 J/SWEEPL ;LOOP FOR ALL ENTRIES
; 7402 ;CLEAR LAST ENTRY AND
; 7403 ZAPPTA: WORK[PTA.U]_O ;FORGET PAGE TABLE ADDRESS
; 7404 WORK[PTA.E]_O, ;FORGET PAGE TABLE ADDRESS
; 7405 J/DONE ;ALL DONE
; 7406
; 7407 SSWEEP: SC_S#, S#/375, ;NUMBER OF STEPS
; 7408 RETURN [1] ;RETURN
; 7409
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 205
IO -- INTERNAL DEVICES -- SUBROUTINES

```
; 7410 ;WE COME HERE EITHER FROM NEXT INSTRUCTION DISPATCH OR PAGE FAIL
; 7411 ; LOGIC. IN ALL CASES, THE CURRENT INSTRUCTION IS CORRECTLY SETUP
; 7412 ; TO RESTART PROPERLY.
; 7413
; 7414 ;FIRST SET THE CORRECT PI IN PROGRESS BIT
; 7415 ; [FLG]_[FLG].OR.#,FLG.PI/1, HOLD RIGHT,
; 7416 ; J/PI ;SET PI CYCLE AND PROCESS PI
; 7417 =1000
; 7418 PI: AD/D, DBUS/PI NEW, ;LOOK AT NEW LEVEL
; 7419 DISP/DP LEFT, 3T, ;DISPATCH ON IT
; 7420 J/PI ;GO TO 1 OF NEXT 7 PLACES
; 7421 =1001 [PI]_[PI].OR.#, #/040000, HOLD LEFT, J/PIP1
; 7422 =1010 [PI]_[PI].OR.#, #/020000, HOLD LEFT, J/PIP2
; 7423 =1011 [PI]_[PI].OR.#, #/010000, HOLD LEFT, J/PIP3
; 7424 =1100 [PI]_[PI].OR.#, #/004000, HOLD LEFT, J/PIP4
; 7425 =1101 [PI]_[PI].OR.#, #/002000, HOLD LEFT, J/PIP5
; 7426 =1110 [PI]_[PI].OR.#, #/001000, HOLD LEFT, J/PIP6
; 7427 =1111 [PI]_[PI].OR.#, #/000400, HOLD LEFT, J/PIP7
; 7428 PIP1: [BRX]_O XWD [1], J/PI10 ;REMEMBER WE ARE AT LEVEL 1
; 7429 PIP2: [BRX]_O XWD [2], J/PI10 ;REMEMBER WE ARE AT LEVEL 2
; 7430 PIP3: [BRX]_O XWD [3], J/PI10 ;REMEMBER WE ARE AT LEVEL 3
; 7431 PIP4: [BRX]_O XWD [4], J/PI10 ;REMEMBER WE ARE AT LEVEL 4
; 7432 PIP5: [BRX]_O XWD [5], J/PI10 ;REMEMBER WE ARE AT LEVEL 5
; 7433 PIP6: [BRX]_O XWD [6], J/PI10 ;REMEMBER WE ARE AT LEVEL 6
; 7434 PIP7: [BRX]_O XWD [7], J/PI10 ;REMEMBER WE ARE AT LEVEL 7
; 7435
; 7436 PI10: [AR]_[PI].AND.# CLR LH, ;TURN OFF PI SYSTEM
; 7437 #/077577 ; TILL WE ARE DONE
; 7438 .NOT.[AR], LOAD PI ;
; 7439 ABORT MEM CYCLE ;NO MORE TRAPS
; 7440 =0 [AR]_VMA IO READ, ;SETUP TO READ WRU BITS
; 7441 WRU_CYCLE/1, ;
; 7442 CALL [STRTIO] ;START THE CYCLE
; 7443 MEM READ, ;WAIT FOR DATA
; 7444 [AR]_IO DATA, 3T, ;PUT DATA IN AR
; 7445 SKIP ADR.EQ.O ;SEE IF ANYONE THERE
; 7446 =0 [ARX]_O, J/VECINT ;YES--VECTORED INTERRUPT
; 7447 [AR]_[BRX]*2 ;N*2
; 7448 [AR]_[AR]+#, #/40, 3T, ;2*N+40
; 7449 HOLD LEFT ;
; 7450 [AR]_[AR]+[EBR], ;ABSOLUTE ADDRESS OF
; 7451 J/PI40 ; INTERRUPT INSTRUCTION
; 7452
```

```
U 0770, 0770,3773,0000,4074,4003,1701,0000,0000,0000
U 0771, 3635,3551,1414,4370,4007,0700,0000,0004,0000
U 0772, 3636,3551,1414,4370,4007,0700,0000,0002,0000
U 0773, 3637,3551,1414,4370,4007,0700,0000,0001,0000
U 0774, 3640,3551,1414,4370,4007,0700,0000,0000,4000
U 0775, 3641,3551,1414,4370,4007,0700,0000,0000,2000
U 0776, 3642,3551,1414,4370,4007,0700,0000,0000,1000
U 0777, 3643,3551,1414,4370,4007,0700,0000,0000,0400
U 3635, 3644,4751,1206,4374,4007,0700,0000,0000,0001
U 3636, 3644,4751,1206,4374,4007,0700,0000,0000,0002
U 3637, 3644,4751,1206,4374,4007,0700,0000,0000,0003
U 3640, 3644,4751,1206,4374,4007,0700,0000,0000,0004
U 3641, 3644,4751,1206,4374,4007,0700,0000,0000,0005
U 3642, 3644,4751,1206,4374,4007,0700,0000,0000,0006
U 3643, 3644,4751,1206,4374,4007,0700,0000,0000,0007

U 3644, 3645,4251,1403,4374,4007,0700,0000,0007,7577
U 3645, 3646,7443,0300,4174,4437,0700,0000,0000,0000
U 3646, 2432,4223,0000,4364,4277,0700,0200,0000,0010

U 2432, 3701,4571,1203,4374,4007,0700,0010,0024,1300

U 2433, 2434,3771,0003,4364,4007,0331,0200,0000,0002
U 2434, 3655,4221,0004,4174,4007,0700,0000,0000,0000
U 2435, 3647,3445,0603,4174,4007,0700,0000,0000,0000

U 3647, 3650,0551,0303,4370,4007,0701,0000,0000,0040

U 3650, 3651,0111,1003,4174,4007,0700,0000,0000,0000
```

; T10KL...R[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS:O MICROCODE V124, 27-JUL-84 Page 206
IO -- INTERNAL DEVICES -- SUBROUTINES

U 3651, 3652,3443,0300,4174,4007,0700,0200,0024,1016
U 3652, 3653,3771,0003,4365,5007,0701,0200,0020,0012
U 3653, 2436,6553,0300,4374,4007,0321,0000,0025,4340
U 2436, 2440,6553,0300,4374,4007,0321,0000,0026,4000
U 2437, 3654,4521,1205,4074,4007,0700,0000,0000,0000
U 3654, 0060,4223,0000,4174,4467,0700,0000,0000,0004

U 2440, 0104,4751,1217,4374,4007,0700,0000,0000,0101
U 2441, 0470,4443,0000,4174,4007,0700,0200,0023,0002
U 0470, 3702,3741,0105,4074,4007,0700,0010,0000,0000
GS
U 0474, 3677,3771,0003,4374,0007,0700,0010,0000,0000
U 0475, 2677,3441,0301,4174,4467,0700,0000,0000,0004

; 7453 ;HERE WITH ABSOLUTE ADDRESS OF INTERRUPT INSTRUCTION IN [AR]
; 7454 PI40: VMA_[AR], VMA PHYSICAL READ ;FETCH THE INSTRUCTION
; 7455 PI50: MEM READ, [AR]_MEM, LOAD VMA, ;FETCH INSTRUCTION
; 7456 3T, FORCE EXEC ;E IS EXEC MODE
; 7457 [AR].XOR.#, #/254340, 3T, SKIP ADL.EQ.O
=0 ; 7458 [AR].XOR.#, #/264000, SKIP ADL.EQ.O, 3T, J/PIJSR
; 7459 [BR]_FLAGS ;SAVE FLAGS
; 7460 AD/ZERO, LOAD FLAGS,
; 7461 J/PIXPCW ;ENTER EXEC MODE AND ASSUME
; 7462 ; WE HAVE AN XPCW
; 7463 ;IF WE HALT HERE ON A VECTORED INTERRUPT, WE HAVE
; 7464 ; TO/ WHAT WE READ FROM BUS AS VECTOR
; 7465 ; ARX/ EPT+100+DEVICE
; 7466 ; BR/ ADDRESS OF ILLEGAL INSTRUCTION
; 7467 ; BRX/ VECTOR (MASKED AND SHIFTED)
; 7468 =0
; 7469 PIJSR: HALT [ILLII] ;NOT A JSR OR XPCW
; 7470 START WRITE, FORCE EXEC ;PREPARE TO STORE OLD PC
=0*0 ; 7471 [BR]_PC WITH FLAGS, ;OLD PC
; 7472 CALL [STOBR] ;STORE OLD PC
; 7473 =1*0 [AR]_#, #/O, HOLD RIGHT, ;PREPARE TO CLEAR FL

; 7474 CALL [INCAR] ;BUMP POINTER
; 7475 =1*1 [PC]_[AR], LOAD FLAGS, ;NEW PC
; 7476 J/PISET ;CLEAR PI CYCLE & START
; 7477 ; INTERRUPT PROGRAM
; 7478 =
; 7479

Produced on Advanced Information Services Electronic Laser Printer, PKCJ1E59, DTN: 222-7881

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; INOUT.MIC[10,1141] 09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 207
IO -- INTERNAL DEVICES -- SUBROUTINES

U 3655, 2442, 3445, 0303, 4174, 4007, 0530, 0000, 0000, 0000
U 2442, 3655, 0111, 1504, 4174, 4007, 0700, 0000, 0000, 0000
U 2443, 2444, 4571, 1203, 4374, 4007, 0700, 0000, 0024, 1240
U 2444, 3701, 3111, 0403, 4174, 4007, 0700, 0010, 0000, 0000
U 2445, 2446, 3771, 0016, 4364, 4007, 0700, 0200, 0000, 0002
U 2446, 3676, 0551, 1005, 4374, 4007, 0701, 0010, 0000, 0100
U 2447, 3656, 0111, 0504, 4174, 4007, 0700, 0200, 0024, 1016
U 3656, 2450, 3771, 0005, 4365, 5007, 0331, 0200, 0000, 0002
U 2450, 3657, 4557, 1606, 4374, 4007, 0701, 0000, 0000, 0774
U 2451, 0104, 4751, 1217, 4374, 4007, 0700, 0000, 0000, 0102
U 3657, 3660, 3447, 0606, 4174, 4007, 0700, 0000, 0000, 0000
U 3660, 3652, 0111, 0605, 4174, 4007, 0700, 0200, 0024, 0012

```
; 7480 ;HERE TO PROCESS A VECTORED INTERRUPT. AT THIS POINT:  
; 7481 ; AR/ WRU BITS (BIT 18 FOR DEVICE 0)  
; 7482 ; ARX/ 0  
; 7483 VECINT: [AR]_[AR]*2, ;SHIFT LEFT (UNSHIFTED ON DP)  
; 7484 SKIP DP18 ;ANYONE THERE?  
; 7485 =0 [ARX]_[ARX]+[XWD1], ;NO--BUMP BOTH HALVES  
; 7486 J/VECINT ;KEEP LOOKING  
; 7487 [AR] VMA IO READ, ;SETUP FOR VECTOR CYCLE  
; 7488 VECTOR CYCLE/1 ;  
; 7489 =0 [AR]_[AR].OR.[ARX], ;PUT IN UNIT NUMBER  
; 7490 CALL [STRTIO] ;START CYCLE  
; 7491 MEM READ, ;WAIT FOR VECTOR (SEE DPM5)  
; 7492 [TO] IO DATA ;GET VECTOR  
; 7493 =0 [BR]_[EBR]+#, 3T, #/100, ;EPT+100  
; 7494 CALL [CLARXL] ;CLEAR ARX LEFT  
; 7495 [ARX]_[ARX]+[BR], ;EPT+100+DEVICE  
; 7496 VMA PHYSICAL READ ;FETCH WORD  
; 7497 MEM READ, [BR]_MEM, 3T, ;GET POINTER  
; 7498 SKIP ADR.EQ.0 ;SEE IF NON-ZERO  
; 7499 =0 [BRX]_([TO].AND.#)*.5, 3T, ;OK--MAKE VECTOR MOD 400  
; 7500 #/774, J/VECIN1 ; AND SHIFT OVER  
; 7501 HALT [ILLINT]  
; 7502 VECIN1: [BRX]_[BRX]*.5 ;SHIFT 1 MORE PLACE  
; 7503 [BR]_[BR]+[BRX], ;ADDRESS OF WORD TO USE  
; 7504 LOAD VMA, FORCE EXEC, ;FORCE EXEC VIRTUAL ADDRESS  
; 7505 START READ, J/PI50 ;GO GET INSTRUCTION  
; 7506
```


; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 208
PRIORITY INTERRUPTS -- DISMISS SUBROUTINE

```
; 7507 .TOC "PRIORITY INTERRUPTS -- DISMISS SUBROUTINE"  
; 7508  
; 7509 ;SUBROUTINE TO DISMISS THE HIGHEST PI IN PROGRESS  
; 7510 ;RETURNS 4 ALWAYS  
; 7511  
; 7512 ;DISMISS:  
; 7513 ; TR [PI], #/077400 ;ANY PI IN PROGRESS?  
; 7514 =0  
; 7515 JEN1: [BR]_#, PI.IP1/1, J/DSMS1 ;YES--START LOOP  
; 7516 RETURN [4] ;NO--JUST RETURN  
; 7517  
; 7518 DSMS1: [PI].AND.[BR], SKIP ADR.EQ.0  
; 7519 =0 [PI]_[PI].AND.NOT.[BR], HOLD LEFT, RETURN [4]  
; 7520 [BR]_[BR]*.5, J/DSMS1  
; 7521
```

U 2452, 3661,3771,0005,4374,4007,0700,0000,0004,0000
U 2453, 0004,4443,0000,4174,4004,1700,0000,0000,0000

U 3661, 2454,4113,0514,4174,4007,0330,0000,0000,0000
U 2454, 0004,5111,0514,4170,4004,1700,0000,0000,0000
U 2455, 3661,3447,0505,4174,4007,0700,0000,0000,0000

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 209
EXTERNAL IO INSTRUCTIONS

```
; 7522 .TOC "EXTERNAL IO INSTRUCTIONS"
; 7523
; 7524 .DCODE
D 0710, 1210,1614,0100 ; 7525 710: IOT, WORD-TNE, J/TIOX
D 0711, 1214,1614,0100 ; 7526 711: IOT, WORD-TNN, J/TIOX
D 0720, 1200,1614,0100 ; 7527 720: IOT, TNE, J/TIOX
D 0721, 1204,1614,0100 ; 7528 721: IOT, TNN, J/TIOX
; 7529 .UCODE
; 7530
; 7531 1614:
U 1614, 2456,4443,0000,4174,4007,0700,0010,0000,0000 ; 7532 TIOX: CALL [IORD]
U 1617, 0014,4551,0305,0274,4003,7700,0000,0000,0000 ; 7533 1617: [BR]_[AR].AND.AC, TEST DISP
; 7534
; 7535 .DCODE
D 0712, 1210,1460,0100 ; 7536 712: IOT, B/10, J/RDIO
D 0713, 1210,1461,0100 ; 7537 713: IOT, B/10, J/WRIO
D 0722, 1200,1460,0100 ; 7538 722: IOT, B/O, J/RDIO
D 0723, 1200,1461,0100 ; 7539 723: IOT, B/O, J/WRIO
; 7540 .UCODE
; 7541
; 7542 1460:
U 1460, 2456,4443,0000,4174,4007,0700,0010,0000,0000 ; 7543 RDIO: CALL [IORD]
U 1463, 1400,3440,0303,0174,4007,0700,0400,0000,0000 ; 7544 1463: AC_[AR], J/DONE
; 7545
; 7546 1461:
U 1461, 2466,3771,0005,0276,6007,0700,0000,0000,0000 ; 7547 WRIO: [BR]_AC, J/IOWR
; 7548
; 7549 .DCODE
D 0714, 1210,1644,0100 ; 7550 714: IOT, B/10, J/BIXUB
D 0715, 1214,1644,0100 ; 7551 715: IOT, B/14, J/BIXUB
D 0724, 1200,1644,0100 ; 7552 724: IOT, B/O, J/BIXUB
D 0725, 1204,1644,0100 ; 7553 725: IOT, B/4, J/BIXUB
; 7554 .UCCODE
; 7555
; 7556 1644:
U 1644, 2456,3441,0306,4174,4007,0700,0010,0000,0000 ; 7557 BIXUB: [BRX]_[AR], ;SAVE EFFECTIVE ADDRESS
; 7558 CALL [IORD] ;GO GET THE DATA
; 7559 1647: [BR]_[AR], ;COPY DATA ITEM
; 7560 B DISP ;SEE IF SET OR CLEAR
; 7561 =1011 [BR]_[BR].OR.AC, ;SET BITS
; 7562 J/BIXUB1 ;GO DO WRITE
; 7563 [BR]_[BR].AND.NOT.AC, ;CLEAR BITS
; 7564 J/BIXUB1 ;GO DO WRITE
; 7565
; 7566 BIXUB1: [AR]_[BRX], ;RESTORE ADDRESS
; 7567 J/IOWR
; 7568
```

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 210
EXTERNAL IO INSTRUCTIONS

U 2456, 2476,4443,0000,4174,4137,0700,0010,0000,0000
U 2457, 0067,4443,0000,4174,4003,7700,0000,0000,0000

U 0067, 2460,4571,1205,4374,4007,0700,0000,0024,1220
U 0077, 2460,4571,1205,4374,4007,0700,0000,0024,1200

U 2460, 3671,3113,0305,4174,4007,0701,0210,0000,0036

U 2461, 1027,3771,0005,4364,4003,7700,0200,0000,0002
U 1027, 2462,4553,0300,4374,4007,0331,0000,0000,0001
U 1037, 0003,3441,0503,4174,4004,1700,0000,0000,0000

U 2462, 2464,3447,0505,4174,4007,0700,2000,0071,0005

U 2463, 0003,4551,0503,4374,4004,1700,0000,0000,0377

U 2464, 2464,3447,0505,4174,4007,0630,2000,0060,0000

U 2465, 0003,4551,0503,4374,4004,1700,0000,0000,0377

```
; 7569 ;SUBROUTINE TO READ FROM AN IO DEVICE
; 7570 ;CALL WITH:
; 7571 ; SECTION 0 EFFECTIVE ADDRESS IN AR
; 7572 ; INSTRUCTION IN HR
; 7573 ;RETURN 3 WITH WORD OR BYTE IN AR
; 7574 ;
; 7575 =0
; 7576 IORD: CLR IO BUSY, ;CLEAR BUSY
; 7577 CALL [IOEA] ;COMPUTE IO EA
; 7578 B DISP
; 7579 =10111 [BR]_VMA IO READ, ;BYTE MODE
; 7580 IO BYTE/1, ;SET BYTE FLAG
; 7581 J/IORD1 ;GO DO C/A CYCLE
; 7582 =11111 [BR]_VMA IO READ ;WORD MODE
; 7583 =
; 7584 =0
; 7585 IORD1: VMA [AR].OR.[BR] WITH FLAGS,
; 7586 CALL [IOWAIT] ;WAIT FOR THINGS COMPLETE
; 7587 MEM READ, ;MAKE SURE REALLY READY
; 7588 [BR] IO DATA, ;PUT DATA IN BR
; 7589 B DISP ;SEE IF BYTE MODE
; 7590 =0111 TR [AR], #/1, J/IORD2 ;BYTE MODE SEE IF ODD
; 7591 [AR]_[BR], RETURN [3] ;ALL DONE
; 7592
; 7593 ;HERE ON WORD MODE
; 7594 =0
; 7595 IORD2: [BR]_[BR]*.5, SC_5, ;LEFT BYTE
; 7596 J/IORD3 ;GO SHIFT IT
; 7597 [AR]_[BR].AND.#, ;MASK IT
; 7598 #/377, RETURN [3] ;ALL DONE
; 7599
; 7600 =0
; 7601 IORD3: [BR]_[BR]*.5, ;SHIFT OVER
; 7602 STEP SC, J/IORD3 ;
; 7603 [AR]_[BR].AND.#, ;MASK IT
; 7604 #/377, RETURN [3] ;ALL DONE
; 7605
```

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 211
EXTERNAL IO INSTRUCTIONS

U 2466, 2476,4443,0000,4174,4137,0700,0010,0000,0000
U 2467, 0227,4443,0000,4174,4003,7700,0000,0000,0000
U 0227, 2472,4553,0300,4374,4007,0331,0000,0000,0001
U 0237, 3663,4571,1204,4374,4007,0700,0000,0021,1200

U 3663, 2470,3113,0304,4174,4007,0701,0200,0000,0036

U 2470, 3671,3333,0005,4175,5007,0701,0210,0000,0002
U 2471, 0110,3443,0100,4174,4156,4700,0200,0014,0012

U 2472, 2474,3445,0505,4174,4007,0700,2000,0071,0005

U 2473, 3663,4571,1204,4374,4007,0700,0000,0021,1220

U 2474, 2474,3445,0505,4174,4007,0630,2000,0060,0000

U 2475, 3663,4571,1204,4374,4007,0700,0000,0021,1220

```
; 7606 ;ROUTINE TO WRITE TO AN IO DEVICE
; 7607 ;CALL WITH:
; 7608 ; SECTION O EFFECTIVE ADDRESS IN AR
; 7609 ; INSTRUCTION IN HR
; 7610 ; WORD OR BYTE IN BR
; 7611 ;RETURNS BACK TO USER
; 7612 ;
; 7613 =0
; 7614 IOWR: CLR IO BUSY, ;CLEAR BUSY
; 7615 CALL [IOEA] ;COMPUTE IO EA
; 7616 B DISP
; 7617 =10111 TR [AR], #/1, J/IOWR2 ;BYTE MODE
; 7618 =11111 [ARX]_VMA IO WRITE ;SETUP FLAGS
; 7619 =
; 7620 IOWR1: VMA_[AR].OR.[ARX] WITH FLAGS
; 7621 =0 MEM WRITE, MEM_[BR], ;SEND DATA
; 7622 CALL [IOWAIT] ;WAIT FOR DATA
; 7623 DONE ;RETURN
; 7624
; 7625 ;HERE FOR BYTE MODE
; 7626 =0
; 7627 IOWR2: [BR]_[BR]*2, SC_5, ;ODD--MOVE LEFT
; 7628 J/IOWR3 ; ..
; 7629 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7630 IO BYTE/1, J/IOWR1 ; ..
; 7631
; 7632 =0
; 7633 IOWR3: [BR]_[BR]*2, STEP SC, ;SHIFT LEFT
; 7634 J/IOWR3 ;KEEP SHIFTING
; 7635 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7636 IO BYTE/1, J/IOWR1 ; ..
; 7637
```

; TIOKL.MIC[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 210
EXTERNAL IO INSTRUCTIONS

```
; 7569 ;SUBROUTINE TO READ FROM AN IO DEVICE
; 7570 ;CALL WITH:
; 7571 ; SECTION O EFFECTIVE ADDRESS IN AR
; 7572 ; INSTRUCTION IN HR
; 7573 ;RETURN 3 WITH WORD OR BYTE IN AR
; 7574 ;
; 7575 =0
; 7576 IORD: CLR IO BUSY, ;CLEAR BUSY
; 7577 CALL [IOEA] ;COMPUTE IO EA
; 7578 B DISP
; 7579 =10111 [BR]_VMA IO READ, ;BYTE MODE
; 7580 IO BYTE/1, ;SET BYTE FLAG
; 7581 J/IORD1 ;GO DO C/A CYCLE
; 7582 =11111 [BR]_VMA IO READ ;WORD MODE
; 7583 =
; 7584 =0
; 7585 IORD1: VMA [AR].OR.[BR] WITH FLAGS.
; 7586 CALL [IOWAIT] ;WAIT FOR THINGS COMPLETE
; 7587 MEM READ, ;MAKE SURE REALLY READY
; 7588 [BR]_IO DATA, ;PUT DATA IN BR
; 7589 B DISP ;SEE IF BYTE MODE
; 7590 =0111 TR [AR], #/1, J/IORD2 ;BYTE MODE SEE IF ODD
; 7591 [AR]_[BR], RETURN [3] ;ALL DONE
; 7592
; 7593 ;HERE ON WORD MODE
; 7594 =0
; 7595 IORD2: [BR]_[BR]*.5, SC_5, ;LEFT BYTE
; 7596 J/IORD3 ;GO SHIFT IT
; 7597 [AR]_[BR].AND.#, ;MASK IT
; 7598 #/377, RETURN [3] ;ALL DONE
; 7599
; 7600 =0
; 7601 IORD3: [BR]_[BR]*.5, ;SHIFT OVER
; 7602 STEP SC, J/IORD3 ;
; 7603 [AR]_[BR].AND.#, ;MASK IT
; 7604 #/377, RETURN [3] ;ALL DONE
; 7605
```

U 2456, 2476,4443,0000,4174,4137,0700,0010,0000,0000
U 2457, 0067,4443,0000,4174,4003,7700,0000,0000,0000

U 0067, 2460,4571,1205,4374,4007,0700,0000,0024,1220
U 0077, 2460,4571,1205,4374,4007,0700,0000,0024,1200

U 2460, 3671,3113,0305,4174,4007,0701,0210,0000,0036

U 2461, 1027,3771,0005,4364,4003,7700,0200,0000,0002
U 1027, 2462,4553,0300,4374,4007,0331,0000,0000,0001
U 1037, 0003,3441,0503,4174,4004,1700,0000,0000,0000

U 2462, 2464,3447,0505,4174,4007,0700,2000,0071,0005

U 2463, 0003,4551,0503,4374,4004,1700,0000,0000,0377

U 2464, 2464,3447,0505,4174,4007,0630,2000,0060,0000

U 2465, 0003,4551,0503,4374,4004,1700,0000,0000,0377

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES6, DTN: 233-7881

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 211
EXTERNAL IO INSTRUCTIONS

```
; 7606 ;ROUTINE TO WRITE TO AN IO DEVICE
; 7607 ;CALL WITH:
; 7608 ; SECTION O EFFECTIVE ADDRESS IN AR
; 7609 ; INSTRUCTION IN HR
; 7610 ; WORD OR BYTE IN BR
; 7611 ;RETURNS BACK TO USER
; 7612 ;
; 7613 =0
; 7614 IOWR: CLR IO BUSY, ;CLEAR BUSY
; 7615 CALL [IOEA] ;COMPUTE IO EA
; 7616 B DISP
; 7617 =10111 TR [AR], #/1, J/IOWR2 ;BYTE MODE
; 7618 =11111 [ARX]_VMA IO WRITE ;SETUP FLAGS
; 7619 =
; 7620 IOWR1: VMA_[AR].OR.[ARX] WITH FLAGS
; 7621 =0 MEM WRITE, MEM_[BR], ;SEND DATA
; 7622 CALL [IOWAIT] ;WAIT FOR DATA
; 7623 DONE ;RETURN
; 7624
; 7625 ;HERE FOR BYTE MODE
; 7626 =0
; 7627 IOWR2: [BR]_[BR]*2, SC_5, ;ODD--MOVE LEFT
; 7628 J/IOWR3 ; ..
; 7629 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7630 IO BYTE/1, J/IOWR1 ; ..
; 7631
; 7632 =0
; 7633 IOWR3: [BR]_[BR]*2, STEP SC, ;SHIFT LEFT
; 7634 J/IOWR3 ;KEEP SHIFTING
; 7635 [ARX]_VMA IO WRITE, ;SETUP FLAGS
; 7636 IO BYTE/1, J/IOWR1 ; ..
; 7637
```

```
U 2466, 2476,4443,0000,4174,4137,0700,0010,0000,0000
U 2467, 0227,4443,0000,4174,4003,7700,0000,0000,0000
U 0227, 2472,4553,0300,4374,4007,0331,0000,0000,0001
U 0237, 3663,4571,1204,4374,4007,0700,0000,0021,1200

U 3663, 2470,3113,0304,4174,4007,0701,0200,0000,0036

U 2470, 3671,3333,0005,4175,5007,0701,0210,0000,0002
U 2471, 0110,3443,0100,4174,4156,4700,0200,0014,0012

U 2472, 2474,3445,0505,4174,4007,0700,2000,0071,0005

U 2473, 3663,4571,1204,4374,4007,0700,0000,0021,1220

U 2474, 2474,3445,0505,4174,4007,0630,2000,0060,0000

U 2475, 3663,4571,1204,4374,4007,0700,0000,0021,1220
```

; T1OKL.MICR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 212
EXTERNAL IO INSTRUCTIONS

U 2476, 3673, 1113, 0701, 4170, 4007, 0700, 4210, 0004, 0012
U 2477, 3664, 7441, 0306, 4174, 4007, 0700, 0000, 0000, 0000
U 3664, 2500, 4553, 0600, 4374, 4007, 0321, 0000, 0070, 0000

U 2500, 2502, 4553, 0200, 4374, 4007, 0321, 0000, 0000, 0020

U 2501, 2500, 4713, 1203, 7174, 4007, 0700, 0400, 0000, 0422

U 2502, 3665, 3771, 0003, 7274, 4007, 0701, 0000, 0000, 0422
U 2503, 1055, 4443, 0000, 2174, 4006, 6700, 0000, 0000, 0000

U 1055, 2506, 3771, 0004, 2274, 4007, 0120, 0000, 0000, 0000

U 1057, 0001, 3771, 0003, 7274, 4124, 1701, 0000, 0000, 0422

U 3665, 3666, 3333, 0002, 4174, 4217, 0700, 0000, 0000, 0000
U 3666, 2504, 4553, 0200, 4374, 4007, 0321, 0000, 0000, 0017
U 2504, 2505, 0551, 0303, 2270, 4007, 0701, 0000, 0000, 0000
U 2505, 3667, 3443, 0300, 4174, 4007, 0700, 0200, 0004, 0012

U 3667, 0001, 3771, 0003, 4365, 5124, 1700, 0200, 0000, 0002

U 2506, 0001, 0551, 0403, 7274, 4124, 1701, 0000, 0000, 0422
U 2507, 3670, 0551, 0403, 7274, 4007, 0701, 0000, 0000, 0422

U 3670, 0001, 4221, 0003, 4174, 0124, 1700, 0000, 0000, 0000

```
; 7638 ;HERE TO COMPUTE IO EFFECTIVE ADDRESS
; 7639 ;CALL WITH:
; 7640 ; SECTION 0 EFFECTIVE ADDRESS IN AR
; 7641 ; INSTRUCTION IN HR
; 7642 ;RETURN 1 WITH EA IN AR
; 7643 ;
; 7644 =0
; 7645 IOEA: VMA [PC]-1, ;GET INSTRUCTION
; 7646 START READ, ;...
; 7647 CALL [LOADAR] ;PUT WORD IN AR
; 7648 [BRX].NOT.[AR] ;SEE IF IN RANGE 700-777
; 7649 TL [BRX], #/700000 ;..
; 7650 =0
; 7651 IOEA1: TL [HR], #/20, J/IOEA2 ;INDIRECT?
; 7652 WORK[YSAVE]_[AR] CLR LH, ;DIRECT IO INSTRUCTION
; 7653 J/IOEA1 ;SAVE Y FOR EA CALCULATION
; 7654 =0
; 7655 IOEA2: [AR] WORK[YSAVE], ;@--GET SAVED Y
; 7656 J/IOEA1 ;GET Y AND GO
; 7657 EA MODE DISP ;WAS THERE INDEXING?
; 7658 =1101 [ARX]_XR, SKIP ADL.LE.O, ;SEE IF LOCAL OR GLOBAL INDEXING
; 7659 2T, J/IOEAX ;...
; 7660 [AR] WORK[YSAVE], ;JUST PLAIN IO
; 7661 CLR IO LATCH, RETURN [1]
; 7662
; 7663 IOEAI: READ [HR], DBUS/DP. ;LOAD XR FLOPS IN CASE
; 7664 LOAD INST EA ; THERE IS INDEXING
; 7665 TL [HR], #/17 ;WAS THERE ALSO INDEXING
; 7666 =0 [AR]_[AR]+XR, 3T, HOLD LEFT ;YES--ADD IN INDEX VALUE
; 7667 VMA [AR], START READ ;FETCH DATA WORD
; 7668 MEM READ, [AR]_MEM, ;GO GET DATA WORD
; 7669 CLR IO LATCH, RETURN [1]
; 7670
; 7671 =0
; 7672 IOEAX: [AR]_[ARX]+WORK[YSAVE], ;GLOBAL INDEXING
; 7673 CLR IO LATCH, RETURN [1]
; 7674 [AR]_[ARX]+WORK[YSAVE] ;LOCAL INDEXING
; 7675 [AR]_O, HOLD RIGHT,
; 7676 CLR IO LATCH, RETURN [1]
; 7677
```

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 213
EXTERNAL IO INSTRUCTIONS

```
U 3671, 1130,3771,0016,4354,4007,0650,2000,0071,0200
; 7678 ;WAIT FOR IO TO COMPLETE
; 7679 ;RETURNS 1 OR PAGE FAILS
; 7680 ;
; 7681 IOWAIT: SC_S#, S#/200, ;DELAY
; 7682 [T0] VMA, ;GET VMA
; 7683 SKIP/-IO BUSY ;SEE IF BUSY YET
; 7684 =00
; 7685 IOW1: CLR IO LATCH, ;WENT BUSY
; 7686 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7687 J/IOW2 ;WAIT FOR IT TO CLEAR
; 7688 SC_SC-1, SCAD DISP, 5T, ;SEE IF DONE YET
; 7689 SKIP/-IO BUSY, ;
; 7690 J/IOW1 ;BACK TO LOOP
; 7691 CLR IO LATCH, ;WENT BUSY AND TIMEOUT
; 7692 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7693 J/IOW2 ;
; 7694 WORK[SV.VMA]_[TO], ;MAKE SURE SV.VMA IS SETUP
; 7695 J/IOW5 ;GO TRAP
; 7696
; 7697 IOW2: SC_S#, S#/777, ;GO TIME IO
; 7698 SKIP/-IO BUSY ;
; 7699 =0
; 7700 IOW3: CLR IO LATCH, ;TRY TO CLEAR LATCH
; 7701 STEP SC, J/IOW4 ;STILL BUSY
; 7702 RETURN [1] ;IDLE
; 7703
; 7704 =0
; 7705 IOW4: CLR IO LATCH, 5T, ;TRY TO CLEAR LATCH
; 7706 SKIP/-IO BUSY, ;SEE IF STILL BUSY
; 7707 J/IOW3 ;
; 7708 IOW5: [BRX]_[200000] XWD 0, J/HARD
; 7709
```


; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 214
SMALL SUBROUTINES

```
U 3673, 0001,3771,0003,4365,5004,1700,0200,0000,0002
U 3674, 0001,3771,0004,4365,5004,1700,0200,0000,0002
U 3675, 0001,3772,0000,4365,5004,1700,0200,0000,0002
U 3676, 0001,4221,0004,4174,0004,1700,0000,0000,0000
U 3677, 0001,0111,0703,4174,4004,1700,0000,0000,0000
U 3700, 0001,3445,0505,4174,4004,1700,0000,0000,0000
U 3701, 0001,3443,0300,4174,4004,1701,0200,0000,0036
U 3702, 0004,3333,0005,4175,5004,1701,0200,0000,0002
U 3703, 0001,3333,0001,4175,5004,1701,0200,0000,0002
U 3704, 0001,3440,0404,0174,4004,1700,0400,0000,0000

; 7710 .TOC "SMALL SUBROUTINES"
; 7711
; 7712 ;HERE ARE A COLLECTION ON 1-LINE SUBROUTINES
; 7713 LOADAR: MEM READ, [AR]_MEM, ;FROM MEMORY TO AR
; 7714 RETURN [1] ;RETURN TO CALLER
; 7715
; 7716 LOADARX: MEM READ, [ARX]_MEM, RETURN [1]
; 7717
; 7718 LOADQ: MEM READ, Q_MEM, RETURN [1]
; 7719
; 7720 CLARXL: [ARX]_0, HOLD RIGHT, RETURN [1]
; 7721
; 7722 INCAR: [AR]_[AR]+1, RETURN [1]
; 7723
; 7724 SBRL: [BR]_[BR]*2, RETURN [1]
; 7725
; 7726 STRTIO: VMA_[AR] WITH FLAGS, RETURN [1]
; 7727
; 7728 STOBR: MEM WRITE, MEM_[BR], RETURN [4]
; 7729
; 7730 STOPC: MEM WRITE, MEM_[PC], RETURN [1]
; 7731
; 7732 AC_ARX: AC_[ARX], RETURN [1]
; 7733
```

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-34 Page 215
UNDEFINED IO INSTRUCTIONS

D 0703, 0003, 1650, 2100
D 0706, 0006, 1650, 2100
D 0707, 0007, 1650, 2100

D 0716, 0006, 1651, 2100
D 0717, 0007, 1651, 2100

D 0726, 0006, 1652, 2100
D 0727, 0007, 1652, 2100

D 0730, 0003, 1653, 2100
D 0731, 0004, 1653, 2100
D 0732, 0005, 1653, 2100
D 0733, 0006, 1653, 2100
D 0734, 0007, 1653, 2100
D 0735, 0008, 1653, 2100
D 0736, 0009, 1653, 2100
D 0737, 0010, 1653, 2100

D 0740, 0003, 1654, 2100
D 0741, 0004, 1654, 2100
D 0742, 0005, 1654, 2100
D 0743, 0006, 1654, 2100
D 0744, 0007, 1654, 2100
D 0745, 0008, 1654, 2100
D 0746, 0009, 1654, 2100
D 0747, 0010, 1654, 2100

D 0750, 0000, 1655, 2100
D 0751, 0001, 1655, 2100
D 0752, 0002, 1655, 2100
D 0753, 0003, 1655, 2100
D 0754, 0004, 1655, 2100
D 0755, 0005, 1655, 2100
D 0756, 0006, 1655, 2100
D 0757, 0007, 1655, 2100

D 0760, 0000, 1656, 2100
D 0761, 0001, 1656, 2100
D 0762, 0002, 1656, 2100
D 0763, 0003, 1656, 2100
D 0764, 0004, 1656, 2100
D 0765, 0005, 1656, 2100
D 0766, 0006, 1656, 2100
D 0767, 0007, 1656, 2100

; 7734 .TOC "UNDEFINED IO INSTRUCTIONS"
; 7735
; 7736 .DCODE
; 7737 703: I, B/3, J/IOT700
; 7738 706: I, B/6, J/IOT700
; 7739 I, B/7, J/IOT700
; 7740
; 7741 716: I, B/6, J/IOT710
; 7742 I, B/7, J/IOT710
; 7743
; 7744 726: I, B/6, J/IOT720
; 7745 I, B/7, J/IOT720
; 7746
; 7747 730: I, B/0, J/IOT730
; 7748 I, B/1, J/IOT730
; 7749 I, B/2, J/IOT730
; 7750 I, B/3, J/IOT730
; 7751 I, B/4, J/IOT730
; 7752 I, B/5, J/IOT730
; 7753 I, B/6, J/IOT730
; 7754 I, B/7, J/IOT730
; 7755
; 7756 740: I, B/0, J/IOT740
; 7757 I, B/1, J/IOT740
; 7758 I, B/2, J/IOT740
; 7759 I, B/3, J/IOT740
; 7760 I, B/4, J/IOT740
; 7761 I, B/5, J/IOT740
; 7762 I, B/6, J/IOT740
; 7763 I, B/7, J/IOT740
; 7764
; 7765 750: I, B/0, J/IOT750
; 7766 I, B/1, J/IOT750
; 7767 I, B/2, J/IOT750
; 7768 I, B/3, J/IOT750
; 7769 I, B/4, J/IOT750
; 7770 I, B/5, J/IOT750
; 7771 I, B/6, J/IOT750
; 7772 I, B/7, J/IOT750
; 7773
; 7774 760: I, B/0, J/IOT760
; 7775 I, B/1, J/IOT760
; 7776 I, B/2, J/IOT760
; 7777 I, B/3, J/IOT760
; 7778 I, B/4, J/IOT760
; 7779 I, B/5, J/IOT760
; 7780 I, B/6, J/IOT760
; 7781 I, B/7, J/IOT760
; 7782

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-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

; 7783 770: I, B/0, J/IOT770
; 7784 I, B/1, J/IOT770
; 7785 I, B/2, J/IOT770
; 7786 I, B/3, J/IOT770
; 7787 I, B/4, J/IOT770
; 7788 I, B/5, J/IOT770
; 7789 I, B/6, J/IOT770
; 7790 I, B/7, J/IOT770
; 7791 .UCODE

U 1650, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7792
; 7793 1650:
; 7794 IOT700: UUD
; 7795 1651:
; 7796 IOT710:
; 7797 .IFNOT/UBABLT
; 7798 UUD

U 1651, 0670,4443,0000,4174,4007,0700,0000,0000,0000

; 7799 .IF/UBABLT
; 7800 J/BLTX
; 7801 .ENDIF/UBABLT

;GO TO COMMON CODE FOR UBABLT INSTRS

U 1652, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7802 1652:
; 7803 IOT720: UUD
; 7804 1653:

U 1653, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7805 IOT730: UUD
; 7806 1654:

U 1654, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7807 IOT740: UUD
; 7808 1655:

U 1655, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7809 IOT750: UUD
; 7810 1656:

U 1656, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7811 IOT760: UUD
; 7812 1657:

U 1657, 2732,4551,0202,4374,0007,0700,0000,0077,7740

; 7813 IOT770: UUD
; 7814

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-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, 3705,3443,0300,4174,4207,0700,0200,0004,0012

U 3705, 1515,3771,0003,4365,5007,0700,0200,0000,0002

U 1755, 3706,3443,0300,4174,4207,0700,0200,0003,0012

U 3706, 1516,3771,0003,0276,6007,0700,0000,0000,0000

```

; 7815 .TOC "UMOVE AND UMOVEM"
; 7816
; 7817 .DCODE
; 7818 704: IOT, J/UMOVE
; 7819 IOT, J/UMOVEM
; 7820 .UCODE
; 7821
; 7822 1754:
; 7823 UMOVE: VMA_[AR], ;LOAD VMA
; 7824 START READ, ;START MEMORY
; 7825 SPEC/PREV ;FORCE PREVIOUS
; 7826 MEM READ, ;WAIT FOR MEMORY
; 7827 [AR]_MEM, ;PUT DATA IN AR
; 7828 J/STAC ;GO PUT AR IN AC
; 7829
; 7830 1755:
; 7831 UMOVEM: VMA_[AR], ;LOAD VMA
; 7832 START WRITE, ;START MEMORY
; 7833 SPEC/PREV ;FORCE PREVIOUS
; 7834 [AR]_AC, ;FETCH AC
; 7835 J/STMEM ;STORE IN MEMORY
; 7836
```

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 218
UMOVE AND UMOVEM

U 0104, 3711,3333,0004,7174,4007,0700,0410,0000,0212
U 0114, 3710,4223,0000,4364,4277,0700,0210,0000,0010
U 0116, 3707,4221,0004,4174,4007,0700,0200,0021,1016
U 3707, 2514,3333,0017,4175,5007,0701,0200,0000,0002
U 2514, 3703,0111,0704,4170,4007,0700,0210,0023,1016
U 2515, 0005,4443,0000,4174,4107,0700,0000,0000,0074

U 0004, 2516,4443,0000,4174,4107,0640,0000,0000,0062
U 0005, 0004,4443,0000,4174,4007,0660,0000,0000,0000

U 2516, 0111,2447,0100,4174,4007,0700,0200,0014,0012
U 2517, 2517,4443,1203,4374,4007,0700,0000,0024,1200

U 2520, 3707,3561,0303,4070,4007,0700,0010,0020,0000

U 2521, 2711,3771,0002,4365,5617,0700,0200,0000,0002

```
; 7837 ;HERE WITH HALT CODE IN THE T1
; 7838 =010*
; 7839 HALTED: WORK[SV.ARX] [ARX], ;SAVE TEMP REGISTER
; 7840 CALL [SAVVMA] ;PUT VMA IN WORK[SV.VMA]
; 7841 =110* ABORT MEM CYCLE, ;ABORT CYCLE IN PROGRESS
; 7842 CALL [WRTHSB] ;WRITE HALT STATUS BLOCK
; 7843 =111*
; 7844 PWRON: [ARX]_O, VMA PHYSICAL WRITE ;STORE HALT CODE
; 7845 =
; 7846 MEM WRITE, MEM [T1] ; IN LOCATION O
; 7847 =0 NEXT [ARX] PHYSICAL WRITE,
; 7848 CALL [STOPC]
; 7849 H1: SET HALT, J/HALTLP ;TELL CONSOLE WE HAVE HALTED
; 7850
; 7851
; 7852 4: UNHALT, ;RESET CONSOLE
; 7853 SKIP EXECUTE, J/CONT ;SEE IF CO OR EX
; 7854 5:
; 7855 HALTLP: SKIP/-CONTINUE, J/4 ;WAIT FOR CONTINUE
; 7856
; 7857 =0
; 7858 CONT: VMA [PC], ;LOAD PC INTO VMA
; 7859 FETCH, ;START READ
; 7860 J/XCTGO ;DO THE INSTRUCTION
; 7861 [AR] VMA IO READ ;PUT FLAGS IN AR
; 7862 =0 [AR]_[AR].OR.#, ;PUT IN ADDRESS
; 7863 #/200000, HOLD LEFT, ; OF CSL REGISTER
; 7864 CALL [STRTIO]
; 7865 CONT1: MEM READ, ;WAIT FOR DATA
; 7866 [HR] MEM, ;PUT IN HR
; 7867 LOAD INST, ;LOAD IR, ETC.
; 7868 J/XCT1 ;GO DO THE INSTRUCTION
; 7869
```

; T1OKL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 219
WRITE HALT STATUS BLOCK

; 7870 .TOC "WRITE HALT STATUS BLOCK"
; 7871
; 7872 ;THE HALT STATUS BLOCK LOOKS LIKE:
; 7873

```

; 7874 ; |=====|
; 7875 ; |00!          MAG          |
; 7876 ; |-----|
; 7877 ; |01!          PC          |
; 7878 ; |-----|
; 7879 ; |02!          HR          |
; 7880 ; |-----|
; 7881 ; |03!          AR          |
; 7882 ; |-----|
; 7883 ; |04!          ARX         |
; 7884 ; |-----|
; 7885 ; |05!          BR          |
; 7886 ; |-----|
; 7887 ; |06!          BRX         |
; 7888 ; |-----|
; 7889 ; |07!          ONE        |
; 7890 ; |-----|
; 7891 ; |10!          EBR         |
; 7892 ; |-----|
; 7893 ; |11!          UBR         |
; 7894 ; |-----|
; 7895 ; |12!          MASK       |
; 7896 ; |-----|
; 7897 ; |13!          FLG        |
; 7898 ; |-----|
; 7899 ; |14!          PI         |
; 7900 ; |-----|
; 7901 ; |15!          XWD1      |
; 7902 ; |-----|
; 7903 ; |16!          TO         |
; 7904 ; |-----|
; 7905 ; |17!          T1        |
; 7906 ; |=====|
; 7907 ; |          VMA FLAGS      !          VMA          |
; 7908 ; |=====|
; 7909
```

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 220
WRITE HALT STATUS BLOCK

U 0001, 3711, 3333, 0004, 7174, 4007, 0700, 0410, 0000, 0212
U 0011, 0024, 3771, 0004, 7274, 4007, 0701, 0000, 0000, 0227
U 0024, 3713, 4223, 0000, 4364, 4277, 0700, 0210, 0000, 0010
U 0026, 2515, 4443, 0000, 4174, 4107, 0700, 0000, 0000, 0074

U 3710, 2522, 3771, 0004, 7274, 4007, 0422, 0000, 0000, 0227
U 2522, 3713, 3333, 0012, 4174, 4437, 0700, 0000, 0000, 0000
U 2523, 0002, 3771, 0004, 7274, 4004, 1701, 0000, 0000, 0212
U 3711, 3712, 3771, 0004, 4354, 4007, 0700, 0000, 0000, 0000
U 3712, 0010, 3333, 0004, 7174, 4004, 1700, 0400, 0000, 0210

; 7910 ;START AT 1 TO DUMP 2901 REGISTERS INTO MAIN MEMORY
; 7911 1: WORK[SV.ARX]_[ARX], ;SAVE TEMP REGISTER
; 7912 CALL [SAVVMA] ;WORK[SV.VMA]_VMA
; 7913 11: [ARX]_WORK[HSBADR]
; 7914 =10* ABORT MEM CYCLE, CALL [DUMP]
; 7915 SET HALT, J/H1
; 7916
; 7917
; 7918 WRTHSB: [ARX]_WORK[HSBADR], ;GET ADDRESS OF HSB
; 7919 SKIP AD.LE.O, 4T ;SEE IF VALID
; 7920 =0 READ [MASK], LOAD PI, ;TURN OFF PI SYSTEM
; 7921 J/DUMP ; AND GO TAKE DUMP
; 7922 [ARX]_WORK[SV.ARX],
; 7923 RETURN [2] ;DO NOT DUMP ANYTHING
; 7924
; 7925 SAVVMA: [ARX]_VMA
; 7926 WORK[SV.VMA]_[ARX],
; 7927 RETURN [10]
; 7928

; T10KL.MCR[10,1141]
; INOUT.MIC[10,1141]

15:34 27-JULY-1984
09:17 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 221
WRITE HALT STATUS BLOCK

```
U 3713, 2375,3333,0004,4174,4007,0700,0200,0021,1016 ; 7929 ;DUMP OUT THE 2901
U 2375, 2741,3333,0000,4175,5007,0701,0210,0000,0002 ; 7930 DUMP: READ [ARX], VMA PHYSICAL WRITE
U 2377, 3714,3333,0001,4175,5007,0701,0200,0000,0002 ; 7931 =O* MEM WRITE, MEM_[MAG], CALL [NEXT]
U 3714, 2524,0111,0704,4170,4007,0700,0200,0023,1016 ; 7932 MEM WRITE, MEM_[PC]
U 2524, 2741,3333,0002,4175,5007,0701,0210,0000,0002 ; 7933 NEXT [ARX] PHYSICAL WRITE
U 2526, 2525,3333,0003,4175,5007,0701,0200,0000,0002 ; 7934 =O* MEM WRITE, MEM_[HR], CALL [NEXT]
U 2525, 2741,3333,0003,7174,4007,0700,0410,0000,0211 ; 7935 MEM WRITE, MEM_[AR]
U 2527, 2530,3771,0003,7274,4007,0701,0000,0000,0212 ; 7936 =O* WORK[SV.AR]_[AR], CALL [NEXT]
U 2530, 2741,3333,0003,4175,5007,0701,0210,0000,0002 ; 7937 [AR]_WORK[SV.ARX]
U 2532, 3715,3333,0005,4175,5007,0701,0200,0000,0002 ; 7938 =O* MEM WRITE, MEM_[AR], CALL [NEXT]
U 3715, 2531,0111,0704,4170,4007,0700,0200,0023,1016 ; 7939 MEM WRITE, MEM_[BR]
U 2531, 2741,3333,0006,4175,5007,0701,0210,0000,0002 ; 7940 NEXT [ARX] PHYSICAL WRITE
U 2533, 3716,3333,0007,4175,5007,0701,0200,0000,0002 ; 7941 =O* MEM WRITE, MEM_[BRX], CALL [NEXT]
U 3716, 2534,0111,0704,4170,4007,0700,0200,0023,1016 ; 7942 MEM WRITE, MEM_[ONE]
U 2534, 2741,3333,0010,4175,5007,0701,0210,0000,0002 ; 7943 NEXT [ARX] PHYSICAL WRITE
U 2536, 3717,3333,0011,4175,5007,0701,0200,0000,0002 ; 7944 =O* MEM WRITE, MEM_[EBR], CALL [NEXT]
U 3717, 2535,0111,0704,4170,4007,0700,0200,0023,1016 ; 7945 MEM WRITE, MEM_[UBR]
U 2535, 2741,3333,0012,4175,5007,0701,0210,0000,0002 ; 7946 NEXT [ARX] PHYSICAL WRITE
U 2537, 3720,3333,0013,4175,5007,0701,0200,0000,0002 ; 7947 =O* MEM WRITE, MEM_[MASK], CALL [NEXT]
U 3720, 2540,0111,0704,4170,4007,0700,0200,0023,1016 ; 7948 MEM WRITE, MEM_[FLG]
U 2540, 2741,3333,0014,4175,5007,0701,0210,0000,0002 ; 7949 NEXT [ARX] PHYSICAL WRITE
U 2542, 3721,3333,0015,4175,5007,0701,0200,0000,0002 ; 7950 =O* MEM WRITE, MEM_[PI], CALL [NEXT]
U 3721, 2541,0111,0704,4170,4007,0700,0200,0023,1016 ; 7951 MEM WRITE, MEM_[XWD1]
U 2541, 2741,3333,0016,4175,5007,0701,0210,0000,0002 ; 7952 NEXT [ARX] PHYSICAL WRITE
U 2543, 2544,3333,0017,4175,5007,0701,0200,0000,0002 ; 7953 =O* MEM WRITE, MEM_[TO], CALL [NEXT]
U 2544, 2741,3771,0003,7274,4007,0701,0010,0000,0210 ; 7954 MEM WRITE, MEM_[T1]
U 2546, 3722,3333,0003,4175,5007,0701,0200,0000,0002 ; 7955 =O* [AR]_WORK[SV.VMA], CALL [NEXT]
U 3722, 3723,3771,0003,7274,4007,0701,0000,0000,0211 ; 7956 MEM WRITE, MEM_[AR]
U 3723, 3724,3771,0004,7274,4007,0701,0000,0000,0210 ; 7957 HSBDON: [AR]_WORK[SV.AR]
U 3724, 3725,3443,0400,4174,4007,0700,0200,0000,0010 ; 7958 [ARX]_WORK[SV.VMA]
; 7959 VMA [ARX]
; 7960 [ARX]_WORK[SV.ARX],
; 7961 RETURN [6]
; 7962
```

Produced on Advanced Information Services Electronic Laser Printer, PKO/IE85, DTN: 223-7881


```

; 7963 .NOBIN
; 7964 .TOC "PAGE FAIL REFIL LOGIC"
; 7965
; 7966 ;WHEN THE CPU CAN NOT COMPLETE A MEMORY REFERENCE BECAUSE THE PAGE
; 7967 ; TABLE DOES NOT CONTAIN VALID INFORMATION FOR THE VIRTUAL PAGE INVOLVED
; 7968 ; THE HARDWARE CALLS THIS ROUTINE TO RELOAD THE HARDWARE PAGE TABLE.
; 7969 ;
; 7970 ;THIS CODE WILL EITHER DO THE RELOAD OR GENERATE A PAGE FAIL FOR THE
; 7971 ; SOFTWARE. THE INFORMATION LOADED CONSISTS OF THE PHYSICAL PAGE NUMBER,
; 7972 ; THE CACHE ENABLE BIT AND THE WRITE ENABLE BIT.
; 7973
; 7974 ;THIS LOGIC USES MANY VARIABLES. THEY ARE DESCRIBED BRIEFLY HERE:
; 7975
; 7976 ;THING WHERE KEPT USE
; 7977 ;OLD VMA WORKSPACE WORD 210 SAVES VMA
; 7978 ;OLD AR WORKSPACE WORD 211 SAVES AR
; 7979 ;OLD ARX WORKSPACE WORD 212 SAVES ARX
; 7980 ;OLD BR WORKSPACE WORD 213 SAVES BR
; 7981 ;OLD BRX WORKSPACE WORD 214 SAVES BRX
; 7982 ;KL PAGING BIT EBR BIT 1 (IN 2901) INDICATES KL STYLE (TOPS-20) PAGING
; 7983 ; INSTEAD OF KI STYLE (TOPS-10 AND DIAGNOSTIC)
; 7984 ; MODE PAGING
; 7985 ;W BIT FLG BIT 4 PAGE CAN BE WRITTEN
; 7986 ;C BIT FLG BIT 6 DATA IN THIS PAGE MAY BE PUT
; 7987 ; INTO CACHE
; 7988 ;PI CYCLE FLG BIT 5 STORING OLD PC DURING PI
; 7989 ;MAP FLAG FLG BIT 18 MAP INSTRUCTION IN PROGRESS
; 7990 ;CLEANUP CODE FLG BITS 32-35 WHAT TO DO SO INSTRUCTION MAY BE
; 7991 ; RESTARTED
; 7992 ;SPT BASE WORKSPACE WORD 215 ADDRESS OF SHARED-POINTER-TABLE
; 7993 ;CST BASE WORKSPACE WORD 216 ADDRESS OF CORE-STATUS-TABLE
; 7994 ;CST MASK WORKSPACE WORD 217 BITS TO KEEP ON CST UPDATE
; 7995 ;CST DATA (PUR) WORKSPACE WORD 220 BITS TO SET ON CST UPDATE
; 7996 ;PAGE TABLE ADDRESS AR WHERE THIS PAGE TABLE IS LOCATED
; 7997 ;PHYSICAL PAGE # (PPN) AR RESULT OF THIS PROCESS
; 7998 ;CST ENTRY AR CORE STATUS TABLE ENTRY
; 7999 ;SPT ENTRY AR WORD FROM SPT
; 8000 ;PAGE TABLE ENTRY AR WORD FROM PT
; 8001 ;PAGE NUMBER BR INDEX INTO CURENT PAGE TABLE
; 8002 ;PAGE FAIL WORD BRX WHAT HAPPENED (ALSO MAP RESULT)
; 8003
; 8004 .IF/INHCST
; 8005 SKIP NO CST "AD/D,DBUS/RAM,RAMADR/#,WORK/CBR,DT/4T,SKIP/ADEQO"
; 8006 .ENDIF/INHCST
; 8007

```

```

; 8008 ;
; 8009 ;
; 8010 ;
; 8011 ;
; 8012 ;
; 8013 ;Section Pointer
; 8014 ;
; 8015 ;The section pointer is found in the user or exec section table.
; 8016 ;(Part of UPT or EPT.)
; 8017 ;
; 8018 ;Section pointer provides (via the SPT) the physical address of
; 8019 ;the PAGE TABLE for the given section.
; 8020 ;

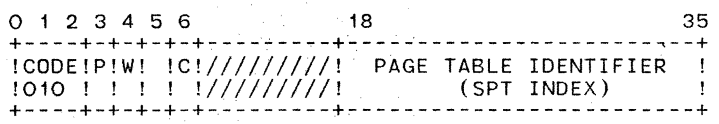
```

KL10 PAGING - WORD FORMATS

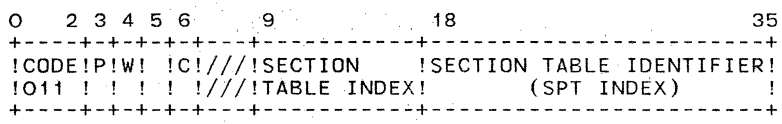
```

; 8021 ; Code: 0 No-access (trap)
; 8022 ; 1 Immediate
; 8023 ; 2 Share
; 8024 ; 3 Indirect
; 8025 ; 4-7 Unused, reserved
; 8026 ;

```



NORMAL SECTION POINTER (Code = 2)



INDIRECT SECTION POINTER (Code = 3)

```

; 8040 ;
; 8041 ;
; 8042 ;

```

```
; 8043 ;PAGE POINTERS
; 8044 ;
; 8045 ;FOUND IN PAGE TABLES
; 8046 ;
; 8047 ;      0 1 2 3 4 5 6      12      35
; 8048 ;      +-----+-----+-----+
; 8049 ;      !CODE!PIW! !C!/////!   PHYSICAL ADDRESS OF PAGE !
; 8050 ;      !001 ! ! ! ! !/////!   !
; 8051 ;      +-----+-----+-----+
; 8052 ;
; 8053 ;          IMMEDIATE POINTER (code field = 1)
; 8054 ;
; 8055 ;      B12-35 give PHYSICAL ADDRESS OF PAGE
; 8056 ;          if B12-17 >< 0, page not in core-trap
; 8057 ;          if B12-17 = 0, B23-35 give CORE PAGE
; 8058 ;                      NUMBER of page, B18-22 MBZ
; 8059 ;
; 8060 ;
; 8061 ;
; 8062 ;
; 8063 ;
; 8064 ;      0      2 3      6      18      35
; 8065 ;      +-----+-----+-----+
; 8066 ;      !CODE !SAME AS!/////!   SPT INDEX
; 8067 ;      !010 ! IMMED.!/////!   !
; 8068 ;      +-----+-----+-----+
; 8069 ;
; 8070 ;          SHARED POINTER (code field = 2)
; 8071 ;
; 8072 ;      B18-35 Give SPT INDEX (SPTX). SPTX + SPT BASE
; 8073 ;      ADDRESS = physical core address of word
; 8074 ;      holding physical address of page.
; 8075 ;
```

Produced on Advanced Information Services Electronic Laser Printer, PK011556, DTN. 223-7881

; 8076 ; 0 1 2 3 6 9 17 18 35
; 8077 ; +-----+-----+-----+-----+
; 8078 ; !CODE!SAME AS !///
; 8079 ; !O11 ! IMMED. !///
; 8080 ; +-----+-----+-----+-----+
; 8081 ;

INDIRECT POINTER (code field = 3)

This pointer type causes another pointer to be fetched and interpreted. The new pointer is found in word N (B9-17) of the page addressed by C(SPT.+ SPTX).

SPT ENTRY

Found in the SPT, i.e., when fetching C(SPT +SPTX)

; 8092 ; 12 35
; 8093 ; +-----+-----+
; 8094 ; !///
; 8095 ; !///
; 8096 ; !///
; 8097 ; !///
; 8098 ; !///
; 8099 ; +-----+-----+
; 8100 ;

B12-35 Give PHYSICAL ADDRESS of page.

The base address (physical core address) of the SPT resides in one AC of the reserved AC block.

; 8101 ;
; 8102 ;
; 8103 ;
; 8104 ;

; 8105 ;PHYSICAL STORAGE ADDRESS
; 8106 ;
; 8107 ;Found in B12-35 of IMMEDIATE POINTERS and SPT ENTRIES.
; 8108 ;
; 8109 ; 12 17 18 23 35
; 8110 ; +-----+-----+-----+
; 8111 ; ! !MBZ ! CORE PAGE NUMBER!
; 8112 ; ! ! ! IF B12-17 = 0 !
; 8113 ; +-----+-----+-----+
; 8114 ;
; 8115 ; If B12-17 = 0, then B23-35 are CORE PAGE NUMBER (i.e.,
; 8116 ; B14-26 of physical core address) of page and B18-22
; 8117 ; MBZ. If B12-17 > 0, then address is not core and
; 8118 ; pager traps.
; 8119 ;
; 8120 ;
; 8121 ;
; 8122 ;CORE STATUS TABLE ENTRY
; 8123 ;
; 8124 ;Found when fetching C(CBR + CORE PAGENO)
; 8125 ;
; 8126 ; 0 5 32 34 35
; 8127 ; +-----+-----+-----+
; 8128 ; ! CODE ! ! !M!
; 8129 ; +-----+-----+-----+
; 8130 ;
; 8131 ; B0-5 are code field:
; 8132 ;
; 8133 ; 0 - unavailable, trap
; 8134 ;
; 8135 ; 1-77 - available
; 8136 ;
; 8137 ;
; 8138 ;
; 8139 ; B32-34 reserved for future hardware specification.
; 8140 ;
; 8141 ; B35 is "modified" bit, set on any write ref to page.
; 8142 ;

```

; 8143 ;QUANTITIES IN HARDWARE REGISTERS
; 8144 ;
; 8145 ;SPT      SPT Base Register
; 8146 ;
; 8147 ;          14                      35
; 8148 ;          +-----+
; 8149 ;          !   PHYSICAL CORE WORD ADDRESS   !
; 8150 ;          +-----+
; 8151 ;
; 8152 ;CBR      CST Base Register
; 8153 ;
; 8154 ;          14                      35
; 8155 ;          +-----+
; 8156 ;          !   PHYSICAL CORE WORD ADDRESS   !
; 8157 ;          +-----+
; 8158 ;
; 8159 ;CSTMSK  CST Update Mask
; 8160 ;
; 8161 ;          0                      32 35
; 8162 ;          +-----+-----+
; 8163 ;          !                      MASK          !1111!
; 8164 ;          +-----+-----+
; 8165 ;
; 8166 ;          ANDed with CST word during update
; 8167 ;
; 8168 ;(B32-35 must be all 1's to preserve existing CST information)
; 8169 ;
; 8170 ;CSTDATA  CST Update Data
; 8171 ;
; 8172 ;          0                      32 34 35
; 8173 ;          +-----+-----+
; 8174 ;          !                      DATA          !000!0!
; 8175 ;          +-----+-----+
; 8176 ;
; 8177 ;          IORed with CST word during update
; 8178 ;
; 8179 ;(B32-35 must be all 0's to preserve existing CST information)
; 8180 ;
; 8181 ;All unspecified bits and fields are reserved for future
; 8182 ;specification by DEC.
; 8183 ;
; 8184 ;

```

; T10KL.MIC[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 228
PAGE FAIL REFIL LOGIC

D 0257, 1215, 1553, 0100

U 1553, 3726, 3551, 0303, 4374, 0007, 0700, 0000, 0016, 0000
U 3726, 3727, 3771, 0006, 4354, 4007, 0700, 0000, 0000, 0000

U 3727, 3730, 4551, 0606, 4374, 0007, 0700, 0000, 0040, 0000
U 3730, 3731, 3333, 0006, 7174, 4007, 0700, 0400, 0000, 0210
U 3731, 3732, 3771, 0005, 7274, 4007, 0701, 0000, 0000, 0230
U 3732, 2550, 4553, 0500, 4374, 4007, 0331, 0000, 0003, 0000
U 2550, 3747, 3771, 0013, 4370, 4007, 0700, 0000, 0040, 0002
U 2551, 0100, 3440, 0303, 0174, 4156, 4700, 0400, 0000, 0000

U 3777, 3733, 3333, 0003, 7174, 4007, 0700, 0400, 0000, 0211
U 3733, 3734, 3333, 0006, 7174, 4007, 0700, 0400, 0000, 0214
U 3734, 3735, 3771, 0006, 4354, 4007, 0700, 0000, 0000, 0000
U 3735, 3736, 3333, 0006, 7174, 4007, 0700, 0400, 0000, 0210

U 3736, 1060, 3333, 0004, 7174, 4007, 0370, 0400, 0000, 0212

U 1060, 1060, 3773, 0000, 4304, 4003, 1702, 0000, 0000, 0000

U 1061, 1572, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213

U 1063, 3737, 3771, 0006, 4374, 4007, 0700, 0000, 0000, 0000

U 1065, 1563, 4571, 1206, 4374, 4007, 0700, 0000, 0037, 0000

U 1067, 1563, 4571, 1206, 4374, 4007, 0700, 0000, 0037, 0000

U 1070, 3747, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213

U 1071, 1572, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213

U 1072, 3747, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213

U 1073, 3747, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213

U 3737, 3745, 3333, 0006, 7174, 4007, 0700, 0400, 0000, 0160

U 3745, 3746, 3333, 0006, 7174, 4007, 0700, 0400, 0000, 0161

```
; 8185 .BIN
; 8186
; 8187 .DCODE
; 8188 257: IOT, AC, J/MAP
; 8189 .UCODE
; 8190
; 8191 1553:
; 8192 MAP: [AR]_[AR].OR.#, ;ASSUME PHYSICAL REF
; 8193 #/160000, ;FAKE ANSWER
; 8194 HOLD RIGHT ;
; 8195 [BRX]_VMA ;PUT VMA AND FLAGS IN BRX
; 8196 [BRX]_[BRX].AND.#, ;JUST KEEP USER BIT
; 8197 #/400000, HOLD RIGHT ;
; 8198 WORK[SV.VMA]_[BRX] ;SAVE IN WORKSPACE
; 8199 [BR] WORK[APR] ;GET APR FLAGS
; 8200 TR [BR], #/030000 ;PAGING ENABLED?
; 8201 =0 STATE [MAP], J/PFMAP ;YES--DO REAL MAP
; 8202 AC [AR], NEXT INST ;NO--RETURN VIRTUAL ADDRESS
; 8203 ;HARDWARE COMES HERE ON PAGE TABLE NOT VALID (OR INTERRUPT) WHEN
; 8204 ; STARTING A MEMORY REFERENCE. MICROWORD ADDRESS OF INSTRUCTION DOING
; 8205 ; MEM WAIT IS SAVED ON THE STACK.
; 8206 3777:
; 8207 PAGE-FAIL:
; 8208 WORK[SV.AR]_[AR]
; 8209 ITRAP: WORK[SV.BRX]_[BRX]
; 8210 [BRX]_VMA
; 8211 WORK[SV.VMA]_[BRX]
; 8212 WORK[SV.ARX]_[ARX],
; 8213 SKIP IRPT ;SEE IF INTERRUPT (SAVE DISPATCH)
; 8214 =0000
; 8215 PFD: DBM/PF DISP, DBUS/DBM, ;BRING CODE TO 2901'S
; 8216 AD/D, DEST/PASS, 4T, ;PUT ON DP 18-21
; 8217 DISP/DP LEFT, J/PFD ;DISPATCH ON IT
; 8218 =0001 ;(1) INTERRUPT
; 8219 WORK[SV.BR]_[BR], J/PFPI1
; 8220 =0011 ;(3) BAD DATA FROM MEMORY
; 8221 [BRX]_IO DATA, ;GET THE BAD DATA
; 8222 AD PARITY OK/O, ; DO NOT LOOK AT PARITY
; 8223 J/BADDATA ;SAVE IN AC BLK 7
; 8224 =0101 ;(5) NXM ERROR
; 8225 [BRX]_[370000] XWD 0, J/HARD
; 8226 =0111 ;(7) NXM & BAD DATA
; 8227 [BRX]_[370000] XWD 0, J/HARD
; 8228 =1000 ;(10) WRITE VIOLATION
; 8229 WORK[SV.BR]_[BR], J/PFMAP
; 8230 =1001 ;[123] (11) 1 ms timer and movsrj
; 8231 WORK[SV.BR]_[BR], J/PFPI1
; 8232 =1010 ;(12) PAGE NOT VALID
; 8233 WORK[SV.BR]_[BR], J/PFMAP
; 8234 =1011 ;(13) EXEC/USER MISMATCH
; 8235 WORK[SV.BR]_[BR], J/PFMAP
; 8236 =
; 8237
; 8238 BADATA:
; 8239 WORK[BADWO]_[BRX] ;SAVE BAD WORD
; 8240 WORK[BADW1]_[BRX] ;AGAIN
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984 MICRO 31(254) KS10 MICROCODE V124, 27-JUL-84 Page 228-1
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984 PAGE FAIL REFIL LOGIC

U 3746, 1563,4571,1206,4374,4007,0700,0000,0036,0000 ; 8241 [BRX]_[360000] XWD 0, J/HARD
; 8242

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 229
PAGE FAIL REFIL LOGIC

U 3747, 3750,4223,0000,4364,4277,0700,0200,0000,0010

U 3750, 3752,3551,1313,4374,0007,0700,0000,0002,4000
U 3752, 2552,4553,0600,4374,4007,0321,0000,0002,0000

U 2552, 2553,3551,0606,4374,0007,0700,0000,0001,0000

U 2553, 3753,4551,0606,4374,0007,0700,0000,0041,1000

U 3753, 3755,6551,0606,4374,0007,0700,0000,0000,1000

U 3755, 2554,3441,0605,4174,4007,0700,2000,0071,0007

U 2554, 2554,3447,0505,4174,4007,0630,2000,0060,0000

U 2555, 3756,4251,0505,4374,4007,0700,0000,0000,0777

U 3756, 2556,3333,0006,4174,4007,0520,0000,0000,0000

```
; 8243 ;WE HAVE SAVED AR, ARX, BR AND BRX. WE MERGE IN HERE FROM MAP
; 8244 ; INSTRUCTION, SAVE THE VMA AND START THE PAGE FAIL WORD.
; 8245 PFMAP: ABORT MEM CYCLE ;CLEAR PAGE FAIL
; 8246 [FLG]_[FLG].OR.#, ;PRESET W AND C TO 1
; 8247 FLG.W/1, FLG.C/1, ;BITS INVOLVED
; 8248 HOLD RIGHT ;LEAVE RH ALONE
; 8249 TL [BRX], WRITE TEST/1 ;IS THIS A WRITE TEST?
; 8250 =0 [BRX]_[BRX].OR.#,
; 8251 #/10000,
; 8252 HOLD RIGHT ;YES--TURN INTO WRITE REF
; 8253 [BRX]_[BRX].AND.#, ;START PAGE FAIL WORD
; 8254 #/411000, ;SAVE 3 INTERESTING BITS
; 8255 HOLD RIGHT ;SAVE VIRTUAL ADDRESS
; 8256 ;USER ADDR (400000)
; 8257 ;WRITE REF (010000)
; 8258 ;PAGED REF (001000)
; 8259 [BRX]_[BRX].XOR.#, ;FIX BIT 8
; 8260 #/1000, HOLD RIGHT
; 8261 [BR]_[BRX], ;COPY VIRTUAL ADDRESS
; 8262 SC_7 ;PREPARE TO SHIFT 9 PLACES
; 8263 =0
; 8264 PF25: [BR]_[BR]*.5, ;RIGHT ADJUST PAGE #
; 8265 STEP_SC, ;COUNT SHIFT STEPS
; 8266 J/PF25 ;LOOP FOR 9
; 8267 [BR]_[BR].AND.# CLR LH, ;MASK TO 9 BITS
; 8268 #/777 ;...
; 8269 .IF/KLPAGE
; 8270 .IF/KIPAGE
; 8271 TL [EBR], ;KI MODE REFILL?
; 8272 #/40 ;FLAG BIT
; 8273 =0
; 8274 .ENDIF/KIPAGE
; 8275 READ [BRX], ;USER REF? (KL MODE)
; 8276 SKIP DPO,
; 8277 J/PF30 ;CONTINUE AT PF30
; 8278 .ENDIF/KLPAGE
; 8279 .IF/KIPAGE
; 8280 [ARX]_[BR]*.5, ;KI10 MODE REFILL
; 8281 J/KIFILL ;GO HANDLE EASY CASE
; 8282 .ENDIF/KIPAGE
; 8283
```

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 230
PAGE FAIL REFIL LOGIC

U 2556, 2562,3771,0004,7274,4007,0622,0000,0000,0423
U 2557, 2560,3771,0004,7274,4007,0622,0000,0000,0424
U 2560, 1134,0113,0405,4174,4007,0700,0200,0024,1016
U 2561, 3757,0551,1103,4374,4007,0701,0000,0000,0540
U 2562, 1134,0113,0405,4174,4007,0700,0200,0024,1016
U 2563, 3757,0551,1003,4374,4007,0701,0000,0000,0540
U 3757, 3760,3443,0300,4174,4007,0700,0200,0024,1016
U 3760, 1000,3771,0003,4365,5007,0700,0200,0000,0002
U 1000, 1503,4443,0000,4174,4007,0700,2010,0071,0007
U 1001, 2566,4553,0300,4374,4007,0321,0000,0000,0077
U 1002, 2243,0551,0303,7274,4007,0701,0000,0000,0215
U 1003, 1511,0551,0303,7274,4007,0701,0010,0000,0215
U 1007, 2564,4553,0300,4374,4007,0321,0000,0000,0077
U 2564, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 2565, 1010,3445,0303,4174,4007,0630,2000,0060,0000
U 1010, 1010,3445,0303,4174,4007,0630,2000,0060,0000
U 1011, 1511,4443,0000,4174,4007,0700,0010,0000,0000
U 1015, 1000,4443,0000,4174,4007,0700,0000,0000,0000

```
; 8284 .IF/KLPAGE
; 8285 ;HERE IN TOPS-20 MODE
; 8286 ;PICK UP CORRECT SECTION POINTER
; 8287 =0
; 8288 PF30: [ARX]_WORK[PTA.E], ;EXEC MODE
; 8289 SKIP AD.EQ.O, 4T, ;SEE IF VALID
; 8290 J/PF35 ;CONTINUE BELOW
; 8291 [ARX]_WORK[PTA.U], ;USER MODE
; 8292 SKIP AD.EQ.O, 4T ;SEE IF VALID
; 8293 =0 VMA_[ARX]+[BR], ;POINTER VALID
; 8294 VMA_PHYSICAL READ, ;START MEMORY
; 8295 J/PF77 ;CONTINUE BELOW
; 8296 [AR]_[UBR]+#, 3T, ;USER MODE
; 8297 #/540, ;OFFSET TO UPT
; 8298 J/PF40 ;GO GET POINTER
; 8299
; 8300 =0
; 8301 PF35: VMA_[ARX]+[BR], ;POINTER VALID
; 8302 VMA_PHYSICAL READ, ;START MEMORY
; 8303 J/PF77 ;CONTINUE BELOW
; 8304 [AR]_[EBR]+#, 3T, ;EXEC MODE
; 8305 #/540 ;OFFSET TO EPT
; 8306 PF40: VMA_[AR], ;LOAD THE VMA
; 8307 START READ, ;START THE MEMORY CRANKING
; 8308 VMA_PHYSICAL ;ABSOLUTE ADDRESS
; 8309 MEM READ, ;WAIT FOR MEMORY
; 8310 [AR] MEM ;POINT POINTER IN AR
; 8311 ;LOOK AT SECTION POINTER AND DISPATCH ON TYPE
; 8312 =000
; 8313 PF45: SC_7, ;FETCH SECTION 0 POINTER
; 8314 CALL [SETPTR] ;FIGURE OUT POINTER TYPE
; 8315 SECIMM: TL [AR], ;IMMEDIATE POINTER
; 8316 #/77, ;TEST FOR 12-17 = 0
; 8317 J/PF50 ;CONTINUE AT PF50
; 8318 [AR]_[AR]+WORK[SBR], ;SHARED SECTION
; 8319 J/SECshr ;GO FETCH POINTER FROM SPT
; 8320 [AR]_[AR]+WORK[SBR], ;INDIRECT SECTION POINTER
; 8321 CALL [RDPT] ;GO FETCH SPT ENTRY
; 8322 =111 TL [AR], ;12 TO 17 = 0?
; 8323 #/77 ; ..
; 8324 =
; 8325 =0 PAGE FAIL TRAP ;NO
; 8326 [AR]_[AR]*2, ;FIRST SHIFT
; 8327 STEP SC ;SC WAS LOADED AT PF45
; 8328 =0*0
; 8329 PF60: [AR]_[AR]*2, ;CONVERT TO ADDRESS OF
; 8330 STEP SC, ;SECTION TABLE
; 8331 J/PF60
; 8332 CALL [RDPT] ;READ SECTION TABLE
; 8333 =1*1 J/PF45 ;TRY AGAIN
; 8334 =
; 8335
```

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 231
PAGE FAIL REFIL LOGIC

U 2243, 1511,4443,0000,4174,4007,0700,0010,0000,0000
U 2247, 2566,4553,0300,4374,4007,0321,0000,0000,0077

U 2566, 2670,4553,1300,4374,4007,0321,0000,0001,0000

U 2567, 2545,4251,0304,4374,4007,0700,0000,0000,3777

U 2545, 1502,3441,0403,4174,4007,0700,0010,0000,0000

U 2547, 2570,3551,0303,7274,4007,0701,0000,0000,0220

U 2570, 2634,3773,0000,7274,4007,0622,0010,0000,0216

U 2574, 2572,4443,0000,4174,4007,0700,2000,0071,0007

U 2572, 2572,3445,0404,4174,4007,0630,2000,0060,0000

```
; 8336 ;STILL .IF/KLPAGE
; 8337 ;HERE FOR SHARED SECTION. AR GETS THE ADDRESS OF PAGE TABLE
; 8338 =O**
; 8339 SECshr: CALL [RDPT] ;READ WORD FROM SPT
; 8340 TL [AR], #/77 ;TEST FOR BITS 12-17 = 0
; 8341
; 8342 ;HERE WITH ADDRESS OF PAGE TABLE IN AR AND SKIP ON
; 8343 ; BITS 12 THRU 17 EQUAL TO ZERO
; 8344 =0
; 8345 PF50: PAGE FAIL TRAP ;BITS 12-17 .NE. 0
; 8346 [AR]_[AR].AND.# CLR LH, ;PAGE NUMBER OF PAGE TABLE
; 8347 #/3777 ;11 BIT PHYSICAL PAGE #
; 8348 .IFNOT/NOCST
; 8349 =O* [AR]_[AR], ;COPY ADDRESS
; 8350 CALL [UPCST] ;UPDATE CSTO
; 8351 PF70: [AR]_[AR].OR.WORK[PUR] ;PUT IN NEW AGE AND
; 8352 ; USE BITS
; 8353 .IFNOT/INHCST
; 8354 =O** START NO TEST WRITE, ;START MEMORY WRITE
; 8355 CALL [IBPX] ;GO STORE IN MEMORY
; 8356 .ENDIF/INHCST
; 8357 .IF/INHCST
; 8358 =O** SKIP NO CST, ;SEE IF A CST
; 8359 CALL [WRCST] ;AND GO WRITE IN MEMORY
; 8360 .ENDIF/INHCST
; 8361 SC_7 ;THIS CAN BE BUMMED
; 8362 =0
; 8363 PF75: [AR]_[AR]*2, ;CONVERT PAGE NUMBER TO
; 8364 STEP SC, ;PAGE ADDRESS
; 8365 J/PF75 ;LOOP OVER 9 STEPS
; 8366
```

; T1OKL.MCR[10,1141] 15:34 27-JULY-1984
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 232
PAGE FAIL REFIL LOGIC

```
U 2573, 1040,3333,0006,4174,4007,0520,0000,0000,0000
; 8367 ;STILL .IF/KLPAGE
; 8368 ;WE NOW HAVE THE ADDRESS OF THE PAGE TABLE ENTRY. GO
; 8369 ; READ IT AND START ANALYSIS
; 8370
; 8371 ;IF WE ARE HERE FOR THE FIRST TIME FOR THE USER OR EXEC SAVE THE
; 8372 ; ADDRESS OF THE PAGE TABLE IN PTA.E OR PTA.U SO THAT WE DO NOT
; 8373 ; HAVE TO DO THE SECTION LOOKUP EVERY TIME.
; 8374 READ [BRX], SKIP DPO ;USER OR EXEC REF?
; 8375 =000 [AR]_WORK[PTA.E], ;EXEC MODE
; 8376 SKIP AD.EQ.O, 4T, ;SEE IF SET YET
; 8377 CALL [SHDREM] ;SHOULD WE REMEMBER PTR
; 8378 [AR]_WORK[PTA.U], ;USER MODE
; 8379 SKIP AD.EQ.O, 4T, ;SEE IF SET YET
; 8380 CALL [SHDREM] ;SHOULD WE REMEMBER PTR
; 8381 WORK[PTA.E]_[ARX], ;SAVE FOR EXEC
; 8382 J/PF76 ;CONTINUE BELOW
; 8383 WORK[PTA.U]_[ARX], ;SAVE FOR USER
; 8384 J/PF76 ;CONTINUE BELOW
; 8385 =111
; 8386 PF76: VMA [ARX]+[BR], ;READ PAGE POINTER
; 8387 START READ,
; 8388 VMA PHYSICAL
; 8389 =
; 8390 =00
; 8391 PF77: MEM READ, ;START ANALYSIS OF POINTER
; 8392 [AR]_MEM,
; 8393 CALL [SETPTR]
; 8394 PTRIMM: TL [AR], ;IMMEDIATE POINTER
; 8395 #/77, ;CHECK FOR BITS 0-5
; 8396 J/PF80 ;GO TO PF80
; 8397 [AR]_[AR]+WORK[SBR], ;SHARED POINTER
; 8398 J/PTRSHR ;GO TO READ SPT
; 8399
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 233
PAGE FAIL REFIL LOGIC

U 1137, 2576,3770,0305,4344,4007,0670,0000,0000,0000
U 2576, 2610,3333,0003,7174,4007,0700,0400,0000,0426
U 2577, 2600,4251,0505,4374,4007,0370,0000,0000,0777
U 2600, 1000,4551,0303,4374,0007,0700,0000,0027,7000
U 2601, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 2571, 1511,4443,0000,4174,4007,0700,0010,0000,0000
U 2575, 1140,4553,0300,4374,4007,0321,0000,0000,0077
U 1140, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 1141, 1502,4251,0304,4374,4007,0700,0010,0000,3777
U 1143, 3761,3551,0606,4374,0007,0700,0000,0010,0000
U 3761, 2602,4553,1300,4374,4007,0321,0000,0002,0000
U 2602, 3762,3551,0606,4374,4007,0700,0000,0002,0000
U 2603, 2604,4553,0600,4374,4007,0321,0000,0003,0000
U 2604, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 2605, 2613,3551,0303,7274,4007,0701,0000,0000,0220
U 2610, 3574,3771,0003,7274,4117,0701,0010,0000,0301
U 2612, 1137,3771,0003,7274,4007,0701,0000,0000,0426

```
; 8400 ;STILL .IF/KLPAGE
; 8401 ;INDIRECT POINTER. CHANGE PAGE # AND LOOK FOR PAGE TABLE
; 8402 PTRIND: [BR]_[AR] SWAP, ;PUT IN RIGHT HALF
; 8403 SKIP7-1 MS ;DID CLOCK GO OFF
; 8404 =0 WORK[SV.AR1]_[AR], ;YES--UPDATE CLOCK
; 8405 J/PFTICK ;...
; 8406 [BR]_[BR].AND.# CLR LH, ;UPDATE PAGE # AND RESTART
; 8407 #/777, ;MASK FOR PAGE #
; 8408 SKIP IRPT ;SEE IF THIS IS A LOOP
; 8409 =0 [AR]_[AR].AND.#, ;CHANGE INDIRECT POINTER
; 8410 #/277000, ; INTO SHARE POINTER
; 8411 HOLD RIGHT, ;...
; 8412 J/PF45 ;GO BACK AND TRY AGAIN
; 8413 PAGE FAIL TRAP ;POINTER LOOP
; 8414
; 8415 =0**
; 8416 PTRSHR: CALL [RDPT] ;GO LOOK AT POINTER
; 8417 TL [AR], ;BITS 12-17 .EQ. 0?
; 8418 #/77
; 8419
; 8420 ;HERE WITH FINAL POINTER. SKIP IF 12-17 NOT EQUAL TO ZERO
; 8421 =00
; 8422 PF80: PAGE FAIL TRAP ;NO--TAKE A TRAP
; 8423 .IFNOT/NOCST
; 8424 [ARX]_[AR].AND.# CLR LH, ;SAVE PHYSICAL PAGE # (=11)
; 8425 #/3777, ;MASK TO 13 BITS ;IF/NOCST
; 8426 CALL [UPCST] ;UPDATE CSTO ;=0
; 8427 .ENDIF/NOCST ;PF80: PAGE FAIL TRAP
; 8428 ;HERE WE HAVE CST ENTRY IN AR, PAGE FAIL WORD IN BRX. GO LOOK
; 8429 ; AT WRITABLE AND WRITTEN BITS
; 8430 =11
; 8431 PF90: [BRX]_[BRX].OR.#, ;TRANSLATION IS VALID
; 8432 #/100000, HOLD RIGHT ;...
; 8433 TL [FLG], FLG.W/1 ;IS THIS PAGE WRITABLE?
; 8434 =0 [BRX]_[BRX].OR.#, ;YES--INDICATE THAT IN PFW
; 8435 #/020000,
; 8436 J/PF100 ;NOT WRITE VIOLATION
; 8437 TL [BRX], ;IS THIS A WRITE REF?
; 8438 WRITE TEST/1, WRITE CYCLE/1
; 8439 =0 PAGE FAIL TRAP ;WRITE VIOLATION
; 8440 PF107:
; 8441 .IFNOT/NOCST
; 8442 [AR]_[AR].OR.WORK[PUR], ;PUT IN NEW AGE
; 8443 J/PF110 ;GO TO STORE CST ENTRY
; 8444 .ENDIF/NOCST
; 8445 .IF/NOCST
; 8446 PFDONE: TR [FLG],
; 8447 #/400000,
; 8448 J/PF140
; 8449 .ENDIF/NOCST
; 8450
; 8451 =0*
; 8452 PFTICK: [AR]_WORK[TIME1], ;UPDATE TIMER
; 8453 SPEC/CLRCLK, CALL [TOCK]
; 8454 [AR]_WORK[SV.AR1], ;RESTORE AR
; 8455 J/PTRIND ;GO TRY AGAIN
```

(=11)
IF/NOCST
=0
PF80: PAGE FAIL TRAP

Produced on Advanced Information Services Electronic Laser Printer. PKO1165. DTN: 223-7881

U 1137, 2576,3770,0305,4344,4007,0670,0000,0000,0000
U 2576, 2610,3333,0003,7174,4007,0700,0400,0000,0426
U 2577, 2600,4251,0505,4374,4007,0370,0000,0000,0777
U 2600, 1000,4551,0303,4374,0007,0700,0000,0027,7000
U 2601, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 2571, 1511,4443,0000,4174,4007,0700,0010,0000,0000
U 2575, 1140,4553,0300,4374,4007,0321,0000,0000,0077
U 1140, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 1141, 1502,4251,0304,4374,4007,0700,0010,0000,3777
U 1143, 3761,3551,0606,4374,0007,0700,0000,0010,0000
U 3761, 2602,4553,1300,4374,4007,0321,0000,0002,0000
U 2602, 3762,3551,0606,4374,4007,0700,0000,0002,0000
U 2603, 2604,4553,0600,4374,4007,0321,0000,0003,0000
U 2604, 2670,4553,1300,4374,4007,0321,0000,0001,0000
U 2605, 2613,3551,0303,7274,4007,0701,0000,0000,0220
U 2610, 3574,3771,0003,7274,4117,0701,0010,0000,0301
U 2612, 1137,3771,0003,7274,4007,0701,0000,0000,0426

```
; 8400 ;STILL .IF/KLPAGE
; 8401 ;INDIRECT POINTER. CHANGE PAGE # AND LOOK FOR PAGE TABLE
; 8402 PTRIND: [BR] [AR] SWAP, ;PUT IN RIGHT HALF
; 8403 SKIP7-1 MS ;DID CLOCK GO OFF
; 8404 =0 WORK[SV.AR1]_[AR], ;YES--UPDATE CLOCK
; 8405 J/PFTICK ;...
; 8406 [BR] [BR].AND.# CLR LH, ;UPDATE PAGE # AND RESTART
; 8407 #/777, ;MASK FOR PAGE #
; 8408 SKIP IRPT ;SEE IF THIS IS A LOOP
; 8409 =0 [AR]_[AR].AND.#, ;CHANGE INDIRECT POINTER
; 8410 #/277000, ; INTO SHARE POINTER
; 8411 HOLD RIGHT, ;...
; 8412 J/PF45 ;GO BACK AND TRY AGAIN
; 8413 PAGE FAIL TRAP ;POINTER LOOP
; 8414
; 8415 =0**
; 8416 PTRSHR: CALL [RDPT] ;GO LOOK AT POINTER
; 8417 TL [AR], ;BITS 12-17 .EQ. 0?
; 8418 #/77
; 8419
; 8420 ;HERE WITH FINAL POINTER. SKIP IF 12-17 NOT EQUAL TO ZERO
; 8421 =00
; 8422 PF80: PAGE FAIL TRAP ;NO--TAKE A TRAP
; 8423 .IFNOT/NOCS
; 8424 [ARX]_[AR].AND.# CLR LH, ;SAVE PHYSICAL PAGE # (=11)
; 8425 #/3777, ;MASK TO 13 BITS ;IF/NOCS
; 8426 CALL [UPCST] ;UPDATE CSTO ;PF80: PAGE FAIL TRAP
; 8427 .ENDIF/NOCS
; 8428 ;HERE WE HAVE CST ENTRY IN AR, PAGE FAIL WORD IN BRX. GO LOOK
; 8429 ; AT WRITABLE AND WRITTEN BITS
; 8430 =11
; 8431 PF90: [BRX]_[BRX].OR.#, ;TRANSLATION
; 8432 #/100000, HOLD RIGHT ;...
; 8433 TL [FLG], FLG.W/1 ;IS THIS PAGE
; 8434 =0 [BRX]_[BRX].OR.#, ;YES--INDICAT
; 8435 #/020000,
; 8436 J/PF100 ;NOT WRITE VI
; 8437 TL [BRX], ;IS THIS A WR
; 8438 WRITE TEST/1, WRITE CYCLE/1
; 8439 =0 PAGE FAIL TRAP ;WRITE VIOLAT
; 8440 PF107:
; 8441 .IFNOT/NOCS
; 8442 [AR]_[AR].OR.WORK[PUR], ;PUT IN NEW A
; 8443 J/PF110 ;GO TO STORE
; 8444 .ENDIF/NOCS
; 8445 .IF/NOCS
; 8446 PFDONE: TR [FLG],
; 8447 #/400000,
; 8448 J/PF140
; 8449 .ENDIF/NOCS
; 8450
; 8451 =0*
; 8452 PFTICK: [AR] WORK[TIME1], ;UPDATE TIMER
; 8453 SPEC/CLRCLK, CALL [TOCK]
; 8454 [AR] WORK[SV.AR1], ;RESTORE AR
; 8455 J/PTRIND ;GO TRY AGAIN
```

WRONG
IF NOCS
IFNOT is w constant
Block
Block should be =0
(NOT =00) if NOCS
PF80: PFT
=0
PF80: PFT
=11

Produced on Advanced Information Services Electronic Laser Printer, PKOJES6, DTN: 223-7881

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 233-1
PAGE FAIL REFIL LOGIC

; 8456

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 234
PAGE FAIL REFIL LOGIC

U 3762, 2606,4553,0600,4374,4007,0321,0000,0001,0000

U 2606, 2614,3551,0303,4370,4007,0700,0000,0000,0001

U 2607, 2614,4553,0300,4374,4007,0331,0000,0000,0001

U 2614, 2605,3551,0606,4374,0007,0700,0000,0004,0000

U 2615, 2605,5551,1313,4374,0007,0700,0000,0002,0000

U 2613, 2634,3773,0000,7274,4007,0622,0010,0000,0216

```
; 8457 ;STILL .IF/KLPAGE
; 8458 ;HERE IF PAGE IS WRITABLE
; 8459 PF100: TL [BRX], WRITE CYCLE/1 ;IS THIS A WRITE REF?
; 8460 =0 [AR]_[AR].OR.#, ;YES--SET WRITTEN BIT
; 8461 #/1,
; 8462 HOLD LEFT,
; 8463 J/PF105
; 8464 TR [AR], ;NOT WRITE, ALREADY WRITTEN?
; 8465 #/1
; 8466 =0
; 8467 PF105: [BRX]_[BRX].OR.#, ;WRITTEN SET BIT
; 8468 #/040000, ;MARK PAGE AS
; 8469 HOLD RIGHT, ;WRITABLE
; 8470 J/PF107 ;STORE CST WORD
; 8471 [FLG]_[FLG].AND.NOT.#, ;NOT WRITTEN, CAUSE TRAP ON
; 8472 FLG.W/1, ; WRITE ATTEMPT
; 8473 HOLD RIGHT, ;ONLY CLEAR LH
; 8474 J/PF107
; 8475 .IFNOT/NOCST
; 8476 =0**
; 8477 PF110:
; 8478 .IFNOT/INHCST
; 8479 START NO TEST WRITE,
; 8480 CALL [IBPX] ;STORE CST ENTRY
; 8481 .ENDIF/INHCST
; 8482 .IF/INHCST
; 8483 SKIP NO CST,
; 8484 CALL [WRCST]
; 8485 .ENDIF/INHCST
; 8486
; 8487 .ENDIF/KLPAGE
; 8488
```


; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 235
PAGE FAIL REFIL LOGIC

```
U 2617, 2620,4553,1300,4374,4007,0331,0000,0040,0000
U 2620, 2622,3441,0403,4174,4007,0700,2000,0071,0007
U 2621, 3767,3771,0003,7274,4007,0701,0000,0000,0210
U 2622, 2622,3445,0303,4174,4007,0630,2000,0060,0000
U 2623, 3763,4551,0303,4374,0007,0700,0000,0000,0003
U 3763, 3764,4221,0013,4170,4007,0700,0000,0000,0000
U 3764, 3765,3551,0606,4374,0007,0700,0000,0010,0000
U 3765, 2624,4553,1300,4374,4007,0321,0000,0000,4000
U 2624, 2625,3551,0606,4374,0007,0700,0000,0000,2000
U 2625, 3766,4551,0606,4370,4007,0700,0000,0000,0777
U 3766, 1500,3111,0603,4174,4003,7700,0200,0003,0001

; 8489
; 8490 ;HERE WHEN WE HAVE FIGURED OUT PHYSICAL ADDRESS (IN ARX) AND FLAGS
; 8491 ; (IN BRX) RELOAD PAGE TABLE.
; 8492 PFDONE: TR [FLG], ;MAP INSTRUCTION?
; 8493 #/400000
; 8494 .ENDIF/NOCST
; 8495 =0
; 8496 PF140: [AR]_[ARX], ;GET PHYSICAL PAGE #
; 8497 SC 7, ;PREPARE TO CONVERT TO
; 8498 J/PF130 ;WORD ADDRESS
; 8499 [AR]_WORK[SV.VMA], ;RESTORE VMA
; 8500 J/PF120
; 8501 =0
; 8502 PF130: [AR]_[AR]*2, ;CONVERT TO WORD #
; 8503 STEP SC,
; 8504 J/PF130
; 8505 [AR]_[AR].AND.#, ;JUST ADDRESS BITS
; 8506 #/3,
; 8507 HOLD RIGHT
; 8508 END MAP ;CLEAR MAP FLAGS
; 8509 [BRX]_[BRX].OR.#, ;TURN ON THE TRANSLATION
; 8510 #/100000, ;VALID BIT
; 8511 HOLD RIGHT ;IN LEFT HALF ONLY
; 8512 TL [FLG], FLG.C/1 ;CACHE BIT SET?
; 8513 [BRX]_[BRX].OR.#, ;YES--SET IN MAP WORD
; 8514 #/002000, HOLD RIGHT ;..
; 8515 [BRX]_[BRX].AND.#, ;PRESERVE WORD #
; 8516 #/777, HOLD LEFT ;IN PAGE FAIL WORD
; 8517 [AR]_[AR].OR.[BRX], ;COMPLETE MAP INSTRUCTION
; 8518 EXIT
; 8519
```

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 236
PAGE FAIL REFIL LOGIC

U 3767, 3770,3441,0305,4174,4007,0700,0000,0000,0000 ; 8520
; 8521
; 8522
; 8523
; 8524
U 3770, 3771,5551,0305,4374,0007,0700,0000,0007,0000 ; 8525
; 8526
U 3771, 3772,3443,0500,4174,4007,0701,0200,0000,0030 ; 8527
; 8528
U 3772, 3773,4251,0404,4374,4007,0700,0000,0000,3777 ; 8529
; 8530
U 3773, 3774,3551,0406,4374,4007,0700,0000,0040,0000 ; 8531
U 3774, 2626,4553,1300,4374,4007,0321,0000,0002,0000 ; 8532
U 2626, 2627,3551,0606,4374,4007,0700,0000,0004,0000 ; 8533
; 8534
U 2627, 2630,4553,1300,4374,4147,0321,0000,0000,4000 ; 8535
; 8536
U 2630, 3775,3551,0606,4374,4007,0700,0000,0002,0000 ; 8537
U 2631, 3775,3333,0006,4174,4007,0700,0000,0000,0000 ; 8538
U 3775, 3776,3771,0004,7274,4007,0701,0000,0000,0212 ; 8539
U 3776, 1435,3771,0005,7274,4007,0701,0000,0000,0213 ; 8540
U 1435, 1500,3771,0006,7274,4007,0701,0000,0000,0214 ; 8541
; 8542
U 1500, 1501,3443,0300,4174,4007,0701,0200,0000,0032 ; 8543
; 8544
; 8545
; 8546
U 1501, 0000,3771,0003,7274,4004,1701,0000,0000,0211 ; 8547
; 8548
; 8549

PF120: [BR]_[AR] ;COPY PAGE FAIL WORD
[BR]_[AR].AND.NOT.#, ;CLEAR BITS WHICH START A CYCLE
READ CYCLE/1, ; ..
WRITE CYCLE/1, ; ..
WRITE TEST/1, ; ..
HOLD RIGHT ;JUST DO LEFT HALF
VMA [BR], 3T, ;RESTORE VMA
DP FUNC/1 ;SET USER ACCORDING TO WHAT IT WAS
[ARX]_[ARX].AND.# CLR LH, ;JUST KEEP PAGE #
#/3777 ; ..
[BRX]_[ARX].OR.#, #/40000 ;SET VALID BITS
TL [FLG], FLG.W/1 ;WANT WRITE SET?
[BRX]_[BRX].OR.#, #/040000 ;SET WRITE BIT
TL [FLG], FLG.C/1, ;WANT CACHE SET?
LOAD PAGE TABLE ;LOAD PAGE TABLE ON NEXT
; MICRO INSTRUCTION
=0 [BRX]_[BRX].OR.#, ;SET CACHE BIT
#/020000, J/PF125 ;CACHE BIT
READ [BRX] ;LOAD PAGE TABLE
PF125: [ARX]_WORK[SV.ARX]
[BR]_WORK[SV.BR]
[BRX]_WORK[SV.BRX]
VMA [AR], ;MAKE MEM REQUEST
DP FUNC/1, 3T, ;FROM DATA PATH
WAIT/1 ;WAIT FOR PREVIOUS CYCLE TO
; COMPLETE. (NEED THIS TO
; START ANOTHER CYCLE)
[AR]_WORK[SV.AR];
RETURN [O]

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 237
PAGE FAIL REFIL LOGIC

```
; 8550 .IF/KLPAGE
; 8551 .IFNOT/NO CST
; 8552 ;SUBROUTINE TO START CST UPDATE
; 8553 ;CALL WITH:
; 8554 ; AR/ PHYSICAL PAGE NUMBER
; 8555 ;RETURN 2 WITH ENTRY IN AR, PAGE FAIL IF AGE TOO SMALL
; 8556 .IFNOT/INHCST
; 8557 =0**
; 8558 UPCST: [AR]_[AR]+WORK[CBR], ;ADDRESS OF CSTO ENTRY
; 8559 CALL [RDPT] ;READ OLD VALUE
; 8560 TL [AR], ;O - 5 = 0?
; 8561 #/770000 ; ..
; 8562 =0 [AR]_[AR].AND.WORK[CSTM], ;CLEAR AGE FIELD
; 8563 RETURN [2] ;AGE IS NOT ZERO
; 8564 PAGE FAIL TRAP ;AGE TOO SMALL
; 8565 .ENDIF/INHCST
; 8566 .IF/INHCST
; 8567 UPCST: SKIP NO CST ;SEE IF A CST IS PRESENT
; 8568 =0*[AR]_[AR]+WORK[CBR], ;YES, ADDRESS OF CSTO ENTRY
; 8569 CALL [RDPT] ;READ OLD VALUE
; 8570 [AR]_0.RETURN [2] ;NO CST, RETURN
; 8571 TL [AR], ;CHECK AGE FIELD
; 8572 #/770000
; 8573 =
; 8574 =0 [AR]_[AR].AND.WORK[CSTM], ;CLEAR AGE FIELD
; 8575 RETURN [2] ;AGE IS NOT ZERO
; 8576 PAGE FAIL TRAP ;AGE TOO SMALL
; 8577
; 8578 =0
; 8579 WRCST: START NO TEST WRITE,
; 8580 J/IBPX
; 8581 RETURN [4]
; 8582 .ENDIF/INHCST
; 8583 .ENDIF/NO CST
; 8584
```

U 1502, 1150,3773,0000,7274,4007,0622,0000,0000,0216

U 1150, 1511,0551,0303,7274,4007,0701,0010,0000,0216

U 1151, 0002,4221,0003,4174,4004,1700,0000,0000,0000

U 1154, 2632,4553,0300,4374,4007,0321,0000,0077,0000

U 2632, 0002,4551,0303,7274,4004,1701,0000,0000,0217

U 2633, 2670,4553,1300,4374,4007,0321,0000,0001,0000

U 2634, 3070,4443,0000,4174,4007,0700,0200,0001,0002

U 2635, 0004,4443,0000,4174,4004,1700,0000,0000,0000

; T1OKL.MICR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 238
PAGE FAIL REFIL LOGIC

U 1503, 1507, 3551, 0304, 4374, 4007, 0700, 0000, 0075, 3777
U 1507, 1510, 4111, 0413, 4174, 0007, 0700, 0000, 0000, 0000
U 1510, 2636, 3333, 0003, 4174, 4007, 0520, 0000, 0000, 0000
U 2636, 2640, 4553, 0300, 4374, 4007, 0321, 0000, 0030, 0000
U 2637, 2670, 4553, 1300, 4374, 4007, 0321, 0000, 0001, 0000
U 2640, 2642, 4553, 0300, 4374, 4007, 0321, 0000, 0010, 0000
U 2641, 2670, 4553, 1300, 4374, 4007, 0321, 0000, 0001, 0000
U 2642, 2644, 4553, 0300, 4374, 4007, 0321, 0000, 0020, 0000
U 2643, 0002, 4443, 0000, 4174, 4004, 1700, 0000, 0000, 0000
U 2644, 0003, 4443, 0000, 4174, 4004, 1700, 0000, 0000, 0000
U 2645, 0001, 4443, 0000, 4174, 4004, 1700, 0000, 0000, 0000

```
; 8585 ;STILL .IF/KLPAGE
; 8586 ;SUBROUTINE TO LOOK AT PAGE POINTER
; 8587 ;CALL WITH POINTER IN AR
; 8588 ;RETURNS 1 IF TYPE 1
; 8589 ;RETURNS 2 IF TYPE 2
; 8590 ;RETURNS 3 IF TYPE 3
; 8591 ;GOES TO PFT IF TYPE 0 OR 4 THRU 7
; 8592 SETPTR: [AR]_[AR].OR.#, ;AND C AND W BITS
; 8593 #/753777 ; OF ALL POINTERS
; 8594 [FLG]_[FLG].AND.[AR], ;
; 8595 HOLD RIGHT ;KEEP IN LH OF FLG
; 8596 READ [AR], ;TYPE 4,5,6 OR 7?
; 8597 SKIP DPO ;
; 8598 =0 TL [AR], ;HERE WE TEST FOR TYPE
; 8599 #/300000, ; ZERO POINTER
; 8600 J/STPTR1 ;CHECK AT STPTR1
; 8601 PAGE FAIL TRAP ;BAD TYPE
; 8602 =0
; 8603 STPTR1: TL [AR], ;NOT ZERO
; 8604 #/100000, ;SEPERATE TYPE 2
; 8605 J/STPTR2 ;
; 8606 PAGE FAIL TRAP ;TYPE 0
; 8607
; 8608 =0
; 8609 STPTR2: TL [AR], ;SEPERATE TYPE 1
; 8610 #/200000, ; AND 3
; 8611 J/STPTR3 ;
; 8612 RETURN [2] ;TYPE 2
; 8613
; 8614 =0
; 8615 STPTR3: RETURN [3] ;TYPE 3
; 8616 RETURN [1] ;TYPE 1
; 8617
```

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 239
PAGE FAIL REFIL LOGIC

U 1511, 2646,3443,0300,4174,4007,0370,0200,0024,1016

U 2646, 0004,3771,0003,4365,5004,1700,0200,0000,0002
U 2647, 2670,4553,1300,4374,4007,0321,0000,0001,0000

U 2650, 0007,4443,0000,4174,4004,1700,0000,0000,0000
U 2651, 1512,7441,1303,4174,4007,0700,0000,0000,0000
U 1512, 2652,4553,0300,4374,4007,0321,0000,0002,4000
U 2652, 0007,4443,0000,4174,4004,1700,0000,0000,0000
U 2653, 0002,4443,0000,4174,4004,1700,0000,0000,0000

```
; 8618 ;STILL .IF/KLPAGE
; 8619 ;SUBROUTINE TO FETCH A PAGE POINTER OR CST ENTRY
; 8620 ;CALL WITH ADDRESS IN AR
; 8621 ;RETURN 4 WITH WORD IN AR
; 8622 ;
; 8623 RDPT: VMA_[AR], ;LOAD THE VMA
; 8624 START READ, ;START MEM CYCLE
; 8625 VMA PHYSICAL, ;ABSOLUTE ADDRESS
; 8626 SKIP IRPT ;CHECK FOR INTERRUPTS
; 8627 =0 MEM READ, ;NO INTERRUPTS
; 8628 [AR]_MEM, ;PUT THE DATA INTO AR
; 8629 RETURN [4] ;AND RETURN
; 8630 PAGE FAIL TRAP ;INTERRUPT
; 8631
; 8632
; 8633 ;SUBROUTINE TO SEE IF WE SHOULD REMEMBER AN EXEC SECTION PTR
; 8634 ;CALL WITH SKIP ON ADR.EQ.0
; 8635 ;RETURNS 2 IF WE SHOULD STORE AND 7 IF WE SHOULD NOT
; 8636 ;
; 8637 =0
; 8638 SHDREM: RETURN [7] ;INDIRECT PTR
; 8639 [AR]_NOT.[FLG] ;FLIP BITS
; 8640 TL [AR], FLG.W/1, FLG.C/1 ;BOTH BITS SET
; 8641 =0 RETURN [7] ;NO--DON'T STORE
; 8642 RETURN [2] ;STORE
; 8643
; 8644 .ENDIF/KLPAGE
; 8645
```

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 240
PAGE FAIL REFIL LOGIC

```
;;8646 .IF/KIPAGE
;;8647 ;HERE IN KI10 MODE
;;8648 ;BR CONTAINS PAGE # AND ARX CONTAINS PAGE #/2
;;8649
;;8650 KIFILL: READ [BRX], ;USER REF?
;;8651 SKIP DPO ;..
;;8652 =0 [BR]-#, ;EXEC--LESS THAN 340?
;;8653 #/340, ;..
;;8654 SKIP DP18, 4T, ;..
;;8655 J/KIF10 ;FOLLOW EXEC PATH
;;8656 KIUPT: [ARX]_[ARX]+[UBR], ;POINTER TO PAGE MAP ENTRY
;;8657 LOAD VMA, ;PUT ADDRESS IN VMA
;;8658 VMA PHYSICAL, ;ABSOLUTE ADDRESS
;;8659 START READ, ;FETCH UPT WORD
;;8660 J/KIF30 ;JOIN COMMON CODE
;;8661 =0
;;8662 KIF10: [BR]-#, ;EXEC ADDRESS .GE. 340
;;8663 #/400, ;SEE IF .GT. 400
;;8664 SKIP DP18, 4T, ;..
;;8665 J/KIEPT ;LOOK AT KIF20
;;8666 [ARX]_[ARX]+#, 3T, ;EXEC ADDRESS .LT. 340
;;8667 #/600, ;IN EBR+600
;;8668 J/KIEPT ;JOIN COMMON CODE
;;8669
;;8670 =0
;;8671 KIEPT: [ARX]_[ARX]+[EBR], ;ADD OFFSET TO
;;8672 LOAD VMA, ;EPT
;;8673 START READ, ;START FETCH
;;8674 VMA PHYSICAL, ;ABSOLUTE ADDRESS
;;8675 J/KIF30 ;GO GET POINTER
;;8676 [ARX]_[ARX]+#, ;PER PROCESS PAGE
;;8677 #/220, 3T, ;IS IN UPT + 400
;;8678 J/KIUPT ;JOIN COMMON CODE
;;8679 KIF30: MEM READ, ;WAIT FOR DATA
;;8680 [ARX]_MEM ;PLACE IT IN ARX
;;8681 TR [BR], ;SEE IF EVEN OR ODD
;;8682 #/1, ;..
;;8683
```

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 241
PAGE FAIL REFIL LOGIC

```
;;8684 ;STILL .IF/KIPAGE
;;8685 =0
;;8686 KIF40: READ [ARX], ;ODD
;;8687 SKIP DP18, ;SEE IF VALID
;;8688 J/KIF50 ;JOIN COMMON CODE
;;8689 [ARX]_[ARX] SWAP, ;EVEN--FLIP AROUND
;;8690 J/KIF40 ; AND CONTINUE
;;8691
;;8692 .ENDIF/KIPAGE
;;8693 =0
;;8694 KIF50: PAGE FAIL TRAP
;;8695 ;AT THIS POINT WE HAVE THE PAGE MAP ENTRY IN RH OF AR
;;8696 [FLG]_[FLG].AND.NOT.#, ;CLEAR W AND C
;;8697 FLG.W/1, FLG.C/1 ; FLAGS
;;8698 TR [ARX], #/020000 ;CACHE ENABLED?
;;8699 =0 [FLG]_[FLG].OR.#, ;SET CACHE BITS
;;8700 FLG.C/1, HOLD RIGHT ; ..
;;8701 TR [ARX], #/040000 ;DO NOT CACHE
;;8702 ;SEE IF CACHE BIT SET
;;8703 =0 [BRX]_[BRX].OR.#, ;COPY BITS TO BRX
;;8704 #/020000,
;;8705 HOLD RIGHT
;;8706 TR [ARX], ; ..
;;8707 #/100000
;;8708 =0 [FLG]_[FLG].OR.#, ;SAVE W
;;8709 FLG.W/1, ; ..
;;8710 HOLD RIGHT, ; ..
;;8711 J/KIF90 ;ALL DONE
;;8712 TL [BRX], ;W=0, WRITE REF?
;;8713 WRITE CYCLE/1
;;8714 =0
;;8715 KIF80: [BRX]_[BRX].OR.#, ;WRITE FAILURE
;;8716 #/100000, HOLD RIGHT, ;INDICATE THAT ACCESS WAS ON
;;8717 J/KIF50 ;GO PAGE FAIL
;;8718 J/PFDONE ;ALL DONE
;;8719
;;8720 KIF90: [BRX]_[BRX].OR.#, ;PAGE IS WRITABLE
;;8721 #/40000, ;TURN ON IN BRX
;;8722 J/PFDONE ;ALL SET
;;8723
```

U 2654, 2670,4553,1300,4374,4007,0321,0000,0001,0000

U 2655, 1513,5551,1313,4374,4007,0700,0000,0002,4000

U 1513, 2656,4553,0400,4374,4007,0331,0000,0002,0000

U 2656, 2657,3551,1313,4374,0007,0700,0000,0000,4000

U 2657, 2660,4553,0400,4374,4007,0331,0000,0004,0000

U 2660, 2661,3551,0606,4374,0007,0700,0000,0002,0000

U 2661, 2662,4553,0400,4374,4007,0331,0000,0010,0000

U 2662, 1562,3551,1313,4374,0007,0700,0000,0002,0000

U 2663, 2664,4553,0600,4374,4007,0321,0000,0001,0000

U 2664, 2654,3551,0606,4374,0007,0700,0000,0010,0000

U 2665, 2617,4443,0000,4174,4007,0700,0000,0000,0000

U 1562, 2617,3551,0606,4374,4007,0700,0000,0004,0000

; T1OKL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 242
PAGE FAIL REFIL LOGIC

```
U 1563, 2666, 3333, 0005, 7174, 4007, 0700, 0400, 0000, 0213 ; 8724 ;HERE ON HARD PAGE FAILURES
; 8725 HARD: WORK[SV.BR]_[BR] ;SAVE BR (CLEANUP MAY NEED IT)
; 8726 =0 [BR]_VMA, ;BUILD PAGE FAIL WORD
U 2666, 1575, 3771, 0005, 4354, 4007, 0700, 0010, 0000, 0000 ; 8727 CALL_[ABORT] ;CLEAR ERROR
; 8728 [BR]_[BR].AND.#, ;SAVE THE FLAGS
; 8729 #/40T237, ;...
U 2667, 1564, 4551, 0505, 4374, 0007, 0700, 0000, 0040, 1237 ; 8730 HOLD RIGHT ;...
; 8731 [BRX]_[BRX].OR.[BR], ;COMPLETE PAGE FAIL WORD
U 1564, 2654, 3111, 0506, 4174, 4007, 0700, 0000, 0000, 0000 ; 8732 J/KIF50 ;GO TRAP
; 8733
U 1572, 1144, 4443, 0000, 4174, 4007, 0370, 0000, 0000, 0000 ; 8734 PFPI1: SKIP IRPT ;TIMER TRAP?
; 8735 =0
; 8736 [AR]_WORK[TIME1], ;YES--GET LOW WORD
; 8737 SPEC/CLRCLK, ;CLEAR CLOCK FLAG
U 1144, 3574, 3771, 0003, 7274, 4117, 0701, 0010, 0000, 0301 ; 8738 CALL [TOCK] ;DO THE UPDATE
U 1145, 2671, 4443, 0000, 4174, 4007, 0700, 0000, 0000, 0000 ; 8739 J/PFT1 ;EXTERNAL INTERRUPT
U 1146, 1573, 4223, 0000, 4364, 4277, 0700, 0200, 0000, 0010 ; 8740 ABORT MEM CYCLE ;CLEAR 1MS FLAGS
; 8741 =
; 8742 PFPI2: [AR]_WORK[SV.VMA], ;RESTORE VMA
; 8743 J/PF125
; 8744
; 8745
U 1573, 3775, 3771, 0003, 7274, 4007, 0701, 0000, 0000, 0210 ; 8746 ABORT: ABORT MEM CYCLE, RETURN [1]
; 8747
U 1575, 0001, 4223, 0000, 4364, 4274, 1700, 0200, 0000, 0010
```


; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 243
PAGE FAIL REFIL LOGIC

```
U 2670, 0104,4751,1217,4374,4007,0700,0000,0000,0100
U 2671, 2672,3771,0003,7274,4007,0611,0000,0000,0210
U 2672, 2674,4553,0300,4374,4007,0321,0000,0010,0000
U 2673, 1602,3771,0003,7274,4007,0701,0000,0000,0425
U 1602, 2676,3333,0003,4174,4467,0700,0000,0000,0004
U 2674, 1100,4443,0000,4174,4007,0700,0000,0000,0000
U 2675, 2676,1111,0701,4170,4007,0700,4000,0000,0000
U 2676, 1100,3333,0013,4174,4003,5701,0000,0000,0000

U 1100, 2700,4221,0013,4170,4007,0370,0000,0000,0000
U 1101, 3162,3771,0003,7274,4007,0701,0000,0000,0212
U 1102, 1604,0111,0701,4174,4007,0700,0000,0000,0000
U 1103, 3477,3771,0013,4370,4007,0700,0000,0000,0011
U 1104, 3477,3771,0013,4370,4007,0700,0000,0000,0012
U 1105, 2326,3771,0013,4370,4007,0700,0000,0000,0003
U 1106, 3506,3771,0013,4370,4007,0700,0000,0000,0012
U 1107, 3503,3771,0013,4370,4007,0700,0000,0000,0011
U 1110, 3475,3771,0013,4370,4007,0700,0000,0000,0011
U 1111, 2324,4221,0013,4170,4007,0700,0000,0000,0000
U 1112, 2326,4221,0013,4170,4007,0700,0000,0000,0000
U 1113, 2326,3771,0013,4370,4007,0700,0000,0000,0011

; 8748 ;HERE ON PAGE FAIL TRAP
; 8749 =0
; 8750 PFT: HALT [IOPF] ;IO PAGE FAILURE
; 8751 PFT1: [AR]_WORK[SV.VMA],
; 8752 SKIP/TRAP CYCLE ;SEE IF TRAP CYCLE
; 8753 =0 TL [AR], FETCH/1, ;IS THIS AN INSTRUCTION FETCH
; 8754 J/PFT1A ;GO LOOK BELOW
; 8755 [AR]_WORK[TRAPPC] ;RESTORE PC
; 8756 READ [AR], LOAD FLAGS, J/CLDISP
; 8757 =0
; 8758 PFT1A: J/CLEANED ;YES--NO PC TO BACK UP
; 8759 FIXPC: [PC]_[PC]-1, HOLD LEFT ;DATA FAILURE--BACKUP PC
; 8760 =0
; 8761 CLDISP: CLEANUP DISP ;GO CLEANUP AFTER PAGE FAIL
; 8762 =0000
; 8763 CLEANUP:
; 8764 CLEANED: ;(0) NORMAL CASE
; 8765 END STATE, SKIP IRPT, ;NO MORE CLEANUP NEEDED
; 8766 J/PFT2 ;HANDLE PAGE FAIL OR INTERRUPT
; 8767 [AR]_WORK[SV.ARX], ;(1) BLT
; 8768 J/BLT-CLEANUP
; 8769 [PC]_[PC]+1, ;(2) MAP
; 8770 J/MAPDON
; 8771 STATE_[EDIT-SRC], ;(3) SRC IN STRING MOVE
; 8772 J/STRPF
; 8773 STATE_[EDIT-DST], ;(4) DST IN STRING MOVE
; 8774 J/STRPF
; 8775 STATE_[SRC], ;(5) SRC+DST IN STRING MOVE
; 8776 J/BACKD
; 8777 STATE_[EDIT-DST], ;(6) FILL IN MOVSRJ
; 8778 J/STRPF4
; 8779 STATE_[EDIT-SRC], ;(7) DEC TO BIN
; 8780 J/PFDBIN
; 8781 STATE_[EDIT-SRC], ;(10) SRC+DST IN COMP
; 8782 J/CMSDST
; 8783 END STATE, J/BACKS ;(11) EDIT SRC FAIL
; 8784 END STATE, J/BACKD ;(12) EDIT DST FAIL
; 8785 STATE_[EDIT-SRC], ;(13) SRC+DST IN EDIT
; 8786 J/BACKD
; 8787 =
; 8788
```

; T10KL.MCR[10,1141]
; PAGEF.MIC[10,1141]

15:34 27-JULY-1984
12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 244
PAGE FAIL REFIL LOGIC

U 2700, 1603,0551,1103,4374,4007,0701,0000,0000,0500
U 2701, 0770,3551,1313,4374,0007,0700,0000,0001,0000
U 1603, 2702,3443,0300,4174,4007,0700,0200,0021,1016
U 2702, 1606,3333,0006,4175,5007,0701,0210,0000,0002
U 2703, 2706,4521,1205,4074,4007,0700,0000,0000,0000
U 1604, 2704,4221,0013,4170,4007,0370,0000,0000,0000
U 2704, 1500,3441,0603,4174,4003,7700,0200,0003,0001
U 2705, 2701,1111,0701,4174,4007,0700,4000,0000,0000

```
; 8789 =0
; 8790 PFT2: [AR]_[UBR]+#, ;PREPARE TO STORE PFW
; 8791 #/500, 3T,
; 8792 J/PFT10
; 8793 PFT3: TAKE INTERRUPT ;PROCESS INTERRUPT
; 8794 PFT10: VMA_[AR], ;WHERE TO STORE PFW
; 8795 VMA_PHYSICAL WRITE
; 8796 =0 MEM_WRITE, ;STORE PFW
; 8797 MEM_[BRX],
; 8798 CALL [NEXTAR] ;ADVANCE POINTER TO
; 8799 ;PREPARE TO STORE PC
; 8800 .IF/KLPAGE
; 8801 .IF/KIPAGE
; 8802 TL [EBR], #/400000 ;KL PAGING?
; 8803 =0
; 8804 .ENDIF/KIPAGE
; 8805 [BR]_PC WITH FLAGS ;GET OLD PC
; 8806 .ENDIF/KLPAGE ;STORE OLD PC
; 8807 MEM_[BR],
; 8808 .IF/KIPAGE J/EAPF1
; 8809 ;
; 8810 MEM_WRITE,
; 8811 MEM_[BR],
; 8812 J/EAPF1
; 8813 .ENDIF/KIPAGE
; 8814
; 8815 MAPDON: END STATE, ;CLEAR MAP BIT
; 8816 SKIP IRPT ;ANY INTERRUPT?
; 8817 =0 [AR]_[BRX], ;RETURN PAGE FAIL WORD
; 8818 EXIT
; 8819 [PC]_[PC]-1, J/PFT3 ;INTERRUPTED OUT OF MAP
; 8820 ; RETRY INSTRUCTION
; 8821
```

; T10KL.MCR[10,1141] 15:34 27-JULY-1984
; PAGEF.MIC[10,1141] 12:32 26-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 245
PAGE FAIL REFIL LOGIC

U 2706, 1606,3333,0005,4175,5007,0701,0210,0000,0002
U 2707, 1605,3333,0001,4175,5007,0701,0200,0000,0002

U 1605, 2736,0111,0703,4174,4007,0700,0200,0024,1016

U 1606, 0001,0111,0703,4170,4004,1700,0200,0023,1016

```
; 8822
; 8823 .IF/KLPAGE
; 8824 =0
; 8825 EAPF: MEM WRITE, MEM_[BR], ;STORE FLAGS
; 8826 CALL [NEXTAR] ;STORE PC WORD
; 8827 MEM WRITE, MEM_[PC] ; ..
; 8828 .ENDIF/KLPAGE
; 8829
; 8830 EAPF1: [AR]_[AR]+1,
; 8831 VMA PHYSICAL READ,
; 8832 J/GOEXEC
; 8833
; 8834 NEXTAR: NEXT [AR] PHYSICAL WRITE, RETURN [1]
; 8835
```

; Number of microwords used:
; D words= 512
; U words= 2032, Highest= 2047

END

Cross Reference Listing

(U) A

AR

637 #												
641 #	2389	2427	2457	2458	2521	2569	2577	2716	2718	2721	2724	
2727	2730	2814	2824	2839	2856	2866	2876	2886	2917	2927	2937	
2972	2984	2991	2995	3005	3006	3008	3010	3011	3019	3042	3043	
3269	3274	3276	3288	3291	3294	3299	3300	3370	3387	3389	3454	
3457	3525	3526	3532	3552	3579	3594	3630	3641	3653	3707	3730	
3735	3770	3814	3818	3825	3827	3987	3992	4022	4057	4071	4087	
4098	4101	4104	4118	4145	4146	4151	4168	4184	4204	4316	4325	
4330	4340	4346	4365	4369	4387	4415	4419	4421	4422	4441	4473	
4476	4479	4493	4518	4535	4536	4541	4546	4554	4556	4558	4629	
4639	4641	4646	4647	4692	4697	4707	4711	4749	4753	4757	4761	
4937	4950	5035	5038	5050	5053	5059	5065	5100	5175	5178	5280	
5286	5290	5291	5321	5327	5346	5352	5353	5354	5355	5364	5369	
5389	5390	5402	5404	5413	5414	5434	5439	5440	5446	5481	5485	
5487	5490	5493	5497	5499	5501	5503	5504	5505	5506	5512	5514	
5515	5524	5529	5545	5571	5579	5584	5586	5630	5643	5662	5671	
5675	5690	5704	5706	5708	5714	5741	5748	5751	5763	5766	5774	
5776	5780	5785	5787	5803	5808	5811	5833	5836	5837	5841	5959	
5963	5967	5974	5985	5994	6009	6029	6033	6046	6049	6075	6159	
6166	6183	6205	6219	6246	6257	6258	6259	6272	6306	6336	6347	
6385	6400	6428	6433	6462	6463	6464	6465	6467	6468	6471	6476	
6525	6546	6553	6554	6580	6618	6630	6631	6638	6658	6661	6664	
6668	6669	6671	6675	6677	6731	6732	6737	6738	6745	6748	6752	
6821	6843	6849	6868	6886	6890	6893	6894	6895	6896	6918	6943	
6944	6948	6954	6955	6959	7029	7032	7053	7055	7057	7059	7062	
7064	7066	7102	7108	7116	7120	7123	7141	7147	7149	7154	7174	
7251	7332	7334	7336	7338	7340	7342	7344	7346	7348	7370	7376	
7438	7448	7454	7457	7458	7475	7483	7533	7544	7557	7559	7585	
7590	7617	7620	7648	7666	7667	7726	7823	7831	7862	8192	8202	
8306	8315	8318	8320	8322	8326	8329	8340	8346	8351	8394	8397	
8402	8409	8417	8424	8442	8460	8464	8502	8505	8520	8521	8542	
8568	8571	8574	8592	8598	8603	8609	8623	8640	8753	8794		

ARX

642 #	2489	2771	2772	3710	3712	3725	3754	4084	4096	4105	4106	
4123	4126	4128	4150	4170	4186	4206	4209	4229	4358	4386	4438	
4464	4465	4544	4636	4637	4645	4714	4904	4911	4912	4940	4943	
4964	5094	5116	5288	5326	5358	5544	5565	5568	5628	5645	5667	
5687	5707	5712	5758	5797	5804	6047	6088	6095	6129	6163	6213	
6222	6232	6241	6264	6265	6273	6305	6309	6386	6423	6431	6435	
6437	6754	6757	6803	6871	6945	7054	7058	7060	7063	7065	7094	
7096	7282	7295	7299	7489	7672	7674	7732	7959	8293	8301	8349	

BR

643 #	2188	2385	2420	2422	2426	2907	3038	3071	3072	3075	3076	
3077	3078	3093	3096	3099	3102	3106	3109	3111	3143	3144	3147	
3148	3429	3432	3435	3438	3441	3444	3447	3450	3775	3780	3808	
3833	3841	3842	3989	3994	4092	4224	4389	4391	4399	4403	4466	
4469	4543	4547	4548	4549	4552	4557	4588	4730	4732	4791	4792	
4793	4794	4795	4802	4811	4815	4824	4885	4886	4887	4888	4889	
4910	4953	4959	5062	5102	5104	5114	5119	5203	5287	5294	5295	
5299	5300	5356	5359	5393	5444	5527	5551	5553	5563	5577	5656	
5674	5677	5913	5925	5926	5928	5929	5956	5965	6010	6015	6040	
6043	6134	6199	6202	6203	6216	6224	6230	6353	6363	6374	6383	
6387	6405	6458	6473	6511	6522	6589	6845	6846	6887	6888	6928	
6936	6947	6951	7028	7031	7051	7085	7087	7090	7132	7140	7153	
7155	7180	7181	7188	7189	7195	7250	7261	7268	7337	7339	7341	
7347	7349	7385	7398	7495	7518	7519	7520	7561	7563	7591	7595	

	7597	7601	7603	7627	7633	7724	8200	8264	8267	8406	8526	8728
	8731											
BRX	644 #	4120	4148	4154	4225	4248	4268	4272	4276	4280	4286	4290
	4294	4298	4375	4428	4468	4580	4583	4594	5041	5093	5122	5130
	5137	5161	5246	5249	5319	5320	5322	5361	5366	5556	5573	5639
	5653	5654	5696	5917	6016	6086	6097	6098	6122	6171	6179	6260
	6289	6302	6311	6316	6324	6326	6335	6341	6346	6358	6397	6399
	6417	6426	6453	6455	6457	6488	6491	6507	6519	6521	6601	6603
	6606	6672	6679	6773	6777	6781	6785	6789	6793	6809	7077	7197
	7198	7199	7297	7447	7502	7503	7566	7649	8196	8249	8250	8253
	8259	8261	8431	8434	8437	8459	8467	8509	8513	8515	8517	8532
	8536	8703	8712	8715	8720	8817						
EBR	646 #	6975	7178	7450	7493	8304						
FLG	649 #	3621	5508	5609	5610	5617	5744	5747	5750	5753	5813	5820
	5821	6597	8246	8325	8345	8413	8422	8433	8439	8471	8492	8512
	8531	8533	8576	8601	8606	8630	8639	8694	8696	8699	8708	8793
HR	640 #	2289	2295	2301	2307	2316	2320	2329	2334	2435	2783	2788
	3529	3538	3541	3543	3544	3545	3564	3568	3578	3583	3586	3606
	3627	3640	3919	3921	3923	3925	3927	3929	3931	3933	3939	3959
	4037	5915	6908	6985	6991	7000	7001	7002	7006	7007	7008	7009
	7014	7015	7016	7019	7020	7021	7022	7023	7024	7025	7026	7218
MAG	7240	7335	7651	7665	7794	7803	7805	7807	7809	7811	7813	
	638 #	2424	4130	4133	4152	4155	4156	4169	4175	4177	4211	4213
	4215	4393	4411	4433	4437	4440	4481	4483	4484	4611	4616	4621
	4627	4628	5223	5224	5547	5638	5693	5719	5754	5789	5793	5795
MASK	648 #	2150	2151	2152	2155	2158	2191	2202	2238	2242	2245	2248
	2257	2260	2263	2266	2517	2740	2761	2968	3016	3580	3628	3642
	3663	3943	3960	3983	3986	4031	4091	4227	4261	4395	4406	4434
	4559	4771	4891	4902	5208	5228	5229	5230	5244	5449	5509	5582
	5632	5722	5791	5880	5882	5884	5886	5888	5890	5892	5894	5896
	6131	6133	6155	6270	6592	7249	7292	7293	7294	7428	7429	7430
	7431	7432	7433	7434	7440	7459	7469	7487	7501	7579	7582	7618
	7629	7635	7652	7708	7861	8225	8227	8241	8750	8805		
ONE	645 #	2180	2184	2229	2231	2276	2363	2414	2417	2947	3338	3404
	3420	3487	3501	3527	3536	3539	3614	3632	3635	3979	4026	4042
	4321	4736	5033	5101	5142	5144	5149	5166	5167	5251	5252	5436
	5448	5523	5672	5830	5951	5992	6025	6031	6052	6070	6080	6099
	6100	6135	6136	6178	6242	6247	6310	6322	6329	6349	6356	6361
	6370	6380	6384	6388	6389	6425	6439	6450	6502	6526	6620	6642
	6695	6818	6819	6825	6847	6889	6931	7256	7279	7309	7361	7645
	7722	7847	7933	7940	7943	7946	7949	7952	8759	8769	8819	8830
	8834											
PC	639 #	2228	2232	2250	2343	2405	2430	3326	3453	3458	3689	3806
	3812	3830	4216	4218	4693	4702	6964	7352	7471	7623	7858	
PI	650 #	3592	3593	3607	7010	7011	7329	7343	7345	7350	7360	7362
	7421	7422	7423	7424	7425	7426	7427	7436				
TO	652 #	4196	4200	4436	4447	4454	4457	4459	4460	5564	5572	5598
	5600	5601	5602	5605	5613	5616	5618	5658	5724	5732	5733	6354
	6421	7363	7499									
T1	653 #	4413	4553	5219	5221	5225	5240	5243	5588	5673	5676	5695
	5705											
UBR	647 #	3945	3985	6967	7113	7135	7187	7194	8296	8790		
XWD1	651 #	3509	3679	7485								
	1315 #											
DBLAC	1319 #	2775										

(D) A

DFP	1325 #	5534	5535	5623	5684								
DREAD	1318 #	2766	2767	4079	4080	4163	4381						
DSHIFT	1321 #	2957	2958										
FP	1323 #	5260	5261	5262	5263	5265	5266	5268	5269	5270	5271	5273	
	5274	5305	5306	5307	5309	5311	5312	5334	5335	5336	5338	5340	
	5341	5419	5420										
FPI	1322 #	5264	5272	5310	5339								
IOT	1326 #	6996	6997	7204	7525	7526	7527	7528	7536	7537	7538	7539	
	7550	7551	7552	7553	7818	7819	8188						
RD-PF	1324 #	2547	2552	2557	2562	2600	2605	2610	2615	2620	2625	2630	
	2635	2642	2647	2652	2657	2662	2667	2672	2677	2807	2817	2827	
	2832	2849	2859	2869	2879	2899	2910	2920	2930	4050	4064	4112	
	4137	4304	4309										
READ	1316 #	2550	2555	2560	2565	2584	2602	2603	2607	2608	2613	2618	
	2623	2628	2633	2638	2644	2645	2649	2650	2655	2660	2665	2670	
	2675	2680	2809	2810	2819	2820	2829	2830	2834	2835	2842	2851	
	2852	2861	2862	2871	2872	2881	2882	2901	2902	2912	2913	2922	
	2923	2932	2933	3186	3187	3188	3189	3190	3191	3202	3203	3204	
	3205	3206	3207	3208	3209	3220	3221	3222	3223	3224	3225	3226	
	3227	3237	3238	3239	3240	3241	3242	3243	3244	3359	3360	3361	
	3362	3363	3364	3365	3366	3376	3377	3378	3379	3380	3381	3382	
	3383	3393	3394	3395	3396	3397	3398	3399	3400	3409	3410	3411	
	3412	3413	3414	3415	3416	3647	4052	4053	4066	4067	4114	4115	
	4139	4140	4306	4307	4311	4312	4682	4683	4684	4685	4686	5384	
SHIFT	1320 #	2952	2953	2954									
WRITE	1317 #	2549	2554	2559	2564	2612	2617	2622	2627	2632	2637	2654	
	2659	2664	2669	2674	2679	2776	2844	2845					
(U) ACALU	1219 #												
AC+N	1221 #	2173	2174	2181	2371	2383	2489	2772	2781	3022	3043	3087	
	3117	4084	4096	4175	4177	4179	4187	4189	4196	4198	4210	4211	
	4212	4213	4214	4215	4327	4411	4433	4437	4440	4454	4455	4457	
	4460	4479	4481	4483	4484	4610	4611	4614	4615	4616	4619	4620	
	4621	4627	4628	5547	5638	5691	5693	5699	5758	5766	5804	5811	
	5955	5963	5973	5976	6006	6009	6022	6029	6036	6046	6084	6094	
	6114	6129	6146	6161	6163	6189	6192	6194	6201	6202	6203	6205	
	6211	6213	6216	6222	6224	6252	6254	6255	6259	6266	6267	6268	
	6272	6273	6280	6282	6287	6293	6309	6320	6332	6341	6347	6358	
	6361	6363	6367	6380	6386	6387	6510	6541	6546	6551	6588	6590	
	6694	6695	6807	6814	6821	6840	6842	6849	6869	6890	6926	6928	
	6934	6936	6940	6943	6948								
B	1220 #												
(D) ACDISP	1346 #	3516	6996	6997	7204								
(U) ACN	1222 #	2371	2383	2489	2772	2781	3022	3043	3087	3117	4084	4096	
	4175	4177	4179	4187	4189	4196	4198	4210	4211	4212	4213	4214	
	4215	4327	4411	4433	4437	4440	4454	4455	4457	4460	4479	4481	
	4483	4484	4610	4611	4614	4615	4616	4619	4620	4621	4627	4628	
	5547	5638	5691	5693	5699	5758	5766	5804	5811	6293	6309	6320	
	6367	6386											
BINO	1229 #	2173	6189	6211	6216	6224	6255	6266	6267	6268	6272	6280	
BIN1	1230 #	2174	2181	6161	6163	6192	6194	6201	6202	6203	6205	6213	
	6222	6252	6254	6259	6273	6282							
DLEN	1226 #	5955	5963	5976	6006	6009	6036	6046	6084	6094	6129	6287	
	6332	6341	6347	6358	6361	6363	6380	6387	6694	6695	6807	6821	
	6840	6940	6943	6948									
DSTP	1227 #	5973	6146	6541	6546	6590	6814	6869	6890	6934	6936		

Produced on Advanced Information Services Electronic Laser Printer. PNO1E55. DTN. 223-7881

	MARK	1228 #	6510	6588																		
	SRCLN	1224 #																				
	SRCP	1225 #	6022	6029	6114	6551	6842	6849	6926	6928												
(U) AD		534 #	2155	2158	2191	2202	2238	2242	2245	2248	2257	2260	2263									
		2266	2320	2517	2738	2740	2759	2761	3580	3663	3943	3983	4031									
		4395	4406	5221	5880	5882	5884	5886	5888	5890	5892	5894	5896									
		5926	6155	6270	6592	7249	7292	7293	7294	7428	7429	7430	7431									
		7432	7433	7434	7440	7469	7487	7501	7579	7582	7618	7629	7635									
		7652	7708	7861	8225	8227	8241	8750														
A		563 #	2150	2152	2228	2232	2250	2295	2307	2334	2343	2385	2426									
		2427	2430	2489	2521	2772	2788	2972	2984	2991	2995	3005	3006									
		3008	3010	3011	3019	3038	3043	3071	3072	3075	3076	3077	3078									
		3093	3096	3099	3102	3106	3109	3143	3144	3147	3148	3300	3326									
		3389	3429	3441	3453	3454	3457	3458	3525	3526	3532	3552	3568									
		3579	3594	3630	3641	3653	3707	3710	3712	3730	3735	3808	3814									
		3825	3830	3842	4118	4120	4123	4145	4146	4148	4150	4168	4170									
		4184	4186	4204	4206	4218	4224	4225	4227	4248	4261	4316	4325									
		4340	4346	4375	4387	4389	4403	4415	4419	4434	4438	4441										
		4454	4460	4464	4473	4493	4541	4546	4547	4548	4553	4554	4556									
		4558	4629	4693	4702	4707	4714	4730	4753	4761	4815	4891	4904									
		4940	4959	4964	5038	5041	5059	5094	5114	5122	5130	5178	5203									
		5208	5223	5224	5228	5229	5230	5243	5244	5246	5286	5287	5288									
		5299	5321	5322	5326	5327	5346	5354	5356	5361	5364	5369	5402									
		5404	5434	5436	5439	5440	5446	5481	5485	5487	5493	5501	5503									
		5504	5505	5523	5524	5529	5556	5563	5564	5568	5571	5572	5579									
		5584	5586	5598	5600	5601	5602	5605	5613	5616	5618	5628	5630									
		5632	5639	5643	5645	5656	5662	5667	5671	5672	5675	5687	5690									
		5712	5722	5724	5732	5733	5741	5748	5751	5763	5766	5780	5785									
		5787	5808	5811	5833	5841	5929	5974	5994	6009	6029	6040	6046									
		6075	6086	6122	6129	6163	6202	6222	6224	6259	6272	6273	6309									
		6335	6341	6346	6347	6354	6358	6363	6385	6386	6387	6421	6455									
		6464	6467	6471	6473	6511	6521	6546	6553	6589	6606	6631	6661									
		6668	6679	6731	6748	6754	6845	6849	6868	6887	6890	6894	6896									
		6928	6936	6943	6944	6948	6955	7010	7011	7102	7132	7147	7149									
		7174	7178	7180	7187	7188	7194	7195	7197	7199	7297	7329	7352									
		7370	7376	7447	7454	7475	7483	7502	7520	7544	7557	7559	7566									
		7591	7595	7601	7623	7627	7633	7667	7724	7726	7732	7823	7831									
		7858	7959	8202	8261	8264	8306	8326	8329	8349	8363	8496	8502									
		8520	8526	8542	8623	8794	8817															
A+B		536 #	2180	2184	2229	2231	2276	2363	2414	2417	2783	3111	3338									
		3404	3614	3632	3635	3775	3780	3945	3979	3985	4026	4042	4196									
		4200	4229	4285	4289	4369	4465	4466	4469	4479	4552	4557	4583									
		4587	4593	4736	5142	5144	5149	5166	5167	5251	5252	5300	5444									
		5448	5527	5577	5658	5951	6015	6047	6052	6070	6099	6100	6205									
		6219	6230	6232	6242	6310	6322	6329	6349	6370	6384	6439	6450									
		6458	6502	6526	6620	6642	6669	6695	6847	6871	6889	6931	6951									
		6967	6975	7250	7256	7279	7309	7450	7485	7495	7503	7722	7847									
		7933	7940	7943	7946	7949	7952	8293	8301	8386	8769	8830	8834									
A+Q		535 #	4436	4447	4468	5101	5565	5573	5674	5677	5830											
A-.25		547 #																				
A-B-.25		552 #	4365	4421	4422	5100	5119	5137	5161	5249	5358	5366	5714									
		5963	6098	6134	6353																	
A-D-.25		548 #	6183	6423	6428	6476																
A-Q-.25		551 #																				
A.AND.B		568 #	4154	4169	4386	4457	4910	5509	5758	5791	5793	5804	5985									

Produced on Advanced Information Services Electronic Laser Printer, PKO IES6, DTN. 223-7881

Cross Reference Listing

A.AND.Q	6131	7029	7032	7362	7518	8594							
	567 #	4152	4393	4437	4481	4483	4559	4611	4616	4621	4627	4628	
	4824	4902	4911	5754	5789								
A.EQV.B	595 #												
A.EQV.Q	594 #	4130	4133	4155	4156	4211	4213	4215	4484				
A.OR.B	560 #	2907	4092	4912	5065	5917	6016	6049	6088	6246	6677	7054	
	7058	7065	7096	7123	7140	7141	7341	7347	7489	7585	7620	8517	
	8731												
A.OR.Q	559 #	5219	5240	5588									
A.XOR.B	587 #	5795	7295										
A.XOR.Q	586 #												
B	562 #	2160	2168	2170	2171	2174	2176	2183	2185	2376	2396	2495	
	2527	2533	2538	2541	2572	2591	2736	2743	2750	2757	2787	3068	
	3084	3140	3285	3329	3332	3335	3341	3344	3347	3406	3560	3608	
	3619	3659	3695	3701	3740	3746	3817	3963	3967	3969	3975	3997	
	4003	4034	4041	4122	4267	4271	4293	4297	4328	4334	4348	4353	
	4355	4360	4427	4429	4446	4462	4471	4532	4539	4560	4690	4699	
	4703	4718	4721	4737	4818	4884	4892	4900	4949	4962	4982	5014	
	5021	5285	5289	5372	5374	5406	5410	5443	5483	5502	5570	5669	
	5678	5727	5745	5783	5919	5922	5935	5937	5938	5968	5980	6014	
	6030	6056	6057	6058	6073	6115	6118	6120	6124	6137	6148	6156	
	6164	6295	6297	6340	6359	6372	6376	6378	6427	6434	6460	6474	
	6479	6501	6505	6557	6577	6591	6643	6651	6725	6746	6768	6800	
	6850	6852	6874	6914	6965	7067	7069	7070	7072	7075	7079	7156	
	7157	7223	7227	7231	7235	7239	7253	7264	7269	7285	7288	7304	
	7308	7310	7318	7319	7371	7383	7389	7396	7621	7663	7686	7692	
	7694	7728	7730	7839	7846	7911	7920	7926	7930	7937	7932	7934	
	7935	7936	7938	7939	7941	7942	7944	7945	7947	7948	7950	7951	
	7953	7954	7956	8198	8208	8209	8211	8212	8219	8229	8231	8233	
	8235	8239	8240	8275	8374	8381	8383	8404	8538	8596	8725	8756	
	8761	8797	8825	8827									
B-.25	546 #												
B-A-.25	544 #	3420	3527	3536	3539	4275	4279	4549	4580	6010	6031	6135	
	6136	6247	6336	6388	6425	6818	6947	7261	7385	7398	7645	8759	
	8819												
D	566 #	2149	2153	2178	2179	2181	2219	2269	2339	2353	2357	2362	
	2366	2370	2371	2383	2384	2389	2394	2405	2411	2569	2588	2689	
	2691	2716	2718	2721	2722	2724	2727	2728	2730	2745	2747	2752	
	2754	2781	2965	2982	2988	3000	3003	3041	3087	3090	3269	3274	
	3387	3432	3435	3438	3444	3447	3450	3473	3549	3556	3573	3588	
	3593	3613	3617	3678	3689	3717	3723	3750	3759	3770	3772	3806	
	3812	3827	3828	3840	3841	3845	3959	4002	4104	4119	4128	4147	
	4151	4189	4194	4209	4212	4214	4216	4317	4326	4327	4409	4538	
	4692	4697	4711	4713	4717	4732	4765	4791	4792	4793	4794	4795	
	4811	4885	4886	4887	4888	4889	4937	4946	4953	4967	4969	4973	
	4977	4979	4983	5012	5017	5040	5053	5055	5093	5102	5106	5113	
	5157	5175	5176	5201	5206	5283	5316	5348	5412	5490	5506	5548	
	5558	5650	5701	5836	5912	5931	5940	5955	5958	5972	5973	5976	
	5998	6001	6006	6022	6023	6033	6043	6060	6061	6062	6063	6071	
	6077	6085	6094	6096	6114	6116	6123	6127	6140	6142	6146	6158	
	6161	6170	6189	6192	6194	6201	6211	6252	6254	6255	6260	6287	
	6292	6293	6294	6306	6315	6320	6332	6338	6348	6351	6352	6367	
	6379	6381	6382	6400	6403	6452	6457	6462	6463	6503	6510	6525	
	6541	6551	6560	6564	6579	6588	6590	6594	6615	6622	6624	6655	
	6671	6692	6694	6732	6757	6807	6812	6814	6821	6823	6840	6842	

D+A

D+Q
D-.25
D-A-.25
D-Q-.25
D.AND.A

D.AND.Q
D.EQV.A
D.EQV.Q
D.OR.A

D.OR.Q
D.XOR.A
D.XOR.Q
Q

6843	6846	6869	6886	6888	6898	6913	6926	6930	6934	6940	6959
6964	6983	7003	7004	7036	7038	7050	7061	7084	7085	7093	7105
7152	7209	7211	7213	7215	7217	7222	7226	7230	7234	7238	7246
7251	7255	7260	7266	7267	7278	7298	7302	7317	7323	7337	7350
7360	7379	7381	7392	7394	7418	7444	7455	7471	7473	7492	7497
7515	7547	7588	7655	7658	7660	7668	7682	7713	7716	7718	7827
7834	7866	7913	7918	7922	7925	7937	7955	7957	7958	7960	8195
8199	8201	8210	8216	8221	8288	8291	8310	8358	8375	8378	8392
8402	8452	8454	8483	8499	8539	8540	8541	8547	8567	8628	8726
8736	8742	8751	8755	8767	8771	8773	8775	8777	8779	8781	8785
540 #	2289	2301	2316	2329	3487	3509	3564	3679	3725	3754	3818
3833	4057	4084	4087	4749	4757	5035	5104	5116	5925	5928	5992
6025	6080	6178	6203	6213	6216	6356	6374	6417	6426	6433	6435
6554	6630	6737	6745	6819	6825	6893	6895	7299	7448	7493	7666
7672	7674	8296	8304	8318	8320	8397	8568	8790			
541 #	4985	4993	5024	5043							
558 #											
556 #	3370	3501	4071	4096	4098	4101	5050	6361	6380		
557 #											
575 #	2422	2424	2435	2457	2771	2814	2937	2968	3016	3276	3299
3529	3538	3541	3543	3544	3545	3578	3583	3586	3592	3606	3607
3627	3628	3640	3642	3919	3921	3923	3925	3927	3929	3931	3933
3939	3960	3986	3987	3992	4037	4105	4175	4177	4358	4411	4433
4440	4636	4771	4943	4950	5062	5225	5290	5294	5319	5352	5389
5393	5413	5508	5547	5638	5673	5676	5693	5744	5747	5750	5753
5813	5913	5915	5956	5959	6095	6097	6133	6159	6166	6179	6199
6241	6257	6264	6289	6305	6324	6383	6397	6399	6431	6437	6453
6465	6468	6488	6491	6507	6522	6638	6672	6675	6738	6752	6908
6918	6985	6991	7000	7001	7002	7006	7007	7008	7009	7014	7015
7016	7019	7020	7021	7022	7023	7024	7025	7026	7053	7055	7057
7059	7062	7064	7066	7087	7087	7108	7113	7116	7120	7135	7153
7154	7181	7189	7198	7218	7240	7282	7332	7334	7335	7336	7338
7340	7342	7344	7346	7348	7459	7499	7533	7590	7597	7603	7617
7649	7651	7665	7794	7803	7805	7807	7809	7811	7813	8196	8200
8249	8253	8315	8322	8325	8340	8345	8394	8409	8413	8417	8422
8433	8437	8439	8459	8464	8492	8505	8512	8515	8531	8533	8571
8574	8576	8598	8601	8603	8606	8609	8630	8640	8694	8698	8701
8706	8712	8728	8753	8805							
576 #	3113	4511	5211	5217	5231	5237	5324	5633			
599 #	2886										
600 #											
564 #	2151	2420	2458	2866	3294	3989	3994	4106	4459	4544	5291
5295	5320	5353	5414	5553	5609	5610	5617	5654	5706	5820	6171
6258	6302	6311	6316	6326	6597	6601	6603	6664	6777	6781	6785
6793	6809	6945	6954	7155	7268	7343	7421	7422	7423	7424	7425
7426	7427	7561	7862	8192	8246	8250	8351	8431	8434	8442	8460
8467	8509	8513	8530	8532	8536	8592	8699	8703	8708	8715	8720
8793											
565 #	3115	4396									
591 #	2188	2856	3291	4413	7457	7458	8259				
592 #											
561 #	3117	4124	4132	4179	4187	4187	4198	4210	4352	4407	4424
4430	4435	4439	4445	4453	4972	5134	5160	5248	5371	5376	5447
5489	5513	5607	5614	5634	5646	5657	5716	5718	5762	5807	6104
6143	6545										

Q-.25	545 #											
Q-A-.25	543 #	4428	5033									
Q-D-.25	549 #	4976										
ZERO	572 #	2161	2161	2163	2166	2173	2175	2177	2194	2216	2400	2804
	2981	3022	3042	3271	3609	3970	4223	4249	4319	4455	4533	4800
	4813	4819	4820	4823	4895	4896	4899	4955	5098	5140	5164	5250
	5297	5368	5379	5391	5395	5397	5398	5411	5424	5516	5689	5717
	5756	5772	5800	5981	5983	6017	6036	6084	6089	6101	6112	6141
	6262	6280	6282	6300	6334	6390	6394	6398	6401	6456	6580	6618
	6658	6728	6803	7028	7031	7094	7128	7138	7182	7258	7333	7378
	7403	7404	7436	7439	7446	7460	7675	7720	7841	7844	7914	8245
	8267	8346	8406	8424	8508	8528	8570	8740	8746	8765	8783	8784
	8815											
-A-.25	555 #	2577	2947	4321	4399	4476	4518	4536	4543	4637	4641	4645
	4647	5280	5355	5359	5390	5499	5515	5544	5545	5696	5707	5708
	5776	5797	5803	6265	6389	7361						
-B-.25	554 #	5697										
-D-.25	550 #	4450	4610	4615	4620	4625	5996	6268	6939	6953		
-Q-.25	553 #	4398	4449	4475	4516	4517	4534	5377	5482	5484	5486	5488
	5771											
.NOT.A	598 #	2839	2876	2917	2927	3288	4091	4330	4535	4639	4646	5449
	5497	5512	5514	5695	5705	5774	5965	5967	6519	7363	7438	7648
	8639											
.NOT.A.AND.B	579 #	4126	5582	7060	7063	7339	7349	7519				
.NOT.A.AND.Q	578 #	5719										
.NOT.B	597 #	5802	6174									
.NOT.D	601 #	2896	2906	4614	4619	4624	6013	6041	6045	6181	6185	6267
	6957											
.NOT.D.AND.A	583 #	2824	3621	4022	5551	5653	5704	5821	6405	6773	6789	7051
	7077	7345	7563	8471	8521	8696						
.NOT.D.AND.Q	584 #											
.NOT.Q	596 #	4909	5770	5984								
O+A	539 #	5837										
O+B	538 #											
O+D	542 #											
O+Q	537 #	5583	5730									
(U) AD PARITY OK	711 #	2181	2357	2370	2371	2588	2689	2691	2722	2728	2781	2965
	2968	2982	3016	3473	3717	3750	3772	3828	3840	4147	4189	4326
	4717	5113	5201	5283	5316	5348	5412	5558	5955	5958	5973	5976
	6006	6022	6094	6096	6114	6146	6158	6161	6189	6192	6194	6201
	6211	6252	6254	6255	6287	6292	6293	6294	6315	6320	6332	6367
	6452	6510	6551	6588	6590	6694	6807	6814	6840	6842	6869	6926
	6934	6940	7547	7834	8222							
(U) ADFLGS	1115 #	2578	3404	3420	3487	3501	4058	4072	4089	4099	4102	4229
	4640	4642										
(U) AREAD	1149 #	2321										
(U) B	657 #											
AR	661 #	2158	2160	2175	2238	2242	2245	2257	2260	2263	2320	2353
	2357	2362	2370	2376	2389	2394	2396	2405	2414	2414	2417	2417
	2426	2427	2457	2458	2495	2521	2527	2533	2538	2541	2569	2572
	2577	2689	2691	2716	2718	2721	2722	2724	2727	2728	2730	2736
	2738	2740	2743	2745	2747	2750	2752	2754	2757	2759	2761	2804
	2814	2824	2839	2856	2866	2876	2886	2896	2907	2907	2917	2927
	2937	2947	2965	2968	2972	2982	2984	2988	2991	2995	3000	3003
	3005	3006	3008	3010	3011	3041	3042	3043	3068	3084	3140	3269

3274	3288	3291	3294	3299	3300	3329	3332	3335	3341	3344	3347
3370	3387	3389	3404	3404	3406	3420	3614	3614	3632	3635	3635
3642	3770	3775	3780	3828	3959	3960	3967	3986	4002	4003	4022
4031	4034	4042	4057	4071	4087	4092	4092	4098	4101	4104	4124
4130	4132	4133	4150	4151	4168	4216	4319	4321	4326	4328	4353
4375	4409	4415	4419	4462	4466	4466	4469	4469	4473	4476	4479
4493	4518	4532	4535	4536	4541	4546	4549	4552	4554	4556	4557
4557	4558	4588	4624	4625	4629	4639	4641	4646	4647	4690	4699
4703	4707	4718	4721	4730	4732	4736	4736	4737	4765	4803	4818
4820	4823	4824	4884	4885	4886	4887	4888	4889	4892	4900	4909
4910	4910	4912	4912	4967	4969	4979	4982	4983	5012	5021	5035
5045	5050	5053	5055	5059	5137	5161	5175	5176	5176	5249	5280
5287	5289	5290	5291	5300	5300	5326	5327	5348	5352	5353	5364
5369	5371	5372	5374	5376	5377	5402	5404	5406	5410	5412	5413
5414	5434	5439	5440	5443	5444	5444	5446	5448	5448	5481	5483
5485	5487	5490	5493	5497	5499	5501	5502	5503	5504	5505	5506
5509	5509	5512	5514	5515	5524	5527	5527	5529	5545	5572	5577
5577	5579	5584	5586	5630	5667	5671	5675	5678	5701	5704	5706
5708	5732	5733	5741	5745	5748	5751	5763	5766	5774	5776	5780
5783	5785	5787	5802	5808	5811	5833	5836	5837	5841	5880	5882
5884	5886	5888	5890	5892	5894	5896	5922	5951	5951	5958	5965
5967	5968	5983	5984	5992	5996	6009	6010	6010	6013	6015	6015
6016	6016	6022	6029	6030	6040	6041	6043	6045	6046	6047	6047
6052	6052	6060	6099	6099	6100	6101	6114	6120	6127	6131	6131
6134	6134	6137	6141	6146	6148	6155	6156	6158	6166	6174	6178
6181	6185	6254	6257	6258	6259	6267	6268	6272	6292	6294	6297
6306	6315	6332	6346	6347	6348	6349	6349	6354	6372	6374	6376
6381	6385	6398	6400	6401	6403	6425	6428	6433	6439	6439	6450
6462	6463	6464	6465	6467	6468	6501	6521	6525	6526	6526	6546
6551	6553	6554	6557	6579	6580	6592	6594	6615	6618	6620	6620
6622	6630	6638	6642	6642	6643	6655	6658	6661	6664	6668	6671
6675	6692	6694	6695	6725	6731	6732	6737	6745	6746	6748	6754
6803	6807	6812	6814	6818	6819	6821	6823	6825	6840	6842	6845
6846	6847	6847	6849	6850	6852	6869	6874	6887	6888	6889	6889
6890	6898	6913	6914	6918	6926	6930	6931	6934	6939	6943	6947
6947	6948	6951	6951	6953	6954	6955	6957	6967	6967	6975	6975
7105	7108	7147	7149	7222	7223	7226	7227	7230	7231	7234	7235
7238	7239	7246	7250	7250	7251	7253	7255	7256	7256	7258	7260
7261	7261	7264	7266	7278	7288	7302	7317	7318	7319	7378	7379
7383	7385	7385	7389	7392	7396	7398	7398	7436	7440	7444	7447
7448	7450	7450	7455	7473	7483	7487	7489	7489	7544	7566	7591
7597	7603	7652	7655	7660	7666	7668	7672	7674	7675	7713	7722
7722	7827	7834	7861	7862	7935	7936	7937	7938	7955	7956	7957
8192	8202	8208	8296	8304	8310	8318	8320	8326	8329	8349	8351
8375	8378	8392	8397	8404	8409	8442	8452	8454	8460	8496	8499
8502	8505	8517	8517	8547	8568	8570	8574	8596	8628	8639	8736
8742	8751	8755	8756	8767	8790	8817	8830	8830	8834		
662 #	2181	2185	2366	2371	2400	2424	2489	2771	2772	2781	2787
3679	3710	3712	3717	3725	3750	3754	3827	3830	3943	3945	3945
3963	3979	4026	4084	4096	4105	4106	4122	4123	4128	4152	4155
4156	4169	4169	4177	4187	4200	4200	4206	4209	4223	4229	4249
4268	4272	4276	4280	4286	4290	4294	4298	4340	4352	4358	4424
4430	4439	4533	4538	4544	4560	4580	4580	4583	4583	4594	4636
4637	4645	4713	4717	4904	4911	4937	4940	4943	4946	4949	5093
5142	5142	5167	5252	5286	5354	5355	5366	5544	5646	5658	5658

ARX

5693	5716	5791	5791	5793	5793	5795	5795	5797	5830	5974	5976	
5994	6025	6031	6057	6062	6070	6070	6073	6075	6080	6094	6098	
6118	6124	6129	6135	6161	6163	6192	6213	6219	6219	6222	6232	
6232	6241	6252	6264	6265	6270	6273	6293	6295	6305	6309	6310	
6310	6320	6338	6367	6378	6382	6384	6384	6386	6423	6431	6435	
6437	6577	6757	6800	6868	6871	6944	6945	6964	6965	7053	7055	
7093	7094	7282	7285	7293	7297	7299	7310	7446	7485	7485	7495	
7495	7618	7620	7629	7635	7658	7716	7720	7732	7839	7844	7847	
7911	7913	7918	7922	7925	7926	7930	7933	7940	7943	7946	7949	
7952	7958	7960	8212	8288	8291	8346	8363	8381	8383	8424	8528	
8539	8592											
BR	663 #	2183	2384	2385	2411	2420	2422	2588	2591	2906	3016	3038
	3071	3072	3075	3076	3077	3078	3090	3093	3096	3099	3102	3106
	3109	3111	3143	3144	3147	3148	3271	3276	3285	3429	3432	3435
	3438	3441	3444	3447	3450	3473	3487	3501	3509	3613	3619	3628
	3678	3689	3695	3701	3723	3740	3746	3772	3806	3808	3812	3817
	3825	3840	3841	3845	3983	3985	3985	3989	3994	3997	4091	4170
	4316	4325	4334	4348	4355	4360	4365	4369	4387	4389	4391	4399
	4403	4421	4422	4539	4543	4547	4548	4692	4697	4711	4771	4791
	4792	4793	4794	4795	4811	4813	4815	4950	4953	4955	4959	4962
	4977	4987	5014	5026	5062	5065	5101	5102	5104	5114	5116	5203
	5283	5285	5288	5294	5295	5299	5346	5390	5436	5449	5523	5548
	5551	5553	5564	5634	5672	5690	5697	5714	5912	5917	5917	5919
	5925	5926	5928	5931	5935	5937	5938	5940	5955	5963	5973	5980
	6006	6014	6033	6056	6061	6133	6189	6194	6201	6202	6203	6205
	6211	6216	6224	6230	6230	6242	6242	6255	6334	6335	6336	6336
	6356	6359	6361	6363	6370	6370	6380	6387	6389	6405	6457	6458
	6460	6471	6473	6474	6510	6519	6588	6590	6591	6843	6886	6928
	6936	6940	6959	7003	7004	7010	7011	7028	7029	7031	7032	7036
	7038	7050	7051	7054	7054	7058	7058	7060	7060	7069	7070	7072
	7084	7085	7087	7090	7096	7096	7120	7128	7132	7152	7153	7155
	7156	7157	7178	7180	7181	7182	7187	7189	7194	7199	7209	7211
	7213	7215	7217	7249	7267	7268	7269	7292	7295	7298	7308	7323
	7329	7336	7337	7371	7381	7394	7459	7471	7493	7497	7503	7503
	7515	7520	7533	7547	7559	7561	7563	7579	7582	7585	7588	7595
	7601	7621	7627	7633	7724	7728	7939	8199	8219	8229	8231	8233
	8235	8261	8264	8267	8293	8301	8386	8402	8406	8520	8521	8540
	8725	8726	8728	8805	8825							
BRX	664 #	2179	2180	2180	4118	4120	4147	4148	4175	4194	4225	4248
	4346	4407	4465	5038	5041	5098	5100	5113	5119	5144	5149	5166
	5166	5201	5251	5251	5316	5319	5320	5322	5356	5358	5359	5361
	5547	5638	5639	5650	5653	5654	5687	5696	5915	5959	6049	6058
	6063	6086	6088	6096	6115	6122	6136	6159	6164	6171	6246	6246
	6260	6287	6289	6300	6302	6311	6316	6322	6322	6326	6329	6329
	6340	6341	6358	6388	6452	6479	6505	6507	6601	6603	6606	6651
	6669	6669	6672	6677	6677	6679	6768	6773	6777	6781	6785	6789
	6793	6809	7061	7063	7063	7065	7065	7067	7075	7077	7079	7188
	7195	7197	7198	7294	7428	7429	7430	7431	7432	7433	7434	7499
	7502	7557	7648	7708	7941	8195	8196	8198	8209	8210	8211	8221
	8225	8227	8239	8240	8241	8250	8253	8259	8275	8374	8431	8434
	8467	8509	8513	8515	8530	8532	8536	8538	8541	8703	8715	8720
	8731	8731	8797									
EBR	666 #	2163	7174	7944								
FLG	669 #	2166	2168	2170	2171	3621	5106	5140	5164	5250	5516	5609
	5610	5617	5756	5772	5800	5820	5821	5972	5998	6001	6017	6023

Produced on Advanced Information Services Electronic Laser Printer, PKC/IESs. DTN. 223-7881

	6071	6077	6085	6089	6112	6116	6123	6142	6170	6262	6352	6379
	6390	6394	6456	6503	6560	6564	6597	6624	6728	7948	8201	8246
	8471	8508	8594	8594	8696	8699	8708	8761	8765	8771	8773	8775
	8777	8779	8781	8783	8784	8785	8793	8815				
HR	660 #	2178	2184	2184	2269	2301	2316	2329	2339	2363	2435	3529
	3538	3541	3543	3544	3545	3549	3560	3564	3573	3578	3583	3586
	3588	3606	3627	3640	3653	3659	3663	3919	3921	3923	3925	3927
	3929	3931	3933	3939	3970	3975	4037	4041	4602	6908	6983	6991
	7000	7001	7002	7006	7007	7008	7009	7014	7015	7016	7019	7020
	7021	7022	7023	7024	7025	7026	7218	7240	7279	7304	7309	7335
	7663	7794	7803	7805	7807	7809	7811	7813	7866	7934		
MAG	658 #	2152	4227	4261	4386	4434	4457	4891	4896	4899	5208	5228
	5229	5230	5244	5632	5722	5758	5804	7931				
MASK	668 #	2149	2150	2151	3608	4126	5223	5224	5985	7920	7947	
ONE	665 #	2155	2174	2176	2783	6353	7942					
PC	659 #	2216	2229	2231	2276	2289	2295	3019	3338	3454	3457	3525
	3526	3527	3536	3539	3552	3579	3594	3617	3707	3759	3818	3833
	3969	6247	6502	6502	7475	7645	7730	7932	8759	8769	8769	8819
	8827											
PI	670 #	2194	7333	7339	7339	7341	7341	7343	7345	7347	7347	7349
	7349	7421	7422	7423	7424	7425	7426	7427	7518	7519	7519	7950
TO	672 #	3593	4146	4154	4186	4189	4196	4435	4441	4445	4447	4449
	4454	4459	4460	5558	5563	5570	5571	5598	5600	5601	5602	5605
	5613	5616	5618	5645	5718	5724	6351	6417	6426	6427	7350	7360
	7361	7362	7362	7492	7682	7686	7692	7694	7953			
T1	673 #	2191	2202	2219	2248	2266	2517	3580	4395	4406	4413	4427
	4429	4446	4471	4553	5212	5220	5225	5232	5243	5582	5582	5607
	5614	5657	5669	5689	5695	5705	5727	6421	6434	7469	7501	7846
	7954	8750										
UBR	667 #	2161	7113	7123	7123	7135	7138	7140	7140	7141	7141	7945
XWD1	671 #	2153	7951									
	1328 #	3177	3178	3179	3180	3181	3182	3186	3187	3188	3189	3190
	3191	3193	3194	3195	3196	3197	3198	3199	3200	3202	3203	3204
	3205	3206	3207	3208	3209	3211	3212	3213	3214	3215	3216	3217
	3218	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230
	3231	3232	3233	3234	3235	3237	3238	3239	3240	3241	3242	3243
	3244	3350	3351	3352	3353	3354	3355	3356	3357	3359	3360	3361
	3362	3363	3364	3365	3366	3376	3377	3378	3379	3380	3381	3382
	3383	3393	3394	3395	3396	3397	3398	3399	3400	3409	3410	3411
	3412	3413	3414	3415	3416	3462	3463	3464	3465	3466	3467	3468
	3469	3476	3477	3478	3479	3480	3481	3482	3483	3490	3491	3492
	3493	3494	3495	3496	3497	3504	3505	3669	3670	3671	3766	3852
	3853	3854	3855	3856	3857	3858	3859	3909	3910	3911	3912	3913
	3914	5849	5850	5851	5852	5853	5854	5855	5857	5858	5859	5860
	5862	5863	5864	5865	5870	5871	5872	5873	5874	7525	7526	7527
	7528	7536	7537	7538	7539	7550	7551	7552	7553	7737	7738	7739
	7741	7742	7744	7745	7747	7748	7749	7750	7751	7752	7753	7754
	7756	7757	7758	7759	7760	7761	7762	7763	7765	7766	7767	7768
	7769	7770	7771	7772	7774	7775	7776	7777	7778	7779	7780	7781
	7783	7784	7785	7786	7787	7788	7789	7790				
AC	1332 #	2547	2548	2552	2553	2557	2558	2562	2563	2584	2600	2601
	2605	2606	2610	2611	2615	2616	2620	2621	2625	2626	2630	2631
	2635	2636	2642	2643	2647	2648	2652	2653	2657	2658	2662	2663
	2667	2668	2672	2673	2677	2678	2767	2797	2798	2807	2808	2817
	2818	2827	2828	2832	2833	2849	2850	2859	2860	2869	2870	2879

(D) B

	2880	2889	2890	2899	2900	2910	2911	2920	2921	2930	2931	2940
	2941	4050	4051	4064	4065	4112	4113	4682	8188			
BOTH	1334 #	2800	2810	2820	2830	2835	2852	2862	2872	2882	2892	2902
	2913	2923	2933	2943	4053	4067	4115					
DBLAC	1330 #	2766	4079	4080	4137	4138	4163	4304	4305	4309	4310	4381
	5623	5684										
DBLB	1331 #	4140	4307	4312								
MEM	1333 #	2549	2554	2559	2564	2602	2607	2612	2617	2622	2627	2632
	2637	2644	2649	2654	2659	2664	2669	2674	2679	2799	2809	2819
	2829	2834	2844	2845	2851	2861	2871	2881	2891	2901	2912	2922
	2932	2942	4052	4066	4114	4139	4306	4311	7204			
SELF	1329 #	2550	2555	2560	2565	2603	2608	2613	2618	2623	2628	2633
	2638	2645	2650	2655	2660	2665	2670	2675	2680			
(U) BWRITE	1158 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804
	2814	2824	2856	2866	2886	2896	2907	2917	2947	4058	4072	4124
	4130	4153	4155	5444	5448	5494	5516	8518	8818			
(U) BYTE	817 #											
BYTE1	818 #	4692	4697	4711	4732	4885	6033	6043	6525	6671	6843	6846
	6886	6888	6959									
BYTE2	819 #	4886										
BYTE3	820 #	4887										
BYTE4	821 #	4888										
BYTE5	822 #	3041	4791	4792	4793	4794	4795	4889	6655			
(U) CALL	969 #	2156	2182	2194	2399	2771	2781	3551	3590	3592	3593	3607
	3631	3634	3829	3832	3946	3962	3967	3971	3977	3980	4121	4149
	4176	4185	4195	4205	4228	4338	4359	4414	4420	4434	4438	4444
	4490	4514	4692	4697	4701	4706	4711	4720	4723	4971	4981	5015
	5042	5097	5113	5115	5202	5214	5216	5234	5236	5323	5365	5368
	5373	5375	5483	5485	5487	5503	5504	5563	5571	5599	5599	5639
	5644	5656	5663	5670	5689	5713	5717	5721	5746	5749	5752	5784
	5786	5788	5794	5914	5954	5957	5961	5993	5999	6024	6048	6059
	6074	6078	6085	6095	6097	6103	6117	6126	6147	6162	6165	6178
	6193	6204	6212	6215	6218	6221	6229	6234	6261	6296	6305	6321
	6333	6360	6371	6379	6396	6422	6451	6454	6459	6534	6542	6552
	6555	6565	6578	6593	6600	6621	6625	6693	6727	6751	6851	6870
	6875	6927	6935	6946	7188	7257	7281	7284	7303	7350	7380	7393
	7442	7472	7474	7490	7494	7532	7543	7558	7577	7586	7615	7622
	7647	7840	7842	7848	7864	7912	7914	7931	7934	7936	7938	7941
	7944	7947	7950	7953	7955	8314	8321	8332	8339	8350	8359	8377
	8380	8393	8416	8426	8453	8484	8569	8727	8738	8798	8826	
(U) CHKL	722 #	2269	2339	2353	2362	2366	2383	2394	2405	2411	2424	2495
	2527	2533	2538	2541	2591	2787	3087	3406	3549	3573	3588	3613
	3617	3678	3695	3701	3723	3740	3746	3759	3817	3845	3967	3969
	3975	4002	4041	4119	4212	4214	4317	4327	4411	4433	4440	4737
	4765	4771	4892	5157	5206	5912	5931	6351	6541	6591	6594	6622
	6898	6913	6983	7105	7222	7226	7230	7234	7238	7278	7308	7310
	7317	7371	7455	7497	7621	7668	7713	7716	7718	7728	7730	7827
	7846	7866	7931	7932	7934	7935	7938	7939	7941	7942	7944	7945
	7947	7948	7950	7951	7953	7954	7956	8310	8392	8628	8797	8825
	8827											
(U) CHKR	729 #	2269	2339	2353	2362	2366	2383	2394	2405	2411	2424	2495
	2527	2533	2538	2541	2591	2787	3087	3406	3549	3573	3588	3613
	3617	3678	3695	3701	3723	3740	3746	3759	3817	3845	3967	3969
	3975	4002	4041	4119	4212	4214	4317	4327	4411	4433	4440	4737
	4765	4771	4892	5157	5206	5912	5931	6351	6541	6591	6594	6622

	6898	6913	6983	7105	7222	7226	7230	7234	7238	7278	7308	7310
	7317	7371	7455	7497	7621	7668	7713	7716	7718	7728	7730	7827
	7846	7866	7931	7932	7934	7935	7938	7939	7941	7942	7944	7945
	7947	7948	7950	7951	7953	7954	7956	8310	8392	8628	8797	8825
	8827											
(U) CLKL	718 #	2229	2231	2276	2289	2290	2296	2301	2316	2317	2329	2340
	2363	2691	2722	2730	2752	2754	3338	3454	3457	3525	3526	3527
	3536	3539	3552	3564	3594	3618	3632	3760	3819	3831	3835	3979
	3991	3996	4026	4042	4093	4736	4766	4975	5034	5057	5106	5118
	5140	5144	5149	5164	5167	5177	5229	5230	5250	5252	5325	5588
	5925	5928	5931	5972	5998	6001	6017	6023	6052	6071	6077	6085
	6089	6100	6112	6116	6123	6142	6170	6247	6262	6300	6335	6352
	6379	6390	6394	6450	6456	6503	6560	6564	6599	6624	6665	6670
	6728	6847	6889	6898	7052	7078	7092	7139	7153	7155	7199	7279
	7284	7309	7343	7345	7347	7349	7361	7421	7422	7423	7424	7425
	7426	7427	7449	7519	7645	7666	7666	7847	7863	7933	7940	7943
	7946	7949	7952	8201	8462	8508	8516	8759	8765	8771	8773	8775
	8777	8779	8781	8783	8784	8785	8815	8834				
(U) CLKR	725 #	2420	2422	2435	2457	2458	2689	2724	2728	2745	2747	2771
	3114	3116	3387	3529	3538	3541	3543	3544	3545	3578	3583	3586
	3606	3627	3640	3771	3919	3921	3923	3925	3927	3929	3931	3933
	3941	3961	3971	4023	4039	4104	4105	4106	4128	4151	4358	4397
	4459	4513	4636	4813	4948	4956	5019	5064	5099	5103	5290	5291
	5294	5295	5319	5320	5352	5353	5413	5414	5490	5506	5609	5610
	5617	5633	5755	5790	5820	5821	5836	5918	5940	6167	6172	6241
	6257	6258	6264	6303	6305	6312	6318	6327	6346	6398	6401	6406
	6431	6437	6508	6602	6604	6674	6678	6732	6774	6778	6782	6786
	6790	6794	6810	6908	6920	6991	7000	7001	7002	7006	7007	7008
	7009	7014	7015	7016	7019	7020	7021	7022	7023	7024	7025	7026
	7046	7086	7089	7110	7115	7125	7130	7137	7142	7183	7190	7218
	7240	7251	7335	7337	7339	7341	7473	7675	7720	7794	7803	7805
	7807	7809	7811	7813	8194	8197	8248	8252	8255	8260	8411	8432
	8469	8473	8507	8511	8514	8525	8595	8700	8705	8710	8716	8730
	8793											
(U) CLRFPD	1087 #	3690	3807	3813	4207	4707	4724	6390				
(D) COND FUNC	1351 #	2549	2550	2554	2555	2559	2560	2564	2565	2602	2603	2607
	2608	2612	2613	2617	2618	2622	2623	2627	2628	2632	2633	2637
	2638	2644	2645	2649	2650	2654	2655	2659	2660	2664	2665	2669
	2670	2674	2675	2679	2680	2799	2800	2809	2810	2819	2820	2829
	2830	2834	2835	2844	2845	2851	2852	2861	2862	2871	2872	2881
	2882	2891	2892	2901	2902	2912	2913	2922	2923	2932	2933	2942
	2943	4052	4053	4066	4067	4114	4115	4139	4140	4306	4307	4311
	4312	5261	5262	5265	5266	5269	5270	5273	5274	5306	5307	5311
	5312	5335	5336	5340	5341	7204						
(U) CRY38	952 #	2577	2947	3370	3420	3501	3527	3536	3539	4071	4088	4096
	4101	4275	4279	4321	4365	4398	4421	4422	4428	4449	4475	4516
	4517	4518	4534	4536	4543	4549	4580	4610	4615	4620	4625	4637
	4641	4645	4647	4976	5033	5050	5100	5119	5137	5161	5249	5280
	5355	5358	5359	5366	5377	5390	5482	5484	5486	5488	5499	5515
	5544	5583	5696	5707	5714	5730	5771	5776	5797	5803	5837	5963
	5996	6010	6031	6098	6134	6135	6136	6183	6247	6265	6268	6336
	6353	6361	6380	6388	6389	6423	6425	6428	6476	6818	6939	6947
	6953	7261	7361	7385	7398	7645	8759	8819				
(U) DBM	697 #											
APR FLAGS	700 #	7003	7004	7061	7093	7267						

Produced on Advanced Information Services Electronic Laser Printer. PLO/RES. DTN: 222-7881

BYTES	701 #											
DP	703 #	3041	4692	4697	4711	4732	4791	4792	4793	4794	4795	4885
	4886	4887	4888	4889	6033	6043	6525	6655	6671	6843	6846	6886
	6888	6959										
DP SWAP	704 #	2389	2569	2716	2718	2721	2724	2727	2730	3269	3274	3593
	3770	3827	3841	3959	4811	4937	4953	5053	5093	5102	5175	6400
	6457	6462	6463	6757	7085	7337	7350	7360	8402			
EXP	702 #	5490	5506	5836	7292	7293	7294					
MEM	706 #	2161	2268	2339	2352	2361	2365	2393	2404	2410	2423	3548
	3572	3587	3609	3613	3617	3677	3722	3759	3844	4001	4764	4770
	5156	5205	5912	5931	6351	6594	6622	6898	6912	6968	6976	6983
	7104	7222	7226	7230	7234	7238	7277	7317	7439	7443	7455	7491
	7497	7587	7668	7713	7716	7718	7826	7841	7865	7914	8245	8309
	8391	8627	8740	8746								
	699 #	8215										
PF DISP	698 #											
SCAD DIAG	705 #	7682	7925	8195	8210	8726						
VMA	707 #	2149	2151	2153	2155	2158	2178	2179	2188	2191	2202	2219
#	2238	2242	2245	2248	2257	2260	2263	2266	2420	2422	2435	2457
	2458	2517	2745	2747	2752	2754	2771	3042	3113	3115	3529	3538
	3541	3543	3544	3545	3578	3580	3583	3586	3592	3606	3607	3621
	3627	3640	3725	3754	3818	3833	3919	3921	3923	3925	3927	3929
	3931	3933	3939	3943	3983	3987	3989	3992	3994	4022	4031	4037
	4105	4106	4358	4395	4396	4406	4413	4459	4511	4538	4544	4636
	4801	4943	4946	4950	4969	4973	4979	4983	5012	5017	5050	5055
	5062	5106	5116	5212	5217	5225	5232	5237	5290	5291	5294	5295
	5319	5320	5324	5352	5353	5389	5393	5413	5414	5508	5551	5553
	5609	5610	5617	5633	5653	5654	5676	5676	5704	5706	5744	5747
	5750	5753	5813	5820	5821	5880	5882	5884	5886	5888	5890	5892
	5894	5896	5913	5915	5940	5956	5959	5972	5998	6001	6023	6071
	6077	6085	6095	6097	6116	6123	6142	6155	6159	6166	6170	6171
	6179	6183	6199	6241	6257	6258	6264	6270	6289	6302	6305	6311
	6316	6324	6326	6352	6379	6383	6397	6399	6405	6417	6426	6431
	6437	6453	6465	6468	6476	6488	6491	6503	6507	6522	6560	6564
	6580	6592	6597	6601	6603	6618	6624	6638	6658	6664	6672	6675
	6752	6773	6777	6781	6785	6799	6793	6803	6809	6908	6918	6985
	6991	7000	7001	7002	7006	7007	7008	7009	7014	7015	7016	7019
	7020	7021	7022	7023	7024	7025	7026	7028	7031	7036	7038	7051
	7053	7055	7057	7059	7062	7064	7066	7077	7087	7090	7094	7108
	7113	7116	7120	7135	7153	7154	7155	7181	7189	7198	7218	7240
	7249	7268	7282	7332	7334	7335	7336	7338	7340	7342	7343	7344
	7345	7346	7348	7379	7381	7392	7394	7421	7422	7423	7424	7425
	7426	7427	7428	7429	7430	7431	7432	7433	7434	7436	7440	7448
	7457	7458	7469	7473	7487	7493	7499	7501	7515	7579	7582	7590
	7597	7603	7617	7618	7629	7635	7649	7651	7665	7708	7794	7803
	7805	7807	7809	7811	7813	7861	7862	8192	8196	8200	8201	8225
	8227	8241	8246	8249	8250	8253	8259	8267	8296	8304	8315	8322
	8325	8340	8345	8346	8394	8406	8409	8413	8417	8422	8424	8431
	8433	8434	8437	8439	8459	8460	8464	8467	8471	8492	8505	8509
	8512	8513	8515	8521	8528	8530	8531	8532	8533	8536	8571	8576
	8592	8598	8601	8603	8606	8609	8630	8640	8694	8696	8698	8699
	8701	8703	8706	8708	8712	8715	8720	8728	8750	8753	8771	8773
	8775	8777	8779	8781	8785	8790	8793					
	688 #											
(U) DBUS	694 #	2149	2151	2153	2155	2158	2161	2178	2179	2188	2191	2202
DBM												

Produced on Advanced Information Services Electronic Laser Printer, PK01E85, DTN: 233-7981

2219	2238	2242	2245	2248	2257	2260	2263	2266	2268	2269	2339
2339	2352	2353	2361	2362	2365	2366	2389	2393	2394	2404	2405
2410	2411	2420	2422	2423	2424	2435	2457	2458	2517	2569	2716
2718	2721	2724	2727	2730	2745	2747	2752	2754	2771	3041	3042
3113	3115	3269	3274	3529	3538	3541	3543	3544	3545	3548	3549
3572	3573	3578	3580	3583	3586	3587	3588	3592	3593	3606	3607
3609	3613	3613	3617	3617	3621	3627	3640	3677	3678	3722	3723
3725	3754	3759	3759	3770	3818	3827	3833	3841	3844	3845	3919
3921	3923	3925	3927	3929	3931	3933	3939	3943	3959	3983	3987
3989	3992	3994	4001	4002	4022	4031	4037	4105	4106	4358	4395
4396	4406	4413	4459	4511	4538	4544	4636	4692	4697	4711	4732
4764	4765	4770	4771	4791	4792	4793	4794	4795	4801	4811	4885
4886	4887	4888	4889	4937	4943	4946	4950	4953	4969	4973	4979
4983	5012	5017	5050	5053	5055	5062	5093	5102	5106	5116	5156
5157	5175	5205	5206	5211	5217	5225	5231	5237	5290	5291	5294
5295	5319	5320	5324	5352	5353	5389	5393	5413	5414	5490	5506
5508	5551	5553	5609	5610	5617	5633	5653	5654	5673	5676	5704
5706	5744	5747	5750	5753	5813	5820	5821	5836	5880	5882	5884
5886	5888	5890	5892	5894	5896	5912	5912	5913	5915	5931	5931
5940	5956	5959	5972	5998	6001	6023	6033	6043	6071	6077	6085
6095	6097	6116	6123	6142	6155	6159	6166	6170	6171	6179	6183
6199	6241	6257	6258	6264	6270	6289	6302	6305	6311	6316	6324
6326	6351	6351	6352	6379	6383	6397	6399	6400	6405	6417	6426
6431	6437	6453	6457	6462	6465	6465	6468	6476	6488	6491	6503
6507	6522	6525	6560	6564	6580	6592	6594	6594	6597	6601	6603
6618	6622	6622	6624	6638	6655	6658	6664	6671	6672	6675	6752
6757	6773	6777	6781	6785	6789	6793	6803	6809	6843	6846	6886
6888	6898	6898	6908	6912	6913	6918	6959	6968	6976	6983	6983
6985	6991	7000	7001	7002	7003	7004	7006	7007	7008	7009	7014
7015	7016	7019	7020	7021	7022	7023	7024	7025	7026	7028	7031
7036	7038	7051	7053	7055	7057	7059	7061	7062	7064	7066	7077
7085	7087	7090	7093	7094	7104	7105	7108	7113	7116	7120	7135
7153	7154	7155	7181	7189	7198	7218	7222	7222	7226	7226	7230
7230	7234	7234	7238	7238	7240	7249	7267	7268	7277	7278	7282
7292	7293	7294	7317	7317	7332	7334	7335	7336	7337	7338	7340
7342	7343	7344	7345	7346	7348	7350	7360	7379	7381	7392	7394
7421	7422	7423	7424	7425	7426	7427	7428	7429	7430	7431	7432
7433	7434	7436	7439	7440	7443	7444	7448	7455	7455	7457	7458
7469	7473	7487	7491	7492	7493	7497	7497	7499	7501	7515	7579
7582	7587	7588	7590	7597	7603	7617	7618	7629	7635	7649	7651
7665	7668	7668	7682	7708	7713	7713	7716	7716	7718	7718	7794
7803	7805	7807	7809	7811	7813	7826	7827	7841	7861	7862	7865
7866	7914	7925	8192	8195	8196	8200	8201	8210	8215	8221	8225
8227	8241	8245	8246	8249	8250	8253	8259	8267	8296	8304	8309
8310	8315	8322	8325	8340	8345	8346	8391	8392	8394	8402	8406
8409	8413	8417	8422	8424	8431	8433	8434	8437	8439	8459	8460
8464	8467	8471	8492	8505	8509	8512	8513	8515	8521	8528	8530
8531	8532	8533	8536	8571	8576	8592	8598	8601	8603	8606	8609
8627	8628	8630	8640	8694	8696	8698	8699	8701	8703	8706	8708
8712	8715	8720	8726	8728	8740	8746	8750	8753	8771	8773	8775
8777	8779	8781	8785	8790	8793						
692 #	2173	2174	2489	2495	2521	2527	2533	2538	2541	2591	2772
2787	3022	3043	3078	3111	3117	3300	3387	3389	3406	3429	3432
3432	3435	3435	3438	3438	3441	3444	3444	3447	3447	3450	3450
3654	3695	3701	3710	3712	3740	3746	3775	3780	3808	3817	3967

DP

Produced on Advanced Information Services Electronic Laser Printer, PKCJ156, DTN: 223-7881

3969	3975	4041	4104	4128	4151	4179	4187	4196	4198	4209	4209	
4210	4211	4213	4215	4437	4453	4454	4455	4457	4460	4479	4481	
4484	4611	4616	4621	4627	4628	4629	4707	4737	4892	5065	5114	
5134	5160	5178	5203	5248	5440	5446	5758	5766	5802	5803	5804	
5811	5963	6009	6029	6036	6046	6049	6084	6086	6088	6122	6129	
6163	6205	6222	6224	6259	6260	6260	6272	6273	6280	6282	6306	
6306	6309	6341	6347	6358	6363	6385	6386	6387	6545	6546	6591	
6606	6679	6695	6732	6821	6821	6849	6890	6928	6936	6943	6948	
6955	7251	7308	7310	7371	7544	7621	7663	7728	7730	7732	7846	
7931	7932	7934	7935	7938	7939	7941	7942	7944	7945	7947	7948	
7950	7951	7953	7954	7956	8202	8761	8797	8825	8827			
PC FLAGS	689 #	3628	3642	3689	3806	3812	3960	3986	4216	6964	7459	7471
	8805											
PI NEW	690 #	7418										
RAM	693 #	2181	2289	2301	2316	2329	2357	2370	2371	2383	2384	2588
	2689	2691	2722	2728	2781	2814	2824	2856	2866	2886	2896	2906
	2937	2965	2968	2982	2988	3000	3003	3016	3087	3090	3276	3291
	3294	3299	3370	3473	3487	3501	3509	3556	3564	3679	3717	3750
	3772	3828	3840	4057	4071	4084	4087	4096	4098	4101	4119	4147
	4175	4177	4189	4194	4212	4214	4317	4326	4327	4409	4411	4433
	4440	4450	4610	4614	4615	4619	4620	4624	4625	4713	4717	4749
	4757	4967	4976	4977	4989	4993	5028	5035	5040	5048	5104	5113
	5176	5201	5283	5316	5348	5412	5547	5548	5558	5638	5650	5693
	5701	5925	5928	5955	5958	5973	5976	5992	5996	6006	6013	6022
	6025	6041	6045	6060	6061	6062	6063	6080	6094	6096	6114	6127
	6133	6140	6146	6158	6161	6178	6181	6185	6189	6192	6194	6201
	6203	6211	6213	6216	6252	6254	6255	6267	6268	6287	6292	6293
	6294	6315	6320	6332	6338	6348	6356	6361	6367	6374	6380	6381
	6382	6403	6423	6428	6433	6435	6452	6510	6541	6551	6554	6579
	6588	6590	6615	6630	6692	6694	6737	6738	6745	6807	6812	6814
	6819	6823	6825	6840	6842	6869	6893	6895	6926	6930	6934	6939
	6940	6945	6953	6954	6957	7050	7084	7152	7209	7211	7213	7215
	7217	7246	7255	7260	7266	7298	7299	7302	7323	7533	7547	7561
	7563	7655	7658	7660	7666	7672	7674	7834	7913	7918	7922	7937
	7955	7957	7958	7960	8199	8288	8291	8318	8320	8351	8358	8375
	8378	8397	8442	8452	8454	8483	8499	8539	8540	8541	8547	8567
	8568	8574	8736	8742	8751	8755	8767					
(U) DEST	623 #											
A	624 #	2389	2405	2489	2521	2569	2716	2718	2721	2724	2727	2730
	2772	3043	3078	3269	3274	3300	3387	3389	3429	3432	3435	3438
	3441	3444	3447	3450	3593	3710	3712	3770	3808	3827	3841	3959
	4104	4128	4151	4209	4454	4460	4629	4692	4697	4707	4711	4732
	4791	4792	4793	4794	4795	4811	4885	4886	4887	4888	4889	4937
	4953	5053	5093	5102	5114	5175	5178	5203	5440	5446	5490	5506
	5766	5811	5836	6009	6029	6033	6043	6046	6086	6122	6129	6163
	6222	6224	6259	6260	6272	6306	6309	6341	6347	6309	6358	6363
	6385	6386	6387	6400	6457	6462	6463	6525	6546	6606	6671	6679
	6732	6757	6821	6843	6846	6849	6886	6888	6890	6928	6936	6943
	6948	6955	6959	7085	7251	7337	7350	7360	7544	7732	8202	8402
AD	626 #	2149	2151	2153	2155	2158	2161	2163	2166	2175	2178	2179
	2180	2181	2184	2191	2194	2202	2216	2219	2229	2231	2238	2242
	2245	2248	2257	2260	2263	2266	2269	2276	2289	2295	2301	2316
	2320	2329	2339	2353	2357	2362	2363	2370	2371	2374	2394	2400
	2411	2414	2417	2420	2422	2435	2457	2458	2517	2577	2588	2689
	2691	2722	2728	2738	2740	2745	2747	2752	2754	2759	2761	2771

Produced on Advanced Information Services Electronic Laser Printer. PRC(JES), DTN: 233-7881

Cross Reference Listing

2781	2804	2814	2824	2839	2856	2866	2876	2886	2896	2906	2907
2917	2927	2937	2947	2965	2968	2982	3016	3019	3042	3271	3276
3288	3291	3294	3299	3338	3370	3404	3420	3454	3457	3473	3487
3501	3509	3525	3526	3529	3538	3541	3543	3544	3545	3549	3552
3564	3573	3578	3579	3580	3583	3586	3588	3594	3606	3613	3614
3617	3621	3627	3628	3632	3635	3640	3642	3653	3678	3679	3689
3707	3717	3723	3725	3750	3754	3759	3772	3806	3812	3818	3825
3828	3830	3833	3840	3845	3919	3921	3923	3925	3927	3929	3931
3933	3939	3943	3945	3960	3970	3979	3983	3985	3986	3989	3994
4002	4022	4026	4031	4037	4042	4057	4071	4084	4087	4091	4092
4096	4098	4101	4105	4106	4118	4124	4130	4132	4133	4146	4147
4152	4155	4156	4170	4186	4200	4216	4223	4316	4319	4321	4325
4326	4340	4346	4352	4358	4375	4395	4406	4407	4413	4424	4430
4439	4447	4459	4466	4469	4518	4533	4535	4536	4538	4543	4544
4557	4624	4625	4636	4637	4639	4641	4645	4646	4647	4717	4730
4736	4765	4771	4813	4824	4909	4910	4911	4912	4946	4955	4967
4969	4977	4979	4983	4986	5012	5025	5035	5038	5044	5055	5062
5098	5101	5104	5106	5113	5116	5140	5142	5144	5149	5164	5166
5167	5176	5201	5219	5225	5250	5251	5252	5280	5283	5286	5287
5288	5290	5291	5294	5295	5300	5316	5319	5320	5326	5346	5348
5352	5353	5354	5355	5390	5412	5413	5414	5444	5448	5449	5497
5499	5509	5512	5514	5515	5516	5527	5544	5545	5558	5563	5564
5571	5572	5582	5607	5609	5610	5614	5617	5634	5645	5657	5658
5687	5689	5690	5695	5696	5697	5705	5716	5718	5732	5733	5756
5772	5774	5776	5791	5793	5795	5797	5800	5820	5821	5837	5880
5882	5884	5886	5888	5890	5892	5894	5896	5912	5915	5917	5925
5926	5928	5931	5940	5951	5955	5958	5959	5965	5967	5972	5973
5974	5976	5984	5992	5994	5996	5998	6001	6006	6010	6013	6015
6016	6017	6022	6023	6025	6031	6040	6041	6045	6047	6052	6060
6061	6062	6063	6070	6071	6075	6077	6080	6085	6089	6094	6096
6099	6100	6101	6112	6114	6116	6123	6127	6131	6133	6134	6135
6136	6141	6142	6146	6155	6156	6159	6161	6166	6170	6171	6178
6181	6185	6189	6192	6194	6203	6211	6213	6216	6219	6230	6232
6241	6242	6246	6247	6252	6254	6255	6257	6258	6262	6264	6265
6267	6268	6270	6287	6289	6292	6293	6296	6300	6302	6305	6310
6311	6315	6316	6320	6322	6326	6329	6332	6334	6335	6336	6338
6346	6348	6349	6351	6352	6354	6356	6361	6367	6370	6374	6379
6380	6381	6382	6384	6388	6389	6390	6394	6398	6401	6403	6405
6417	6421	6423	6425	6426	6428	6431	6433	6435	6437	6439	6450
6452	6456	6465	6468	6502	6503	6507	6510	6519	6521	6526	6551
6554	6560	6564	6579	6580	6588	6590	6592	6594	6597	6601	6603
6615	6618	6620	6622	6624	6630	6638	6658	6664	6669	6672	6675
6677	6692	6694	6728	6737	6745	6754	6773	6777	6781	6785	6789
6793	6803	6807	6809	6812	6814	6818	6819	6823	6825	6840	6842
6845	6847	6868	6869	6887	6889	6898	6908	6913	6918	6926	6930
6934	6939	6940	6944	6945	6947	6951	6953	6954	6957	6964	6967
6975	6983	6991	7000	7001	7002	7003	7004	7006	7007	7008	7009
7010	7011	7014	7015	7016	7019	7020	7021	7022	7023	7024	7025
7026	7028	7031	7036	7038	7050	7051	7053	7054	7055	7058	7060
7061	7063	7065	7077	7084	7087	7090	7093	7094	7096	7105	7108
7113	7120	7123	7128	7135	7138	7140	7141	7152	7153	7155	7174
7181	7182	7187	7189	7194	7199	7199	7209	7211	7213	7215	7217
7218	7222	7226	7230	7234	7238	7240	7246	7249	7250	7255	7256
7258	7260	7261	7266	7267	7268	7278	7279	7282	7292	7293	7294
7297	7298	7299	7302	7309	7317	7323	7329	7333	7335	7336	7339

Produced on Advanced Information Services Electronic Laser Printer, PKO IES6, DTN: 222-7881

7341	7343	7345	7347	7349	7361	7362	7378	7379	7381	7385	7392	
7394	7398	7421	7422	7423	7424	7425	7426	7427	7428	7429	7430	
7431	7432	7433	7434	7436	7440	7444	7446	7448	7450	7455	7459	
7469	7471	7473	7475	7485	7487	7489	7492	7493	7495	7497	7501	
7503	7515	7519	7533	7547	7557	7559	7561	7563	7566	7579	7582	
7588	7591	7597	7603	7618	7629	7635	7648	7655	7658	7660	7666	
7668	7672	7674	7675	7682	7708	7713	7716	7720	7722	7794	7803	
7805	7807	7809	7811	7813	7827	7834	7844	7847	7861	7862	7866	
7913	7918	7922	7925	7933	7937	7940	7943	7946	7949	7952	7955	
7957	7958	7960	8192	8195	8196	8199	8201	8210	8221	8225	8227	
8241	8246	8250	8253	8259	8261	8267	8288	8291	8296	8304	8310	
8318	8320	8346	8349	8351	8375	8378	8392	8397	8406	8409	8424	
8431	8434	8442	8452	8454	8460	8467	8471	8496	8499	8505	8508	
8509	8513	8515	8517	8520	8521	8528	8530	8532	8536	8539	8540	
8541	8547	8568	8570	8574	8592	8594	8628	8639	8696	8699	8703	
8708	8715	8720	8726	8728	8731	8736	8742	8750	8751	8755	8759	
8765	8767	8769	8771	8773	8775	8777	8779	8781	8783	8784	8785	
8790	8793	8805	8815	8817	8819	8830	8834					
AD*.5	633 #	2152	2424	2426	2427	2984	2988	3000	3003	3005	3006	3010
	3041	3093	3096	4168	4169	4175	4177	4187	4189	4194	4225	4387
	4409	4415	4546	4547	4548	4554	4804	4815	4891	4940	4943	4950
	4959	5361	5369	5371	5376	5377	5436	5485	5487	5503	5504	5523
	5524	5529	5547	5548	5551	5553	5638	5639	5646	5650	5653	5654
	5672	5693	5701	5704	5706	5830	5841	6464	6467	6471	6473	6553
	6655	6661	6668	6731	7178	7180	7188	7195	7197	7499	7502	7520
	7595	7601	8264									
AD*2	631 #	2150	2972	2995	3008	3011	3038	4123	4150	4206	4449	4589
	4713	5050	5059	5212	5232	5243	5356	5359	5364	5439	6201	6202
	6458	6642	6748	7132	7147	7149	7447	7483	7627	7633	7724	8326
	8329	8363	8502									
PASS	628 #	2160	2168	2170	2171	2174	2176	2177	2183	2185	2228	2232
	2250	2307	2334	2343	2430	2495	2527	2533	2538	2541	2591	2787
	2788	3111	3326	3406	3453	3458	3532	3568	3630	3641	3663	3695
	3701	3730	3735	3740	3746	3775	3780	3814	3817	3842	3967	3969
	3975	4041	4126	4154	4196	4211	4213	4215	4218	4393	4437	4457
	4479	4481	4484	4611	4616	4621	4627	4628	4690	4693	4702	4737
	4753	4761	4892	4949	4962	4972	4982	5014	5094	5122	5130	5246
	5758	5802	5803	5804	5922	5929	5935	5938	5963	5968	5985	6030
	6056	6057	6058	6073	6104	6120	6143	6156	6174	6205	6295	6297
	6353	6359	6372	6378	6455	6511	6577	6589	6591	6631	6695	6725
	6894	6896	6914	6931	6965	7029	7032	7072	7102	7157	7223	7227
	7231	7235	7239	7253	7264	7285	7288	7295	7308	7310	7318	7319
	7352	7370	7371	7376	7403	7404	7454	7518	7585	7620	7621	7623
	7652	7667	7686	7692	7694	7726	7728	7730	7823	7831	7839	7846
	7858	7911	7926	7931	7932	7934	7935	7936	7938	7939	7941	7942
	7944	7945	7947	7948	7950	7951	7953	7954	7956	7959	8198	8208
	8209	8211	8212	8216	8219	8229	8231	8233	8235	8239	8240	8293
	8301	8306	8381	8383	8386	8404	8526	8542	8623	8725	8794	8797
	8825	8827										
Q_AD	627 #	2383	2981	3087	3113	3115	4119	4145	4184	4204	4212	4214
	4224	4317	4327	4386	4396	4398	4411	4433	4436	4438	4440	4451
	4464	4465	4468	4475	4483	4511	4516	4517	4534	4559	4610	4614
	4615	4619	4620	4819	4895	4902	4964	4973	4976	4993	5017	5033
	5040	5100	5157	5206	5217	5222	5237	5240	5297	5321	5324	5368
	5379	5391	5395	5397	5398	5411	5424	5482	5484	5486	5488	5556

	5565	5568	5573	5583	5588	5628	5633	5643	5656	5662	5674	5677
	5707	5712	5717	5719	5730	5754	5770	5771	5789	5981	6140	6541
	7718											
Q_Q*.5	632 #	2384	2385	3071	3090	3099	3143	4120	4148	4227	4248	4249
	4261	4268	4276	4286	4294	4389	4391	4399	4403	4419	4434	4441
	4493	4541	5041	5208	5228	5229	5230	5244	5299	5322	5327	5402
	5404	5434	5577	5602	5605	5613	5616	5618	5632	5667	5708	5722
	5748	5751	5785	5787	5833							
Q_Q*2	630 #	2991	3072	3075	3076	3077	3102	3106	3109	3144	3147	3148
	4272	4280	4290	4298	4435	4445	4473	4476	4549	4552	4553	4556
	4558	4580	4583	4595	4600	4820	4823	4896	4899	4904	5223	5224
	5481	5493	5501	5505	5579	5584	5586	5598	5600	5601	5630	5671
	5675	5724	5741	5763	5780	5808	5983					
(U) DISP	862 #	4561										
ADISP	869 #											
AREAD	865 #	2321										
BDISP	870 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804
	2814	2824	2856	2866	2886	2896	2907	2917	2947	3271	3277	3370
	3389	3390	3473	3487	3501	3511	3696	3703	3741	4058	4072	4124
	4130	4153	4155	5207	5436	5444	5448	5494	5506	5516	5523	5927
	5969	6113	6168	6375	6729	6801	6808	7533	7560	7578	7589	7616
	8518	8818										
BYTE	875 #	4705	4722	6150	6853	6874						
CONSOLE	863 #											
DP	868 #	4427	4429	4446	4471	6474	6501	8761				
DP LEFT	866 #	6460	6755	7419	8217							
DROM	864 #	2354	2358	2367	2372	2380	2401	2407	2431	2436	5936	5939
EAMODE	876 #	2279	2345	3554	4744	4747	5923	6891	6899	7657		
MUL	872 #	4268	4276	4286	4294	5580	5606	5608	5725			
NICOND	874 #	2228	2229	2231	2232	2292	2298	2522	3043	3089	3117	3326
	3338	3429	3453	3454	3457	3458	3525	3526	3552	3594	3711	3787
	3794	4218	4482	4485	4693	5440	5446	5805	7352	7623	8202	
NORM	867 #	5300	5329	5406	5481	5483	5493	5497	5499	5501	5502	5505
	5527	5678	5732	5733	5743	5745	5765	5775	5777	5782	5783	5810
	5839											
PAGE FAIL	873 #											
RETURN	871 #	2457	2458	3559	3563	4024	4027	4230	4273	4281	4291	4299
	4493	4515	4516	4518	4602	4629	4731	4733	4737	4772	4805	4824
	4892	5106	5243	5244	5526	5528	5529	5618	5813	5820	5821	5836
	6225	6235	6240	6243	6281	6282	6432	6438	6609	6632	6697	6728
	6740	6771	6787	6804	6813	6815	6841	6876	6909	6915	6920	6960
	7199	7265	7289	7364	7408	7516	7519	7591	7598	7604	7661	7669
	7673	7676	7702	7714	7716	7718	7720	7722	7724	7726	7728	7730
	7732	7923	7927	7961	8548	8570	8575	8581	8612	8615	8616	8629
	8638	8641	8642	8746	8834							
SCADO	877 #	4692	4697	4711	4787	5283	5426	5428	5981	5983	6034	6844
	6886	7688										
(U) DIVIDE	963 #	4552	4553	4597	4601							
(U) DONT CACHE	1133 #											
(U) DP FUNC	1150 #	7585	7620	7726	8527	8543						
(U) DT	932 #											
2T	935 #											
3T	936 #	2181	2371	2383	2384	2494	2526	2532	2537	2540	2590	2781
	2787	2988	3000	3003	3017	3087	3090	3332	3335	3344	3347	3406
	3435	3438	3447	3450	3592	3592	3607	3607	3694	3700	3739	3745

Produced on Advanced Information Services Electronic Laser Printer, PRC7186, DTN: 2237881

	3816	3966	3968	3974	3987	3992	4040	4084	4096	4122	4127	4189
	4194	4212	4214	4327	4329	4331	4335	4349	4388	4394	4409	4411
	4423	4433	4440	4453	4534	4610	4612	4614	4615	4617	4619	4620
	4622	4638	4645	4692	4697	4705	4711	4713	4722	4737	4892	4963
	4967	4976	4977	4992	5022	5035	5040	5104	5133	5159	5176	5248
	5300	5329	5346	5366	5372	5374	5406	5447	5481	5483	5489	5493
	5497	5499	5501	5502	5505	5508	5513	5527	5548	5650	5669	5673
	5676	5678	5690	5701	5720	5732	5733	5743	5744	5745	5747	5750
	5753	5762	5765	5771	5775	5777	5782	5783	5807	5810	5813	5831
	5839	5913	5955	5956	5973	5976	5992	5996	6006	6013	6022	6025
	6041	6045	6060	6061	6062	6063	6080	6094	6095	6097	6114	6127
	6133	6137	6140	6146	6150	6161	6178	6179	6181	6185	6189	6190
	6192	6194	6199	6201	6203	6211	6213	6216	6252	6254	6255	6266
	6267	6268	6269	6287	6293	6320	6324	6332	6337	6338	6348	6356
	6361	6367	6374	6380	6381	6382	6383	6397	6399	6403	6453	6488
	6491	6510	6541	6545	6551	6554	6558	6579	6588	6590	6591	6595
	6615	6616	6630	6692	6694	6737	6738	6739	6745	6752	6807	6812
	6814	6819	6823	6825	6840	6842	6853	6869	6874	6926	6930	6934
	6939	6940	6945	6953	6954	6957	7003	7004	7050	7057	7059	7061
	7062	7064	7066	7084	7093	7152	7209	7211	7213	7215	7217	7246
	7255	7260	7262	7266	7267	7296	7298	7299	7302	7308	7310	7323
	7332	7334	7338	7340	7342	7344	7346	7348	7371	7585	7590	7617
	7620	7621	7649	7651	7655	7658	7660	7665	7672	7674	7726	7728
	7730	7846	7913	7918	7919	7922	7931	7932	7934	7935	7937	7938
	7939	7941	7942	7944	7945	7947	7948	7950	7951	7953	7954	7955
	7956	7957	7958	7960	8199	8200	8249	8288	8289	8291	8292	8315
	8318	8320	8322	8325	8340	8345	8351	8375	8376	8378	8379	8394
	8397	8413	8417	8422	8433	8437	8439	8442	8452	8454	8459	8464
	8492	8499	8512	8531	8533	8539	8540	8541	8547	8568	8571	8574
	8576	8598	8601	8603	8606	8609	8630	8640	8694	8698	8701	8706
	8712	8736	8742	8751	8753	8755	8767	8796	8825	8827		
	937 #	8358	8483	8567								
	938 #											
(U) EXT ADR	1156 #	3965	3979	3999	4026	6972	6980	7454	7496	7585	7620	7726
	7844	7847	7930	7933	7940	7943	7946	7949	7952	8294	8302	8308
(U) FETCH	8388	8625	8795	8831	8834							
	1122 #	2228	2229	2231	2232	2251	2277	2291	2297	2344	2405	2430
	3020	3326	3338	3453	3454	3457	3458	3525	3526	3552	3594	3709
	4218	4693	4702	7352	7623	7859	8753					
(D) FL-B	1339 #											
AC	1340 #	5260	5263	5264	5268	5271	5272	5305	5309	5310	5334	5338
	5339	5384	5385	5419	5420							
	1342 #	5262	5266	5270	5274	5307	5312	5336	5341			
BOTH	1341 #	5261	5265	5269	5273	5306	5311	5335	5340			
MEM	1305 #	8247	8512	8533	8640	8697	8700					
(U) FLG.C	1304 #	3622	8325	8345	8413	8422	8439	8576	8601	8606	8630	8694
(U) FLG.PI	8793											
	1306 #	5508	5609	5610	5617	5744	5747	5750	5753	5813	5820	5821
(U) FLG.SN	1303 #	8247	8433	8472	8531	8640	8697	8709				
(U) FLG.W	958 #	2160	2168	2170	2171	2173	2174	2176	2177	2183	2185	2489
(U) FMWRITE	2521	2772	3022	3043	3078	3111	3117	3300	3389	3429	3432	3435
	3438	3441	3444	3447	3450	3663	3710	3712	3775	3780	3808	4179
	4187	4196	4198	4209	4210	4211	4213	4215	4437	4453	4454	4455
	4457	4460	4479	4481	4484	4611	4616	4621	4627	4628	4629	4690
	4707	4949	4962	4972	4982	5014	5065	5114	5178	5203	5440	5446

Produced on Advanced Information Services Electronic Laser Printer, PKC/155, DTN: 223-7881

	5758	5766	5802	5803	5804	5811	5922	5935	5938	5963	5968	5985
	6009	6029	6030	6036	6046	6049	6056	6057	6058	6073	6084	6086
	6088	6104	6120	6122	6129	6143	6156	6163	6174	6205	6222	6224
	6259	6260	6272	6273	6280	6282	6295	6297	6306	6309	6341	6347
	6353	6358	6359	6363	6372	6378	6385	6386	6387	6546	6577	6606
	6679	6695	6725	6821	6849	6890	6914	6928	6931	6936	6943	6948
	6955	6965	7072	7157	7223	7227	7231	7235	7239	7253	7264	7285
	7288	7318	7319	7403	7404	7544	7652	7686	7692	7694	7732	7839
	7911	7926	7936	8198	8202	8208	8209	8211	8212	8219	8229	8231
	8233	8235	8239	8240	8381	8383	8404	8725				
(U) FORCE EXEC	1120 #	3965	3979	3999	4026	6972	6980	7440	7454	7456	7470	7487
	7496	7504	7579	7582	7618	7629	7635	7844	7847	7861	7930	7933
	7940	7943	7946	7949	7952	8294	8302	8308	8388	8625	8795	8831
	8834											
(U) FORCE USER	1119 #	3965	3979	3999	4026	6972	6980	7440	7454	7487	7496	7579
	7582	7618	7629	7635	7844	7847	7861	7930	7933	7940	7943	7946
	7949	7952	8294	8302	8308	8388	8625	8795	8831	8834		
(U) GENL	720 #	2229	2231	2276	2289	2290	2296	2301	2316	2317	2329	2340
	2363	2691	2722	2730	2752	2754	3338	3454	3457	3525	3526	3527
	3536	3539	3552	3564	3594	3618	3632	3760	3819	3831	3835	3979
	3991	3996	4026	4042	4093	4736	4766	4975	5034	5057	5106	5118
	5140	5144	5149	5164	5167	5177	5229	5230	5250	5252	5325	5588
	5925	5928	5931	5972	5998	6001	6017	6023	6052	6071	6077	6085
	6089	6100	6112	6116	6123	6142	6170	6247	6262	6300	6335	6352
	6379	6390	6394	6450	6456	6503	6560	6564	6599	6624	6665	6670
	6728	6847	6889	6898	7052	7078	7092	7139	7153	7155	7199	7279
	7284	7309	7343	7345	7347	7349	7361	7421	7422	7423	7424	7425
	7426	7427	7449	7519	7645	7666	7666	7847	7863	7933	7940	7943
	7946	7949	7952	8201	8462	8508	8516	8759	8765	8771	8773	8775
	8777	8779	8781	8783	8784	8785	8815	8834				
(U) GENR	727 #	2420	2422	2435	2457	2458	2689	2724	2728	2745	2747	2771
	3114	3116	3387	3529	3538	3541	3543	3544	3545	3578	3583	3586
	3606	3627	3640	3771	3919	3921	3923	3925	3927	3929	3931	3933
	3941	3961	3971	4023	4039	4104	4105	4106	4128	4151	4358	4397
	4459	4513	4636	4813	4948	4956	5019	5064	5099	5103	5290	5291
	5294	5295	5319	5320	5352	5353	5413	5414	5490	5506	5609	5610
	5617	5633	5755	5790	5820	5821	5836	5918	5940	6167	6172	6241
	6257	6258	6264	6303	6305	6312	6318	6327	6346	6398	6401	6406
	6431	6437	6508	6602	6604	6674	6678	6732	6774	6778	6782	6786
	6790	6794	6810	6908	6920	6991	7000	7001	7002	7006	7007	7008
	7009	7014	7015	7016	7019	7020	7021	7022	7023	7024	7025	7026
	7046	7086	7089	7110	7115	7125	7130	7137	7142	7183	7190	7218
	7240	7251	7335	7337	7339	7341	7473	7675	7720	7794	7803	7805
	7807	7809	7811	7813	8194	8197	8248	8252	8255	8260	8411	8432
	8469	8473	8507	8511	8514	8525	8595	8700	8705	8710	8716	8730
	8793											
(U) HALT	1282 #											
BW14	1292 #	2517										
CSL	1286 #	2248	2266									
HALT	1285 #	3580										
ILLII	1289 #	7469										
ILLINT	1290 #	7501										
IOPF	1288 #	8750										
MULERR	1294 #	2191										
NICOND 5	1293 #											

	POWER	1284 #	2219																					
(U)	HOLD USER	1089 #	2217	2578	3404	3420	3487	3501	3558	3562	3598	3620	3690											
		3702	3758	3788	3792	3807	3813	4005	4058	4072	4089	4099	4102											
		4132	4133	4157	4207	4219	4229	4232	4339	4374	4405	4491	4640											
		4642	4707	4724	4746	4998	5350	5362	5427	5715	6389	6390	6989											
(D)	I	1347 #	2548	2553	2558	2563	2601	2606	2611	2616	2621	2626	2631											
		2636	2643	2648	2653	2658	2663	2668	2673	2678	2797	2798	2799											
		2800	2808	2818	2828	2833	2843	2850	2860	2870	2880	2889	2890											
		2891	2892	2900	2911	2921	2931	2940	2941	2942	2943	2955	2956											
		3175	3176	3177	3178	3179	3180	3181	3182	3184	3185	3193	3194											
		3195	3196	3197	3198	3199	3200	3211	3212	3213	3214	3215	3216											
		3217	3218	3228	3229	3230	3231	3232	3233	3234	3235	3350	3351											
		3352	3353	3354	3355	3356	3357	3462	3463	3464	3465	3466	3467											
		3468	3469	3476	3477	3478	3479	3480	3481	3482	3483	3490	3491											
		3492	3493	3494	3495	3496	3497	3504	3505	3516	3518	3669	3670											
		3671	3672	3766	3799	3800	3801	3802	3852	3853	3854	3855	3856											
		3857	3858	3859	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	3894	3898											
		3899	3900	3901	3905	3906	3907	3908	3909	3910	3911	3912	3913											
		3914	3915	4051	4065	4113	4138	4305	4310	5109	5385	5849	5850											
		5851	5852	5853	5854	5855	5857	5858	5859	5860	5862	5863	5864											
		5865	5867	5868	5869	5870	5871	5872	5873	5874	5908	7737	7738											
		7739	7741	7742	7744	7745	7747	7748	7749	7750	7751	7752	7753											
		7754	7756	7757	7758	7759	7760	7761	7762	7763	7765	7766	7767											
		7768	7769	7770	7771	7772	7774	7775	7776	7777	7778	7779	7780											
		7781	7783	7784	7785	7786	7787	7788	7789	7790														
(U)	I.CO3	1184 #																						
(U)	I.CO4	1185 #																						
(U)	I.CO5	1186 #																						
(U)	I.CO6	1187 #																						
(U)	I.CO7	1188 #																						
(U)	IO BYTE	1163 #	7580	7630	7636																			
(U)	IO CYCLE	1163 #	7440	7487	7579	7582	7618	7629	7635	7861														
(U)	J	525 #	2321	2354	2358	2367	2372	2380	2401	2407	2431	2436	2457											
		2458	3559	3563	4024	4027	4230	4273	4281	4291	4299	4493	4515											
		4516	4518	4562	4602	4629	4731	4733	4737	4772	4805	4824	4892											
		5106	5243	5244	5526	5528	5529	5618	5813	5820	5821	5836	5936											
		5939	6225	6235	6240	6243	6281	6282	6432	6438	6609	6632	6697											
		6728	6740	6771	6787	6804	6813	6815	6841	6876	6909	6915	6920											
		6960	7199	7265	7289	7364	7408	7516	7519	7591	7598	7604	7661											
		7669	7673	7676	7702	7714	7716	7718	7720	7722	7724	7726	7728											
		7730	7732	7855	7923	7927	7961	8548	8570	8575	8581	8612	8615											
		8616	8629	8638	8641	8642	8746	8834																
	ABORT	8746 #	3946	8727																				
	ACBSET	7120 #	7112																					
	AC_ARX	7732 #	3832	6946																				
	ADD	4057 #	4050	4051	4052	4053																		
	ADDCRY	6240 #	6215	6221	6234																			
	ADJBP	4937 #	4691																					
	ADJBPO	4940 #	4942																					
	ADJBP1	4959 #	4961																					
	ADJBP2	4969 #	4966																					
	ADJBP3	5012 #	4997																					
	ADJBP4	5024 #	5036																					

ADJBP5	5038 #	5032								
ADJBP6	5059 #	5061								
ADJSP	3770 #	3766								
ADJSP1	3787 #	3779								
ADJSP2	3792 #	3784								
AND	2814 #	2807	2808	2809	2810	2839				
ANDCA	2824 #	2817	2818	2819	2820	2876				
ANDCB	2876 #	2869	2870	2871	2872					
ANDCM	2839 #	2832	2833	2834	2835					
AOBJ	3509 #	3504	3505							
AOU	3487 #	3476	3477	3478	3479	3480	3481	3482	3483	
AOS	3404 #	3393	3394	3395	3396	3397	3398	3399	3400	
APRID	7036 #									
APRSO	7028 #	7004								
APRSZ	7031 #	7003								
ARSIGN	2457 #	2399								
ASH	2981 #	2952								
ASHC	3084 #	2956								
ASHC1	3090 #	3088								
ASHCL	3106 #	3095	3108							
ASHCQ1	3117 #	3078	3114							
ASHCR	3099 #	3101								
ASHCX	3111 #	3103								
ASHL	2991 #	2991								
ASHLO	2988 #	2981								
ASHR	2984 #	2970								
ASHX	2995 #	3008								
ASHXX	3008 #	3013								
BACKBP	6959 #	6927	6935							
BACKD	6934 #	8776	8784	8786						
BACKS	6926 #	8783								
BADDATA	8238 #	8223								
BDABT	6403 #	6339								
BDCFLG	6405 #	6384								
BDEC	6287 #									
BDECO	6293 #	6291								
BDEC1	6300 #	6309								
BDEC2	6305 #	6313								
BDEC3	6315 #	6301	6323							
BDEC4	6320 #	6315								
BDEC5	6332 #	6318	6328							
BDECLP	6370 #	6389								
BDFILL	6354 #	6363								
BDSET	6378 #	6398	6401							
BDSUB	6417 #	6321	6371							
BDSUB1	6421 #	6418								
BDSUB2	6437 #	6440								
BDTBL	6394 #	6377								
BITCHK	6908 #	5914	5957	6095	6097	6454				
BIXUB	7557 #	7550	7551	7552	7553					
BIXUB1	7566 #	7562	7564							
BLT	5113 #	5109								
BLT-CLEANUP	5174 #	8768								
BLTBU1	5228 #	5209								
BLTCLR	5133 #	5148								

BLTGOT	5161 #	5136											
BLTGO	5159 #	5125	5153										
BLTLP	5156 #	5171											
BLTLP1	5122 #	5158											
BLTX	5201 #	7800											
BLTXLP	5205 #	5254											
BLTXV	5240 #	5226											
BLTXW	5246 #	5241											
BOTH	2329 #												
BWRITE	2468 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804	
	2814	2824	2856	2866	2886	2896	2907	2917	2947	4058	4072	4124	
	4130	4153	4155	8518	8818								
BYTEAS	4744 #	4768	6851										
BYTEA	4746 #	4701	4720										
BYTEAO	4749 #	4745											
BYTFET	4770 #	4752	4756	6893	6894								
BYTIND	4764 #	4760											
CAIM	3370 #	3351	3352	3353	3354	3355	3356	3357	3359	3360	3361	3362	
	3363	3364	3365	3366									
CHKSN	5813 #	5794											
CLARXL	7720 #	5097	7494										
CLARXO	6431 #	4640	4642	4646	4647	6162	6296	6422					
CLDISP	8761 #	6928	6936	6949	6955	8756							
CLEANED	8764 #	5179	8758										
CLEANUP	8763 #	8761											
CLRB1	6282 #	6280											
CLRBIN	6280 #	6165	6261										
CLRFLG	6918 #	5961	6333										
CLRPTL	7385 #	7388											
CLRPT	7376 #												
CLRSN	5821 #	5599	5689										
CMPDST	6146 #	6126											
CMS	6094 #												
CMS2	6135 #	6105											
CMS3	6111 #	6137	6141										
CMS4	6114 #												
CMS5	6140 #	6119											
CMS6	6126 #	6143											
CMS7	6131 #	6128											
CMS8	6142 #	6140											
CMSDST	6930 #	8782											
COMO	4629 #	4624	4625										
COMOA	4624 #	4628											
COM1	4628 #	4619											
COM1A	4619 #	4627											
COM2	4627 #	4614											
COM2A	4614 #												
CONSO	7029 #	7011											
CONSZ	7032 #	7010											
CONT	7858 #	7853											
CONT 1	7865 #	4045											
CPYSGN	4104 #	4090	4100										
DAC	2489 #	2496	2766	6403									
DADD	4084 #	4079											
DADD1	4087 #	4093											

DBABT	6246 #	6182	6186		
DBDN1	6259 #	6257			
DBDONE	6254 #	6248	6273		
DBFAST	6199 #	6195			
DBIN	6155 #				
DBIN1	6170 #	6173			
DBIN2	6174 #	6170			
DBINLP	6178 #	6196	6206		
DBLDBL	6230 #	6218	6229		
DBLDIV	4580 #	4434	4444	5717	5721
DBLMUL	4224 #	4195			
DBLNEG	4636 #	2781			
DBLNGA	4637 #	2771			
DBLNG1	4645 #	4359	6305		
DBNEG	6264 #	6253			
DBSLOW	6211 #	2182	6193		
DBSLO	6192 #	6200			
DBXIT	6252 #	6180			
DDIV	4386 #	4381			
DDIV1	4403 #	4390	4394		
DDIV2	4406 #	4404			
DDIV3A	4411 #	4416			
DDIV3	4407 #	4401			
DDIV4	4419 #	4412			
DDIV5A	4433 #	4424			
DDIV5B	4457 #	4448			
DDIV5C	4459 #	4454			
DDIV5	4427 #	4423			
DDIV6	4462 #	4455	4457		
DDIV7	4464 #				
DDIV8A	4473 #				
DDIV8	4471 #	4464			
DDIV9	4479 #	4474			
DDIVS	4493 #	4420	4438	5713	
DFAD	5547 #	5534			
DFADJ	5598 #	5563	5571		
DFADJ1	5605 #	5606			
DFADJ2	5607 #				
DFADJ3	5613 #	5609	5613		
DFADJ4	5614 #	5610			
DFADJ5	5616 #	5608	5617		
DFADJ6	5618 #	5616			
DFAS1	5556 #	5552	5554		
DFAS2	5563 #	5557			
DFAS3	5568 #	5559			
DFAS5	5577 #	5565	5573		
DFAS6	5582 #	5583			
DFAS7	5584 #	5582			
DFDV	5687 #	5684			
DFDV1	5693 #	5700			
DFDV2	5701 #	5694			
DFDV3	5712 #	5704			
DFDV4A	5727 #	5730			
DFDV4B	5732 #	5729			
DFDV4	5714 #	5710			

DFMP	5628 #	5623																			
DFMP1	5630 #	5631																			
DFMP2	5656 #	5653																			
DFPR1	2422 #	2416																			
DFPR2	2423 #	2421																			
DFSB	5544 #	5535																			
DIV	4325 #	4309	4310	4311	4312																
DIV1	4334 #	4320	4322	4376																	
DIV2	4337 #	4350																			
DIVA	4346 #	4332																			
DIVB	4352 #	4347																			
DIVC	4365 #	4357	4362																		
DIVHI	4587 #	4581	4584	4598																	
DIVIDE	4552 #	4552																			
DIVSET	4546 #	4542																			
DIVSGN	4532 #	4514	5368																		
DIVSUB	4511 #	4338	4971	4981	5015																
DMLINT	4232 #	4197																			
DMOVNM	2781 #	2776																			
DMOVN	2771 #	2767																			
DMOVN1	2783 #	2775																			
DMTRAP	4216 #	4210																			
DMUL	4168 #	4163																			
DMUL1	4189 #	4178																			
DMUL2	4194 #	4188																			
DMULGO	4223 #	4176																			
DNEG	5770 #	5744	5747	5750	5753																
DNEG1	5772 #	5770																			
DNEG2	5774 #	5771																			
DNN1	5800 #	5796																			
DNN2	5802 #	5798																			
DNNORM	5780 #	5733	5775	5777	5782	5810															
DNNRM1	5807 #	5790																			
DNORM	5741 #	5679	5732	5743	5765																
DNORMO	5678 #	5588																			
DNORM1	5762 #	5755																			
DNORM2	5804 #	5802																			
DOCVT	6292 #	6342	6358																		
DOCVT1	6367 #	6292																			
DOCVT2	6388 #	6368																			
DOCVT3	6385 #	6407																			
DONE	2228 #	2528	2542	2842	2843	3022	3175	3176	3184	3185	3285	3329									
	3332	3335	3350	3462	3713	4219	4724	5066	5141	5165	5250	7032									
	7073	7127	7223	7227	7231	7235	7239	7286	7310	7320	7371	7405									
	7544																				
DPB	4713 #	4686																			
DPB1	4884 #	4723	6875																		
DPB7	4891 #	4885	4886	4887	4888	4889	4913														
DPBSLO	4895 #	4884																			
DRND1	5836 #	5832	5840	5843																	
DROUND	5830 #	5746	5749	5752	5784	5786	5788	5834													
DSMS1	7518 #	7515	7520																		
DSTEA	6893 #	6899																			
DSTIND	6898 #	6895	6896																		
DSUB	4096 #	4080																			

Produced on Advanced Information Services Electronic Laser Printer. PKO/IES6, DTN: 223-7881

Cross Reference Listing

DUMP	7930 #	7914	7921																		
DVSUB1	4538 #	4535																			
DVSUB2	4539 #	4533																			
DVSUB3	4541 #	4545																			
EACALC	2283 #	2346																			
EAPF	8825 #	8805																			
EAPF1	8830 #																				
EDBYTE	6630 #	6593	6621																		
EDEXMD	6541 #	6512																			
EDFILL	6557 #																				
EDFIL1	6572 #	6559																			
EDFLT	6588 #	6534	6578																		
EDFLT1	6606 #	6602	6604																		
EDISP	6471 #	6465																			
EDISP1	6473 #	6473																			
EDIT	6450 #																				
EDITLP	6455 #	6679																			
EDMSG	6615 #	6480																			
EDMSG1	6624 #	6617																			
EDN1A	6661 #	6666																			
EDNOP	6650 #	6482	6484	6486	6500	6509	6535	6546	6567	6573	6626	6627									
	6641																				
EDNOP1	6651 #	6645																			
EDNOP2	6668 #	6662	6668																		
EDOPR	6500 #	6477																			
EDSEL	6551 #	6504																			
EDSFLT	6577 #	6561																			
EDSKP	6638 #	6489	6492	6494																	
EDSKP1	6642 #	6639																			
EDSPUT	6564 #	6572	6582																		
EDSSIG	6534 #	6506																			
EDSTOP	6519 #	6502	6562																		
EDSTP1	6525 #	6528																			
ENDSKP	6262 #	6086																			
EQV	2886 #	2879	2880	2881	2882																
EXCH	2588 #	2584																			
EXTDSP	5926 #																				
EXTEA	5925 #																				
EXTEAO	5922 #	5920																			
EXTEA1	5923 #	5931																			
EXTEND	5912 #	5908																			
EXTXT	5935 #	5927	5941																		
EXTIND	5931 #	5928																			
FAD	5283 #	5260	5261	5262	5263	5264	5265	5266													
FAS1	5285 #																				
FAS2	5294 #	5285																			
FAS3	5297 #	5290	5291	5294	5295																
FAS4	5299 #	5299																			
FDV	5346 #	5334	5335	5336	5338	5339	5340	5341													
FDVO	5352 #	5349																			
FDV1	5354 #	5352																			
FDV2	5355 #	5353																			
FDV3	5356 #	5354	5355																		
FDV4	5358 #	5359																			
FDV5	5361 #	5358																			

Produced on Advanced Information Services Electronic Laser Printer, PK011E55, DTN: 232-7881

FDV6	5364 #	5361							
FDV7	5368 #	5369							
FDV8	5371 #								
FDV9	5379 #	5371	5376	5377					
FETIND	2339 #	2304	2309	2331					
FIX	5424 #	5419	5420						
FIX++	4556 #	4557							
FIX1++	4558 #	4556							
FIXL	5439 #	5429	5439						
FIXPC	8759 #	4232	6419						
FIXR	5434 #	5435							
FIXT	5446 #	5443							
FIXX	5443 #	5436							
FIXX1	5444 #	5450							
FL-BWRITE	2515 #	5444	5448	5494	5516				
FLEX	5494 #	5490							
FLTR	5389 #	5384							
FLTR1A	5397 #	5394							
FLTR1	5393 #	5390							
FLTR2	5402 #	5395	5397						
FLTR3	5404 #	5405							
FMP	5316 #	5305	5306	5307	5309	5310	5311	5312	
FMP1	5321 #	5319	5320						
FP-LONG	3931 #	3909	3910	3911	3912	3913	3914		
FPRO	2396 #	2390							
FPR1	2400 #								
FSB	5280 #	5268	5269	5270	5271	5272	5273	5274	
FSC	5410 #	5385							
GETPCW	7194 #	3977							
GETSRC	6842 #	6117	6552						
GOEXEC	4001 #	8832							
GRP700	7035 #	6996							
GRP701	7014 #	6997							
GRP702	7208 #	7204							
GSRC	6840 #	6074	6727						
GSRC1	6849 #	6845							
GTFILL	6912 #	5954	6451						
GTPCW1	7197 #	7188	7197						
H1	7849 #	7915							
HALT	3578 #	3530							
HALTED	7839 #	2191	2248	2266	2517	3580	7469	7501	8750
HALTLP	7855 #	7849							
HARD	8725 #	7708	8225	8227	8241				
HLL	2691 #	2600	2601	2644	2716				
HLLC	2757 #	2630	2631	2632	2633				
HLLD	2761 #	2620	2621	2622	2623				
HLLZ	2759 #	2610	2611	2612	2613				
HLR	2718 #	2647	2648						
HLRE	2750 #	2677	2678	2679	2680				
HLRM	2727 #	2649							
HLRO	2754 #	2667	2668	2669	2670				
HLRS	2730 #	2650							
HLRZ	2752 #	2657	2658	2659	2660				
HRL	2716 #	2605	2606						
HRLE	2743 #	2635	2636	2637	2638				

HRLM	2721 #	2607							
HRLO	2747 #	2625	2626	2627	2628				
HRLS	2724 #	2608							
HRLZ	2745 #	2615	2616	2617	2618				
HRR	2689 #	2602	2642	2643	2718				
HRRE	2736 #	2672	2673	2674	2675				
HRRO	2740 #	2662	2663	2664	2665				
HRRZ	2738 #	2652	2653	2654	2655				
HSBDON	7957 #								
IBP	4689 #	4682							
IBPS	4730 #	4692	4697	4711					
IBPX	4737 #	3829	4730	8580					
IDIV	4316 #	4304	4305	4306	4307				
IDPB	4711 #	4685							
IDST	6886 #	6147	6870						
IDSTX	6890 #	6887							
ILDB	4697 #	4683							
IMUL	4118 #	4112	4113	4114	4115				
IMUL1	4124 #	4341							
IMUL2	4126 #	4123							
IMUL3	4132 #	4129							
INCAR	7722 #	6048	7474						
INCPC	2276 #	2271							
INDEX	2316 #								
INDRCT	2334 #								
IOEA	7645 #	7577	7615						
IOEA1	7651 #	7653							
IOEA2	7655 #	7651							
IOEA1	7663 #	7656							
IOEAX	7672 #	7659							
IOR	2866 #	2859	2860	2861	2862	2927			
IORD	7576 #	7532	7543	7558					
IORD1	7585 #	7581							
IORD2	7595 #	7590							
IORD3	7601 #	7596	7602						
IOT700	7794 #	7737	7738	7739					
IOT710	7796 #	7741	7742						
IOT720	7803 #	7744	7745						
IOT730	7805 #	7747	7748	7749	7750	7751	7752	7753	7754
IOT740	7807 #	7756	7757	7758	7759	7760	7761	7762	7763
IOT750	7809 #	7765	7766	7767	7768	7769	7770	7771	7772
IOT760	7811 #	7774	7775	7776	7777	7778	7779	7780	7781
IOT770	7813 #	7783	7784	7785	7786	7787	7788	7789	7790
IOW1	7685 #	7690							
IOW2	7697 #	7687	7693						
IOW3	7700 #	7707							
IOW4	7705 #	7701							
IOW5	7708 #	7695							
IOWAIT	7681 #	7586	7622						
IOWR	7614 #	7547	7567						
IOWR1	7620 #	7630	7636						
IOWR2	7627 #	7617							
IOWR3	7633 #	7628	7634						
ITRAP	8209 #	6050							
JEN	3586 #	3540							

JEN1	7515 #	3592	3607						
JEN2	3592 #	3584							
JFCL	3598 #	3518							
JFFO	3016 #	2955							
JFFO1	3024 #	3021							
JFFOL	3038 #	3040							
JMPA	3441 #	3846							
JRA	3840 #	3802							
JRST	3525 #	3466	3516						
JRSTO	3554 #	3551	3575	3590					
JRST1	3572 #	3567	3571						
JRST10	3583 #	3537							
JRSTF	3548 #	3528							
JSA	3825 #	3801							
JSP	3806 #	3800							
JSR	3812 #	3799							
JSTAC	3707 #	3697	3704	3742					
JSTAC1	3710 #	3761							
JSYS	3925 #	3906							
JUMP	3473 #	3463	3464	3465	3467	3468	3469		
JUMP-TABLE	3426 #	3473	3487	3501	3511				
JUMP-	3453 #	3432	3435	3438	3601				
JUMPA	3457 #	3441	3444	3447	3450	3809	4007		
KIF50	8694 #	8717	8732						
KIF80	8715 #								
KIF90	8720 #	8711							
L-BDEC	5886 #	5859	5860						
L-CMS	5880 #	5849	5850	5851	5853	5854	5855		
L-DBIN	5884 #	5857	5858						
L-EDIT	5882 #	5852							
L-MVS	5888 #	5862	5863	5864	5865				
L-SPARE-A	5892 #	5868							
L-SPARE-B	5894 #	5869							
L-SPARE-C	5896 #	5870	5871	5872	5873	5874			
L-XBLT	5890 #	5867							
LDB	4699 #	4684							
LDB1	4786 #	4706	6150	6855					
LDB7	4800 #	4791	4792	4793	4794	4795			
LDBSH	4815 #	4810							
LDBSWP	4809 #	4788							
LDPI2	7361 #	3593	7350						
LOADAR	7713 #	6396	6459	6542	7647				
LOADARX	7716 #	6751	7281						
LOADPI	7360 #	2194							
LOADQ	7718 #	5115	6103						
LSH	2965 #	2954							
LSHC	3067 #	2958							
LSHCL	3075 #	3067	3075						
LSHCR	3071 #	3071							
LSHCX	3077 #	3072	3144	3149					
LSHL	2972 #	2967							
LUUO	4031 #	3852	3853	3854	3855	3856	3857	3858	3859
LUUO1	4034 #	5880	5882	5884	5886	5888	5890	5892	5894
MAP	8192 #	8188							
MAPDON	8815 #	8770							5896

MOVE	2575 #	2549	2572	2579	2603	2645	2827	2828	2829	2830	2844	2845
	4105	4106	4132	4133	4157	4967						
MOVELP	5992 #	5972	6001									
MOVF1	6692 #	6698										
MOVFIL	6698 #	6024	6085									
MOVLPO	6001 #	5986										
MOVMM	2572 #	2562	2564	2565								
MOVN	2577 #	2557	2558	2559	2560							
MOVPAT	6467 #	6462	6464	6467								
MOVRJ	6022 #	5978										
MOVVS	2569 #	2552	2553	2554	2555	2722	2728	2745	2747	2752	2754	
MOVSTX	6075 #											
MOVSTO	6070 #	5975										
MOVST1	6071 #	6026	6081									
MOVST2	6084 #	5995	6076									
MOVST3	6088 #	6084										
MOVST4	6080 #	6037										
MSKPAT	6468 #	6463										
MUL	4145 #	4137	4138	4139	4140							
MUL+	4267 #	4251	4263	4269	4287							
MUL-	4285 #	4277	4295									
MULBY4	6229 #	6212										
MULSB1	4249 #	5639										
MULSUB	4248 #	4121	4149	5042	5323							
MULTIPLY	4260 #	4228	5644	5656	5663							
MUUO	3938 #	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	3894			
MVABT	6006 #	5997										
MVABT1	6009 #	6011										
MVABT2	6013 #	6009										
MVEND	6016 #											
MVS	5951 #											
MVS1	5968 #	5966										
MVSK2	6056 #	6030										
MVSK3	6046 #	6042										
MVSKP	6029 #	6022	6040	6053	6064							
MVSKP1	6040 #	6035										
MVSKP2	6052 #	6044										
MVSO	5980 #	5973										
MVSO1	5983 #	5983										
NEXT	4026 #	3967	3971	7931	7934	7936	7938	7941	7944	7947	7950	7953
	7955											
NEXTAR	8834 #	8798	8826									
NICOND	2237 #	2522	3043	3089	3117	3429	3711	3787	3794	4482	4485	5440
	5446	5805	8202									
NICOND-FETCH	2256 #	2228	2229	2231	2232	2292	2298	3326	3338	3453	3454	3457
	3458	3525	3526	3552	3594	4218	4693	7352	7623			
NIDISP	3089 #	3789	3793	4339	4374	4405	4491	4708	4998	5350	5362	5427
	5715											
NODDIV	4490 #	4427	4429									
NODIV	4374 #	4368	4372									
NOMOD	2320 #											
NXTWRD	4736 #	4732										
ORCA	2906 #	2899	2900	2901	2902							

Produced on Advanced Information Services Electronic Laser Printer, PK011556, DTN: 223-7881

ORCB	2937 #	2930	2931	2932	2933																
ORCM	2927 #	2920	2921	2922	2923																
PAGE-FAIL	8207 #																				
PF100	8459 #	8436																			
PF105	8467 #	8463																			
PF107	8440 #	8470	8474																		
PF110	8477 #	8443																			
PF120	8520 #	8500																			
PF125	8539 #	8537	8743																		
PF130	8502 #	8498	8504																		
PF140	8496 #																				
PF25	8264 #	8266																			
PF30	8288 #	8277																			
PF35	8301 #	8290																			
PF40	8306 #	8298																			
PF45	8313 #	8333	8412																		
PF50	8345 #	8317																			
PF60	8329 #	8331																			
PF70	8351 #																				
PF75	8363 #	8365																			
PF76	8386 #	8382	8384																		
PF77	8391 #	8295	8303																		
PF80	8422 #	8396																			
PF90	8431 #																				
PFD	8215 #	8217																			
PFDBIN	6953 #	8780																			
PFDONE	8492 #	8718	8722																		
PFGACO	6955 #																				
PFMAP	8245 #	8201	8229	8233	8235																
PFPI1	8734 #	8219	8231																		
PFPI2	8742 #																				
PFT	8750 #	8325	8345	8413	8422	8439	8576	8601	8606	8630	8694										
PFT1	8751 #	8739																			
PFT10	8794 #	8792																			
PFT1A	8758 #	8754																			
PFT2	8790 #	8766																			
PFT3	8793 #	8819																			
PFTICK	8452 #	8405																			
PI	7418 #	7420	8793																		
PI10	7436 #	7428	7429	7430	7431	7432	7433	7434													
PI40	7454 #	7451																			
PI50	7455 #	7505																			
PIEXIT	7350 #	3622																			
PIJSR	7469 #	7458																			
PIP1	7428 #	7421																			
PIP2	7429 #	7422																			
PIP3	7430 #	7423																			
PIP4	7431 #	7424																			
PIP5	7432 #	7425																			
PIP6	7433 #	7426																			
PIP7	7434 #	7427																			
PISET	3621 #	7476																			
PIXPCW	3630 #	7461																			
POP	3717 #	3671																			
POPJ	3750 #	3672																			

Produced on Advanced Information Services Electronic Laser Printer, PKO11556, DTN: 223-7881

SNNORM	5501 #	5497	5499	5501	5505																
SNNOT	5512 #	5508																			
SNNOT1	5515 #	5510																			
SNNOT2	5516 #	5512	5514	5515																	
SNORM	5481 #	5300	5329	5406	5481	5493															
SNORMO	5406 #	5379	5391	5398	5413	5414															
SNORM1	5493 #	5489																			
SOJ	3501 #	3490	3491	3492	3493	3494	3495	3496	3497												
SOS	3420 #	3409	3410	3411	3412	3413	3414	3415	3416												
SRCMOD	6725 #	5993	6178	6820	6824	6826															
SRND1	5526 #	5523																			
SROUND	5523 #	5483	5485	5487	5502	5503	5504	5524													
SSWEEP	7407 #	7380	7393																		
STAC	2521 #	2490	2534	2539	2547	2548	2563	2592	2772	2973	2986	2995									
	5759	5766	5811	6017	6525	7828															
STAC34	6272 #	6267																			
START	2250 #	3821	3836																		
STBOTH	2532 #																				
STBTH1	2537 #	2485																			
STDBTH	2494 #																				
STMAC	3694 #	3686																			
STMEM	2526 #	7835																			
STOBR	7728 #	3631	3980	7472																	
STOPC	7730 #	3634	7848																		
STORE	2540 #	2792	3642																		
STPF1A	6951 #	6943																			
STPTR1	8603 #	8600																			
STPTR2	8609 #	8605																			
STPTR3	8615 #	8611																			
STRPF	6939 #	8772	8774																		
STRPFO	6940 #	6957																			
STRPF1	6943 #																				
STRPF2	6954 #	6951																			
STRPF3	6948 #																				
STRPF4	6957 #	8778																			
STRTIO	7726 #	7442	7490	7864																	
STSELF	2484 #	2550																			
STUBRS	7132 #	7134																			
SUB	4071 #	4064	4065	4066	4067																
SWEEP1	7398 #	7401																			
SWEEP	7392 #	7144	7174																		
T1LSH	5243 #	5214	5234																		
TDONE	3300 #	3291	3294																		
TDX	3271 #	3179	3188	3193	3197	3202	3206	3211	3215	3220	3224	3228									
	3232	3237	3241																		
TDXX	3276 #	3177	3181	3186	3190	3195	3199	3204	3208	3213	3217	3222									
	3226	3230	3234	3239	3243																
TENLP	2180 #	2187																			
TEST-TABLE	3282 #	3271	3277	7533																	
TICK	7246 #	6059																			
TIOX	7532 #	7525	7526	7527	7528																
TOCK	7249 #	7303	8453	8738																	
TOCK1	7253 #	7259																			
TOCK2	7260 #	7254																			
TOCK3	7264 #	7271																			

Produced on Advanced Information Services Electronic Laser Printer, PK01E56, DTN: 223-7881

Cross Reference Listing

TRAP	6964 #	2241	2244	2247	2259	2262	2265						
TRNAR	6746 #	6555											
TRNFNC	6768 #	6756	6775	6779	6783	6795							
TRNNS1	6818 #	6808											
TRNNS2	6821 #	6818											
TRNRET	6800 #	6769											
TRNSIG	6781 #	6791											
TRNSS	6807 #	6802											
TRNSS1	6803 #	6811											
TRP1	6983 #	6973	6981										
TSX	3269 #	3180	3189	3194	3198	3203	3207	3212	3216	3221	3225	3229	
	3233	3238	3242										
TSXX	3274 #	3178	3182	3187	3191	3196	3200	3205	3209	3214	3218	3223	
	3227	3231	3235	3240	3244								
TXXX	3285 #	3300											
TXZX	3299 #	3288											
UMOVEM	7831 #	7819											
UMOVE	7823 #	7818											
UPCST	8567 #	8350	8426										
UUD	3937 #	3898	3905										
UUD101	3919 #	3899											
UUD102	3921 #	3900											
UUD103	3923 #	3901											
UUD106	3927 #	3907											
UUD107	3929 #	3908											
UUD247	3933 #	3915											
UUDFLG	4022 #	3962											
UUDGO	3943 #	2435	3529	3538	3541	3543	3544	3545	3578	3583	3586	3606	
	3627	3640	3919	3921	3923	3925	3927	3929	3931	3933	6908	6991	
	7000	7001	7002	7006	7007	7008	7009	7014	7015	7016	7019	7020	
	7021	7022	7023	7024	7025	7026	7218	7240	7335	7794	7803	7805	
	7807	7809	7811	7813									
UUOPCW	3974 #												
VECINT	7483 #	7446	7486										
VECIN1	7502 #	7500											
WRAPR	7050 #												
WRAPR1	7069 #	7081											
WRAPR2	7075 #	7068											
WRCSE	7225 #												
WRCSTM	7233 #												
WRCST	8579 #	8359	8484										
WREBR	7147 #												
WREBR1	7149 #	7149											
WRHSB	7237 #												
WRINT	7316 #												
WRIO	7547 #	7537	7539										
WRPI	7332 #												
WRPUR	7229 #												
WRSPB	7221 #												
WRTHSB	7918 #	7842											
WRTIME	7276 #												
WRTIM1	7288 #	7257	7284										
WRUBR	7102 #												
XCT	3651 #	3647											
XCT1	3663 #	3657	6990	7868									

AC[]	1850 #	4483	5691	5699	6202	6266								
AC[]_Q	1751 #	3117	4179	4187	4198	4210								
AC[]_Q.AND.[]	1740 #	4437	4481	4611	4616	4621	4627	4628						
AC[]_[]	1727 #	2489	2772	3043	4454	4460	5766	5811	6009	6029	6046	6129		
	6163	6222	6224	6259	6272	6273	6309	6341	6347	6358	6363	6386		
	6387	6546	6849	6890	6928	6936	6943	6948						
AC[]_[] TEST	1728 #	6821												
AC[]_[] VIA AD	1725 #													
AC[]_[]*2	1730 #	4196												
AC[]_[]+1	1729 #	6695												
AC[]_[]+Q	1735 #													
AC[]_[]+[]	1737 #	4479	6205											
AC[]_[]-[]	1736 #	5963												
AC[]_[] .AND.[]	1739 #	4457	5758	5804										
AC[]_[] .EQV.Q	1741 #	4211	4213	4215	4484									
AC[]_ -Q	1747 #													
AC[]_ -[]	1742 #													
AC[]_ .NOT.[]	1745 #													
AC[]_O	1749 #	2173	3022	4455	6036	6084	6280	6282						
AC[]_1	1750 #	2174												
AC_ -[]	1743 #	5803												
AC_ .NOT.[]	1746 #	5802												
AC_Q	1748 #	4453												
AC_[]	1731 #	2521	3078	3300	3389	3429	3441	3710	3712	3808	4629	4707		
	5114	5178	5203	5440	5446	6086	6122	6385	6606	6679	6955	7544		
	7732	8202												
AC_[] TEST	1732 #	3432	3435	3438	3444	3447	3450	4209	6260	6306				
AC_[] VIA AD	1726 #													
AC_[]+1	1733 #	6931												
AC_[]+Q	1734 #													
AC_[]+[]	1738 #	3111	3775	3780										
AC_[] .OR.[]	1744 #	5065	6049	6088										
AD_FLAGS	2027 #	2578	3404	3420	3487	3501	4058	4072	4089	4099	4102	4229		
	4640	4642												
AD_FLAGS EXIT	2075 #	4058	4072											
AD_PARITY	1953 #	2181	2357	2370	2371	2588	2689	2691	2722	2728	2781	2965		
	2968	2982	3016	3473	3717	3750	3772	3828	3840	4147	4189	4326		
	4717	5113	5201	5283	5316	5348	5412	5558	5955	5958	5973	5976		
	6006	6022	6094	6096	6114	6146	6158	6161	6189	6192	6194	6201		
	6211	6252	6254	6255	6287	6292	6293	6294	6315	6320	6332	6367		
	6452	6510	6551	6588	6590	6694	6807	6814	6840	6842	6869	6926		
	6934	6940	7547	7834										
ADD .25	1984 #	2577	2947	3370	3420	3501	3527	3536	3539	4071	4088	4096		
	4101	4275	4279	4321	4365	4398	4421	4422	4428	4449	4475	4516		
	4517	4518	4534	4536	4543	4549	4580	4610	4615	4620	4625	4637		
	4641	4645	4647	4976	5033	5050	5100	5119	5137	5161	5249	5280		
	5355	5358	5359	5366	5377	5390	5482	5484	5486	5488	5499	5515		
	5544	5583	5696	5707	5714	5730	5771	5776	5797	5803	5837	5963		
	5996	6010	6031	6098	6134	6135	6136	6183	6247	6265	6268	6336		
	6353	6361	6380	6388	6389	6423	6425	6428	6476	6818	6939	6947		
	6953	7261	7361	7385	7398	7645	8759	8819						
ADL_PARITY	1946 #													
ADR_PARITY	1950 #													
AREAD	2070 #	2321												
ASH	1961 #	2425	2985											

ASH AROV	2012 #	2991	3107	3110									
ASH36 LEFT	2978 #	2991											
ASHC	1966 #	2991	3100	3103	3107	3110	4268	4276	4286	4294	4400	4404	
	4473	4477	4493	5299	5402	5404	5434	5578	5580	5584	5586	5602	
	5606	5613	5616	5618	5630	5671	5675	5710	5724	5742	5748	5751	
	5764	5781	5785	5787	5809	5834							
B DISP	2071 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804	
	2814	2824	2856	2866	2886	2896	2907	2917	2947	3271	3277	3370	
	3389	3390	3473	3487	3501	3511	3696	3703	3741	4058	4072	4124	
	4130	4153	4155	5207	5436	5444	5448	5494	5506	5516	5523	5927	
	5969	6113	6168	6375	6729	6801	6808	7533	7560	7578	7589	7616	
	8518	8818											
BAD PARITY	1955 #												
BASIC DIV STEP	4528 #	4549	4552										
BWRITE DISP	2072 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804	
	2814	2824	2856	2866	2886	2896	2907	2917	2947	4058	4072	4124	
	4130	4153	4155	8518	8818								
BYTE DISP	2089 #	4705	4722	6150	6853	6874							
BYTE STEP	4677 #	5981	5983										
CALL IBP	4671 #	4692	4697	4711									
CALL LOAD PI	2095 #	3593	7350										
CALL []	1985 #	2182	2194	2399	2771	2781	3551	3590	3592	3593	3607	3631	
	3634	3829	3832	3946	3962	3967	3971	3977	3980	4121	4149	4176	
	4185	4195	4205	4228	4338	4359	4414	4420	4434	4438	4444	4490	
	4514	4692	4697	4701	4706	4711	4720	4723	4971	4981	5015	5042	
	5097	5113	5115	5202	5214	5216	5234	5236	5323	5365	5368	5373	
	5375	5483	5485	5487	5502	5503	5504	5563	5571	5599	5639	5644	
	5656	5663	5670	5689	5713	5717	5721	5746	5749	5752	5784	5786	
	5788	5794	5914	5954	5957	5961	5993	5999	6024	6048	6059	6074	
	6078	6085	6095	6097	6103	6117	6126	6147	6162	6165	6178	6193	
	6204	6212	6215	6218	6221	6229	6234	6261	6296	6305	6321	6333	
	6360	6371	6379	6396	6422	6451	6454	6459	6534	6542	6552	6555	
	6565	6578	6593	6600	6621	6625	6693	6727	6751	6851	6870	6875	
	6927	6935	6946	7188	7257	7281	7284	7303	7350	7380	7393	7442	
	7472	7474	7490	7494	7532	7543	7558	7577	7586	7615	7622	7647	
	7840	7842	7848	7864	7912	7914	7931	7934	7936	7938	7941	7944	
	7947	7950	7953	7955	8314	8321	8332	8339	8350	8359	8377	8380	
	8393	8416	8426	8453	8484	8569	8727	8738	8798	8826			
CHANGE FLAGS	2005 #	2217	2218	2578	3404	3420	3487	3501	3557	3558	3561	3562	
	3598	3619	3620	3690	3702	3758	3788	3792	3807	3813	4004	4005	
	4006	4058	4072	4089	4099	4102	4132	4133	4157	4207	4219	4229	
	4232	4339	4374	4405	4491	4640	4642	4707	4724	4746	4998	5350	
	5362	5427	5715	6389	6390	6988	7460	7475	8756				
CHK PARITY	1954 #	2269	2339	2353	2362	2366	2383	2394	2405	2411	2424	2495	
	2527	2533	2538	2541	2591	2787	3087	3406	3549	3573	3588	3613	
	3617	3678	3695	3701	3723	3740	3746	3759	3817	3845	3967	3969	
	3975	4002	4041	4119	4212	4214	4317	4327	4411	4433	4440	4737	
	4765	4771	4892	5157	5206	5912	5931	6351	6541	6591	6594	6622	
	6898	6913	6983	7105	7222	7226	7230	7234	7238	7278	7308	7310	
	7317	7371	7455	7497	7621	7668	7713	7716	7718	7728	7730	7827	
	7846	7866	7931	7932	7934	7935	7938	7939	7941	7942	7944	7945	
	7947	7948	7950	7951	7953	7954	7956	8310	8392	8628	8797	8825	
	8827												
CHK PARITY L	1947 #												
CHK PARITY R	1951 #												

CLEANUP DISP	2097 #	8761											
CLEAR ARXO	1703 #	2771	4636	6264	6305	6431	6437						
CLEAR CONTINUE	1994 #												
CLEAR EXECUTE	1995 #												
CLEAR RUN	1996 #												
CLEAR []O	1702 #	2771	4358	4636	6241	6257	6264	6305	6431	6437			
CLR FPD	2014 #	3690	3807	3813	4207	4707	4724	6390					
CLR IO BUSY	2000 #	7576	7614										
CLR IO LATCH	2001 #	7661	7669	7673	7676	7685	7691	7700	7705				
CLRCSH	1991 #	7382	7383	7386									
DFADJ	5595 #	5606											
DISMISS	2094 #	3592	3607										
DIV	1969 #	4549	4552	4553	4600	548.1	5493	5501	5505	5598	5600	5601	
DIV DISP	2088 #												
DIV STEP	4529 #	4552											
DONE	2080 #	2228	2232	3326	3453	3458	4218	4693	7352	7623			
DPB SCAD	4880 #	4885	4886	4887	4888	4889							
EA MODE DISP	2069 #	2279	2345	3554	4744	4747	5923	6899	7657				
END BLT	2037 #	5140	5164	5250									
END MAP	2038 #	8508											
END STATE	2035 #	5140	5164	5250	6017	6089	6112	6262	6390	6394	6456	6728	
	8508	8765	8783	8784	8815								
EXIT	2074 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804	
	2814	2824	2856	2866	2886	2896	2907	2917	2947	4124	4130	4153	
	4155	8518	8818										
EXP TEST	2026 #	5490	5506	5836									
FETCH	1806 #	2228	2229	2231	2232	2251	2291	2297	2405	2430	3020	3326	
	3338	3453	3454	3457	3458	3525	3526	3552	3594	3709	4218	4693	
	4702	7352	7623	7859									
FE_-1	1926 #												
FE_-12.	1924 #	3024											
FE_-2	1923 #												
FE_-FE	1910 #	4809	4903										
FE_-FE+200	1937 #	5354	5355										
FE_-FE+S#	1913 #	4814											
FE_-FE-1	1911 #	2966	3001										
FE_-S-10	1920 #	4818	4884										
FE_-S-20	1919 #												
FE_O	1925 #												
FE_EXP	1932 #	2397	2412	5289	5570								
FE_FE+1	1927 #	2969	2983	3004	5369	5485	5487	5503	5504	5524	5529	5749	
	5752	5786	5788	5834	5842								
FE_FE+10	1929 #	4816	4821	4897	4905								
FE_FE+2	1928 #	5326											
FE_FE+4	1931 #	3039											
FE_FE+P	1935 #	6521											
FE_FE+S#	1938 #	5668	5679	6654									
FE_FE+SC	1914 #	6653											
FE_FE-1	1930 #	5481	5493	5501	5505	5585	5587	5742	5764	5781	5809		
FE_FE-19	1912 #												
FE_FE-200	1936 #	5328											
FE_FE.AND.S#	1915 #	4704	4901	6149	6527	6652	6854						
FE_P	1916 #	4700	4900	6651	6851	6891							
FE_S	1917 #												
FE_S#	1921 #	3024	5213	5215	5233	5235	5391	5395	5397	5398	6520		

	8469	8473	8507	8511	8514	8525	8595	8700	8705	8710	8716	8730
	8793											
IBP DP	4668 #	4692	4697	4711	6033	6843	6886	6959				
IBP SCAD	4669 #	4692	4697	4711	6033	6843	6886					
IBP SPEC	4670 #	4692	4697	4711								
INH CRY18	1957 #	3510	3680	3727	3756	3776	3781	5030	5105			
INST DISP	2073 #	2354	2358	2367	2372	2380	2401	2407	2431	2436		
INTERRUPT TRAP	2086 #											
JFCL FLAGS	2023 #	3598										
JUMP DISP	2079 #	3473	3487	3501	3511							
JUMPA	2081 #	3454	3457	3525	3526	3552	3594					
LDB SCAD	4782 #	4791	4792	4793	4794	4795						
LEAVE USER	2021 #	2217	4005									
LOAD AC BLOCKS	1989 #	2163	7126	7143								
LOAD BYTE EA	1980 #	4700	4719	4767	5919	5931	6850					
LOAD DST EA	1983 #	6887	6889	6898								
LOAD FE	1863 #	2379	2397	2412	2966	2969	2973	2983	2985	3001	3004	3007
	3012	3024	3039	3085	4700	4704	4716	4809	4814	4816	4818	4821
	4884	4897	4900	4901	4903	4905	5213	5215	5233	5235	5243	5244
	5289	5317	5326	5328	5348	5354	5355	5369	5391	5395	5397	5398
	5412	5481	5485	5487	5493	5501	5503	5504	5505	5524	5529	5570
	5585	5587	5651	5668	5679	5702	5728	5742	5749	5752	5764	5781
	5786	5788	5809	5834	5842	5980	5981	5983	6149	6520	6521	6527
	6651	6652	6653	6654	6851	6854	6873	6891				
LOAD FLAGS	2025 #	2218	3557	3561	3619	4004	7460	7475	8756			
LOAD IND EA	1981 #	2341										
LOAD INST	1978 #	2270	3655	6984	7867							
LOAD INST EA	1979 #	3550	3574	3589	7664							
LOAD IR	1974 #	5918										
LOAD PAGE TABLE	1988 #	7377	8534									
LOAD PCU	2019 #	4006										
LOAD PI	1992 #	3608	7363	7438	7920							
LOAD PXCT	1986 #	3660										
LOAD SC	1862 #	2175	2185	2378	2397	2412	2428	2989	2991	3067	3068	3069
	3071	3075	3085	3094	3097	3101	3108	3139	3140	3141	3143	3147
	4119	4147	4184	4204	4224	4250	4262	4268	4276	4286	4294	4337
	4434	4442	4549	4552	4592	4938	4941	4954	4960	4965	4978	4994
	5039	5054	5060	5283	5287	5288	5297	5299	5321	5368	5402	5405
	5410	5428	5429	5430	5432	5435	5439	5549	5559	5569	5603	5606
	5613	5628	5631	5639	5643	5646	5656	5662	5717	5718	6294	6323
	6462	6464	6467	6471	6473	6644	6650	6656	6661	6668	7129	7133
	7147	7149	7178	7180	7188	7195	7197	7387	7400	7407	7595	7602
	7627	7633	7681	7688	7697	7701	8262	8265	8313	8327	8330	8361
	8364	8497	8503									
LOAD SRC EA	1982 #											
LOAD VMA	1785 #	2161	2180	2184	2228	2228	2229	2231	2232	2232	2250	2276
	2291	2297	2303	2307	2330	2334	2343	2363	2405	2415	2418	2430
	2784	2788	2789	3020	3326	3326	3338	3453	3453	3454	3457	3458
	3458	3525	3526	3527	3532	3536	3539	3552	3565	3568	3594	3609
	3615	3630	3632	3632	3641	3681	3708	3718	3730	3735	3751	3814
	3842	3964	3965	3965	3979	3979	3998	3999	4026	4026	4035	4042
	4043	4218	4218	4693	4693	4702	4749	4753	4757	4761	5094	5122
	5130	5144	5145	5149	5150	5167	5168	5246	5252	5252	5926	5928
	5929	5952	6100	6349	6376	6421	6427	6434	6450	6455	6511	6589
	6631	6746	6893	6894	6895	6896	6972	6980	7102	7279	7304	7309

Produced on Advanced Information Services Electronic Laser Printer, PKC/ISLS, DTN: 223-7881

LSH
LSHC
LUUO
MEM CYCLE

7309	7352	7352	7370	7376	7439	7454	7454	7454	7455	7496	7496
7504	7585	7620	7623	7623	7645	7667	7726	7823	7831	7841	7844
7844	7847	7847	7858	7914	7930	7930	7933	7933	7940	7940	7943
7943	7946	7946	7949	7949	7952	7952	7959	8245	8293	8294	8294
8301	8302	8302	8306	8308	8386	8388	8526	8542	8623	8625	8740
8746	8794	8795	8795	8831	8831	8834	8834				
1962	#										
1965	#	3071	3075	4581	4584						
2083	#	5880	5882	5884	5886	5888	5890	5892	5894	5896	
1782	#	2161	2180	2184	2228	2228	2228	2229	2229	2231	2232
2232		2232	2250	2251	2268	2276	2291	2291	2297	2297	2302
2307		2308	2330	2331	2334	2335	2339	2343	2352	2361	2363
2365		2393	2404	2405	2405	2415	2416	2418	2419	2423	2430
2494		2526	2532	2537	2540	2589	2590	2784	2785	2787	2788
2791		3020	3020	3326	3326	3326	3338	3338	3405	3406	3453
3453		3454	3454	3454	3457	3457	3458	3458	3525	3525	3526
3527		3527	3532	3532	3536	3536	3539	3539	3548	3552	3552
3566		3568	3569	3572	3587	3594	3594	3609	3613	3615	3616
3630		3630	3632	3632	3633	3636	3641	3641	3677	3681	3682
3700		3708	3709	3718	3719	3722	3730	3732	3735	3737	3739
3751		3753	3759	3814	3815	3816	3826	3842	3843	3844	3964
3965		3966	3968	3974	3979	3979	3979	3998	3999	4000	4001
4026		4026	4035	4036	4040	4042	4043	4044	4218	4218	4693
4693		4693	4702	4702	4730	4736	4737	4749	4750	4753	4754
4758		4761	4762	4764	4770	4891	4892	5094	5095	5122	5123
5131		5133	5144	5145	5147	5149	5150	5152	5156	5159	5167
5170		5205	5246	5246	5248	5252	5252	5253	5912	5926	5928
5929		5929	5931	5952	5953	6100	6100	6349	6350	6351	6376
6421		6427	6434	6450	6450	6455	6455	6511	6511	6543	6545
6589		6591	6594	6622	6631	6631	6746	6747	6893	6893	6894
6895		6895	6896	6896	6898	6912	6968	6971	6972	6976	6979
6983		7102	7103	7104	7221	7222	7225	7226	7229	7230	7233
7237		7238	7276	7277	7279	7280	7304	7305	7308	7309	7309
7310		7316	7317	7352	7352	7352	7370	7370	7371	7376	7439
7454		7454	7454	7455	7455	7470	7491	7496	7496	7497	7504
7585		7587	7620	7621	7623	7623	7645	7646	7667	7667	7668
7713		7716	7718	7726	7728	7730	7823	7824	7826	7831	7832
7844		7844	7846	7847	7847	7847	7858	7859	7865	7914	7930
7931		7932	7933	7933	7933	7934	7935	7938	7939	7940	7940
7941		7942	7943	7943	7943	7944	7945	7946	7946	7946	7948
7949		7949	7949	7950	7951	7952	7952	7952	7953	7954	7956
8245		8293	8294	8294	8301	8302	8302	8306	8307	8308	8309
8387		8388	8391	8526	8542	8579	8623	8624	8625	8627	8740
8794		8795	8795	8796	8825	8827	8831	8831	8834	8834	8834
1810	#	2268	2339	2352	2361	2365	2393	2404	2423	3548	3572
3613		3617	3677	3722	3759	3844	4001	4764	4770	5156	5205
5931		6351	6594	6622	6898	6912	6968	6976	6983	7104	7222
7230		7234	7238	7277	7317	7443	7455	7491	7497	7587	7668
7716		7718	7826	7865	8309	8391	8627				
1809	#	2268	2339	2352	2361	2365	2393	2404	2423	2494	2526
2537		2540	2590	2787	3406	3548	3572	3587	3613	3617	3677
3700		3722	3739	3745	3759	3816	3844	3966	3968	3974	4001
4737		4764	4770	4892	5133	5156	5159	5205	5248	5912	5931
6545		6591	6594	6622	6898	6912	6968	6976	6983	7104	7222
7230		7234	7238	7277	7308	7310	7317	7371	7443	7455	7491

MEM READ

MEM WAIT

	7587	7621	7668	7713	7716	7718	7728	7730	7826	7846	7865	7931
	7932	7934	7935	7938	7939	7941	7942	7944	7945	7947	7948	7950
	7951	7953	7954	7956	8309	8391	8627	8796	8825	8827		
MEM WRITE	1811 #	2494	2526	2532	2537	2540	2590	2787	3406	3694	3700	3739
	3745	3816	3966	3968	3974	4040	4737	4892	5133	5159	5248	6545
	6591	7308	7310	7371	7621	7728	7730	7846	7931	7932	7934	7935
	7938	7939	7941	7942	7944	7945	7947	7948	7950	7951	7953	7954
	7956	8796	8825	8827								
MEM_Q	1818 #	5134	5160	5248	6545							
MEM_[]	1817 #	2495	2527	2533	2538	2541	2591	2787	3406	3695	3701	3740
	3746	3817	3967	3969	3975	4041	4737	4892	6591	7308	7310	7371
	7621	7728	7730	7846	7931	7932	7934	7935	7938	7939	7941	7942
	7944	7945	7947	7948	7950	7951	7953	7954	7956	8797	8825	8827
MUL DISP	2087 #	4268	4276	4286	4294	5580	5606	5608	5725			
MUL FINAL	4246 #	4272	4280	4290	4298							
MUL STEP	4245 #	4268	4276	4286	4294							
NEXT INST	2067 #	2522	3043	3089	3117	3429	3711	3787	3794	4482	4485	5440
	5446	5805	8202									
NEXT INST FETCH	2068 #	2228	2229	2231	2232	2292	2298	3326	3338	3453	3454	3457
	3458	3525	3526	3552	3594	4218	4693	7352	7623			
NEXT [] PHYSICAL WRI	1840 #	3979	4026	7847	7933	7940	7943	7946	7949	7952	8834	
NO DIVIDE	2029 #	4339	4374	4405	4491	4998						
NORM DISP	2093 #	5300	5329	5406	5481	5483	5493	5497	5499	5501	5502	5505
	5527	5678	5732	5733	5743	5745	5765	5775	5777	5782	5783	5810
	5839											
ONES	1968 #	4820	4823	4896	4899	5983						
PAGE FAIL TRAP	2084 #	8325	8345	8413	8422	8439	8576	8601	8606	8630	8694	
PI DISP	2092 #											
PXCT BLT DEST	1794 #	5124	5132	5146	5151	5247						
PXCT BLT SRC	1798 #	5096	5169	5253								
PXCT BYTE DATA	1796 #	4751	4755	6893	6894							
PXCT BYTE PTR EA	1795 #	4759	4763									
PXCT DATA	1793 #	2321	2363	2415	2418	2786	2790	3731	3736			
PXCT EA	1792 #	2330	2335	3570								
PXCT EXTEND EA	1799 #	5928	5929									
PXCT STACK WORD	1797 #	3683	3720	3752								
Q-[]	1667 #	4428										
Q.AND.NOT.[]	1668 #											
Q_#	1683 #	4973	5017									
Q_-1	1675 #											
Q_-AC[]	1676 #	4610	4615	4620								
Q_-Q	1677 #	4398	4475	4516	4517	4534	5482	5484	5486	5488	5771	
Q_-[]	1674 #	5707										
Q_.NOT.AC[]	1673 #	4614	4619									
Q_.NOT.Q	1682 #	5770										
Q_O	1684 #	2981	4819	4895	5297	5368	5379	5391	5395	5397	5398	5411
	5424	5717	5981									
Q_O XWD []	1685 #											
Q_AC	1678 #	4119	4317									
Q_AC[]	1679 #	2383	3087	4212	4214	4327	6541					
Q_AC[].AND.MASK	1680 #											
Q_AC[].AND.[]	1681 #	4411	4433	4440								
Q_MEM	1828 #	5157	5206	7718								
Q_Q*.5	1690 #	4227	4261	4434	5208	5228	5229	5230	5244	5632	5722	
Q_Q*2	1691 #	5223	5224									

Q_Q+.25	1686 #	5583	5730										
Q_Q+1	1687 #												
Q_Q+AC	1689 #	4993											
Q_Q+[]	1696 #	4436	4468	5565.	5573	5674	5677						
Q_Q-1	1688 #	5033											
Q_Q-WORK[]	1775 #	4976											
Q_Q.AND.#	1693 #	3113	4511	5217	5237	5324	5633						
Q_Q.AND.NOT.[]	1695 #	5719											
Q_Q.AND.[]	1694 #	4483	4902										
Q_Q.OR.#	1692 #	3115	4396										
Q_WORK[]	1757 #	5040	6140										
Q_[]	1669 #	4145	4184	4204	4224	4438	4464	4964	5321	5556	5568	5628	
	5643	5656	5662	5712									
Q_[]+[]	1671 #	4465											
Q_[]-[]	1670 #	5100											
Q_[]AND.Q	1697 #	4559	5754	5789									
Q_[]AND.[]	1672 #	4386											
Q_[]OR.Q	1698 #	5240	5588										
RAM []	1778 #	2183	2185										
READ Q	1708 #	5447	5489	5513	5762	5807							
READ XR	1706 #	3556											
READ []	1707 #	2376	2396	2572	2736	2743	2750	2757	3068	3084	3140	3285	
	3329	3332	3335	3341	3344	3347	3560	3608	3619	3659	3963	3997	
	4003	4034	4122	4328	4334	4348	4353	4355	4360	4427	4429	4446	
	4462	4471	4532	4539	4560	4699	4703	4718	4721	4818	4884	4900	
	5021	5285	5289	5372	5374	5406	5410	5443	5483	5502	5570	5669	
	5678	5727	5745	5783	5919	5937	5980	6014	6115	6118	6124	6137	
	6148	6164	6340	6376	6427	6434	6460	6474	6479	6501	6505	6557	
	6643	6651	6746	6768	6800	6850	6852	6874	7067	7069	7070	7075	
	7079	7156	7269	7304	7383	7389	7396	7663	7920	7930	8275	8374	
	8538	8596	8756	8761									
RETURN []	2091 #	2457	2458	3559	3563	4024	4027	4230	4273	4281	4291	4299	
	4493	4515	4516	4518	4602	4629	4731	4733	4737	4772	4805	4824	
	4892	5106	5243	5244	5526	5528	5529	5618	5813	5820	5821	5836	
	6225	6235	6240	6243	6281	6282	6432	6438	6609	6632	6697	6728	
	6740	6771	6787	6804	6813	6815	6841	6876	6909	6915	6920	6960	
	7199	7265	7289	7364	7408	7516	7519	7591	7598	7604	7661	7669	
	7673	7676	7702	7714	7716	7718	7720	7722	7724	7726	7728	7730	
	7732	7923	7927	7961	8548	8570	8575	8581	8612	8615	8616	8629	
	8638	8641	8642	8746	8834								
ROT	1964 #	3007	3012										
ROTC	1967 #	3143	3147										
SCAD DISP	2090 #	4692	4697	4711	4787	5283	5426	5428	5981	5983	6034	6844	
	6886	7688											
SC_-1	1907 #												
SC_-2	1908 #												
SC_-SHIFT	1871 #												
SC_-SHIFT-1	1872 #	3068	3140										
SC_-SHIFT-2	1873 #												
SC_O	1906 #	6650	6661										
SC_1	1905 #												
SC_11.	1896 #												
SC_14.	1895 #												
SC_19.	1894 #	2175											
SC_2	1904 #	6471											

SC_20.	1893 #	6294																			
SC_22.	1892 #																				
SC_24.	1891 #																				
SC_26.	1890 #	5717																			
SC_27.	1889 #	5321	5368																		
SC_28.	1888 #																				
SC_3	1903 #	4954																			
SC_34.	1887 #	4337	4434	4442	4965	4978	4994														
SC_35.	1886 #	4119	4147	4184	4204	4224	5039	5639	5643	5656	5662	5718									
SC_36.	1885 #																				
SC_4	1902.#																				
SC_5	1901 #	7595	7627																		
SC_6	1900 #	5402	5628	6464	7147	7178	7188	7195													
SC_7	1899 #	6462	7129	8262	8313	8361	8497														
SC_8.	1898 #																				
SC_9.	1897 #	4938	5054																		
SC_EXP	1878 #	2397	2412	5287	5559																
SC_FE	1881 #	2428	5646																		
SC_FE+S#	1880 #	3094	5428																		
SC_S#	1882 #	2175	4119	4147	4184	4204	4224	4337	4434	4442	4938	4954									
		4965	4978	4994	5039	5054	5321	5368	5402	5628	5639	5643	5656	5662	5718						
		5662	5717	5718	6294	6462	6464	6471	6650	6661	7129	7147	7178								
		7188	7195	7407	7595	7627	7681	7697	8262	8313	8361	8497									
SC_S#-FE	1879 #	3097	5430																		
SC_SC-1	1867 #	7688																			
SC_SC-EXP	1874 #																				
SC_SC-EXP-1	1875 #	5283	5549																		
SC_SC-FE	1877 #																				
SC_SC-FE-1	1876 #	5288	5569																		
SC_SHIFT	1868 #	3085	5410																		
SC_SHIFT-1	1869 #	2378																			
SC_SHIFT-2	1870 #																				
SET APR ENABLES	1998 #	2166	7071	7156																	
SET AROV	2007 #	4132	4133	4157	4219	5427															
SET FL NO DIVIDE	2010 #	5350	5362	5715																	
SET FOV	2008 #																				
SET FPD	2013 #	4232	4746	6389																	
SET HALT	1993 #	7849	7915																		
SET NO DIVIDE	2009 #	4339	4374	4405	4491	4998	5350	5362	5715												
SET PDL OV	2016 #	3702	3758	3788	3792																
SET P TO 36-S	4673 #	4732	6043	6846	6888																
SET TRAP1	2017 #																				
SHIFT	1865 #	2973	2985	3007	3012	5243	5244														
SKIP AC REF	2061 #																				
SKIP AD.EQ.O	2045 #	3017	3332	3344	3435	3447	4122	4127	4329	4331	4335	4349									
	4394	4423	4453	4534	4612	4617	4622	4638	4645	4963	4992	5346									
	5372	5374	5447	5489	5513	5669	5720	5762	5807	6137	6190	6266									
	6269	6558	6595	6616	6739	7296	8289	8292	8376	8379											
SKIP AD.LE.O	2046 #	3335	3347	3438	3450	4388	5022	5366	5690	6337	7262	7919									
SKIP ADL.EQ.O	2057 #	2188	3987	3992	5389	5394	5508	5673	5676	5744	5747	5750									
	5753	5813	5913	5956	6095	6097	6179	6199	6383	6399	6453	6488									
	6491	6523	6987	7111	7118	7154	7457	7458	7649	7651	7665	8249									
	8315	8322	8325	8340	8345	8394	8413	8417	8422	8433	8437	8439									
	8459	8512	8531	8533	8571	8576	8598	8601	8603	8606	8609	8630									
	8640	8694	8712	8753																	

SKIP ADL.LE.O	2048 #	7658											
SKIP ADL.EQ.O	2058 #	3086	3592	3607	5120	6324	6397	6752	7029	7032	7057	7059	
	7062 #	7064	7066	7332	7334	7338	7340	7342	7344	7346	7348	7445	
	7498	7518	7590	7617	8200	8464	8492	8698	8701	8706			
SKIP CRYO	2053 #	3685	3728	3757	5358	5482	5484	5486	5488				
SKIP CRY1	2054 #	4085	4097	5798	6214	6220	6233	6424	6435				
SKIP CRY2	2055 #	5771	5831										
SKIP DPO	2043 #	2398	2413	2572	2750	2757	3040	3112	3329	3341	3432	3444	
	3774	3777	3782	4104	4129	4151	4154	4209	4217	4318	4354	4356	
	4361	4366	4370	4392	4409	4421	4428	4443	4462	4480	4532	4539	
	4554	4995	5139	5163	5249	5285	5289	5318	5348	5356	5390	5412	
	5443	5550	5652	5699	5703	5714	5727	5964	5977	5994	6007	6014	
	6032	6074	6076	6098	6115	6118	6124	6164	6256	6261	6298	6307	
	6340	6357	6429	6479	6505	6696	6726	6768	6822	6941	7106	7251	
	8276	8374	8597										
SKIP DP18	2044 #	2377	2736	2743	5937	6184	6388	6477	6800	6801	7484		
SKIP EXECUTE	2062 #	7853											
SKIP FPD	2050 #	4171	4692	4697	4711	6288							
SKIP IF ACO	2042 #	2484	3388	4689									
SKIP IO LEGAL	2052 #	2434	3537	3540									
SKIP IRPT	2059 #	5143	6034	8213	8408	8626	8734	8765	8816				
SKIP JFCL	2056 #	3599											
SKIP KERNEL	2051 #	3530	3534	3535	3542	3651	6966						
SKIP NO CST	8005 #	8358	8483	8567									
SKIP -1MS	2060 #	6029											
SKIP-COMP DISP	2078 #	3370	3389	3390	6113								
SPEC MEM READ	1812 #	2410											
SPEC MEM WRITE	1813 #												
START NO TEST WRITE	1805 #	8579											
START READ	1803 #	2228	2229	2231	2232	2251	2291	2297	2302	2308	2331	2335	
	2364	2405	2416	2419	2430	3020	3326	3338	3453	3454	3457	3458	
	3525	3526	3527	3532	3536	3539	3552	3566	3569	3594	3616	3636	
	3709	3719	3753	3843	4000	4044	4218	4693	4702	4750	4754	4758	
	4762	5095	5170	5253	5928	5929	5953	6100	6350	6377	6450	6455	
	6511	6631	6747	6893	6894	6895	6896	6971	6979	7103	7221	7225	
	7229	7233	7237	7276	7280	7316	7352	7505	7623	7646	7667	7824	
	7859	8307	8387	8624									
START WRITE	1804 #	2589	2785	2791	3405	3630	3633	3641	3682	3732	3737	3815	
	3826	3979	4026	4036	4730	4736	4891	5123	5131	5147	5152	5246	
	6543	6589	7305	7309	7370	7470	7832	7847	7933	7940	7943	7946	
	7949	7952	8834										
STATE_[]	2034 #	5106	5972	5998	6001	6023	6071	6077	6085	6116	6123	6142	
	6170	6352	6379	6503	6560	6564	6624	8201	8771	8773	8775	8777	
	8779	8781	8785										
STEP SC	1864 #	2185	2989	2991	3067	3069	3071	3075	3101	3108	3139	3141	
	3143	3147	4250	4262	4268	4276	4286	4294	4549	4552	4592	4941	
	4960	5060	5297	5299	5405	5429	5432	5435	5439	5603	5606	5613	
	5631	6323	6467	6473	6668	7133	7149	7180	7197	7387	7400	7602	
	7633	7701	8265	8327	8330	8364	8503						
SWEEP	1990 #	7395	7396	7399									
TAKE INTERRUPT	2085 #	8793											
TEST DISP	2077 #	3271	3277	7533									
TL []	1712 #	3987	3992	5508	5673	5676	5744	5747	5750	5753	5813	5913	
	5956	6095	6097	6179	6199	6383	6399	6453	6488	6491	7649	7651	
	7665	8249	8315	8322	8325	8340	8345	8394	8413	8417	8422	8433	

	8437	8439	8459	8512	8531	8533	8571	8576	8598	8601	8603	8606
	8609	8630	8640	8694	8712	8753						
TR []	1711 #	3592	3607	6324	6397	6752	7057	7059	7062	7064	7066	7332
	7334	7338	7340	7342	7344	7346	7348	7590	7617	8200	8464	8492
	8698	8701	8706									
TURN OFF PXCT	1987 #	2240	2243	2246	2258	2261	2264	2278				
TXXX TEST	2063 #	3285										
UNHALT	1997 #	7852										
UPDATE USER	2020 #	3558	3562	3620								
UUD	2082 #	2435	3529	3538	3541	3543	3544	3545	3578	3583	3586	3606
	3627	3640	3919	3921	3923	3925	3927	3929	3931	3933	3908	6991
	7000	7001	7002	7006	7007	7008	7009	7014	7015	7016	7019	7020
	7021	7022	7023	7024	7025	7026	7218	7240	7335	7794	7803	7805
	7807	7809	7811	7813								
VMA	1852 #											
VMA EXTENDED	1790 #											
VMA PHYSICAL	1787 #	3965	3979	3999	4026	6972	6980	7454	7496	7844	7847	7930
	7933	7940	7943	7946	7949	7952	8294	8302	8308	8388	8625	8795
	8831	8834										
VMA PHYSICAL READ	1789 #	7454	7496	8294	8302	8831						
VMA PHYSICAL WRITE	1788 #	3965	7844	7930	8795							
VMA_[]	1832 #	2228	2232	2250	2307	2334	2343	2430	2788	3326	3453	3458
	3532	3568	3630	3641	3730	3735	3814	3842	4218	4693	4702	4753
	4761	5094	5122	5130	5246	5929	6455	6511	6589	6631	6894	6896
	7102	7352	7370	7376	7454	7623	7667	7823	7831	7858	7959	8306
	8526	8542	8623	8794								
VMA_[] WITH FLAGS	1833 #	7726										
VMA_[]+1	1835 #	2229	2231	2276	2363	3338	3632	4042	5144	5149	5167	5252
	6100	6450	7279	7309								
VMA_[]+XR	1837 #	4749	4757	6893	6895							
VMA_[]+[]	1838 #	8293	8301	8386								
VMA_[]-1	1836 #	3527	3536	3539	7645							
VMA_[]_OR.[] WITH FL	1834 #	7585	7620									
WORK[]	1853 #	6132	6354	6553	6620	6698	6729	6732				
WORK[]_NOT.[]	1763 #	6174										
WORK[]_O	1758 #	2177	7403	7404								
WORK[]_1	1759 #	2176										
WORK[]_Q	1756 #	4972	6104	6143								
WORK[]_[]	1760 #	2160	2168	2170	2171	4690	4949	4962	4982	5014	5922	5935
	5938	5968	6030	6056	6057	6058	6073	6120	6156	6295	6297	6359
	6372	6378	6577	6725	6914	6965	7072	7157	7223	7227	7231	7235
	7239	7253	7264	7285	7288	7318	7319	7686	7692	7694	7839	7911
	7926	7936	8198	8208	8209	8211	8212	8219	8229	8231	8233	8235
	8239	8240	8381	8383	8404	8725						
WORK[]_[] CLR LH	1761 #	3663	7652									
WORK[]_[]-1	1762 #	6353										
WORK[]_[]_AND.[]	1764 #	5985										
WRITE TEST	1802 #	2589	2785	2791	3405	3630	3633	3641	3682	3732	3737	3815
	3826	3979	4026	4036	4730	4736	4891	5123	5131	5147	5152	5246
	6543	6589	7305	7309	7370	7470	7832	7847	7933	7940	7943	7946
	7949	7952	8834									
XR	1851 #											
[] LEFT -1	1653 #	2740										
[] LEFT_O	1651 #	2738										
[] RIGHT -1	1654 #	2761										

Cross Reference Listing

[] RIGHT_O	1652 #	2759											
[]+[]	1518 #	2783	4229	4369									
[]-#	1520 #	6183	6476										
[]-[]	1519 #	4365	4421	4422	5119	5137	5161	5249	5358	5366	5714	6098	
[] .AND. #	1521 #	5389	5393	6522	6985	7116	7154						
[] .AND. NOT. WORK[]	1765 #												
[] .AND. NOT. []	1524 #	4126											
[] .AND. Q	1522 #	4393											
[] .AND. WORK[]	1766 #	6738											
[] .AND. []	1523 #	4154	7029	7032	7518								
[] .OR. []	1525 #												
[] .XOR. #	1526 #	2188	7457	7458									
[] .XOR. []	1527 #	7295											
[]_#	1529 #	2149	2153	2178	2179	2219	2745	2747	2752	2754	4538	4946	
	4969	4979	4983	5012	5055	5106	5940	5972	5998	6001	6023	6071	
	6077	6085	6116	6123	6142	6170	6352	6379	6503	6560	6564	6624	
	7036	7038	7379	7381	7392	7394	7473	7515	8201	8771	8773	8775	
	8777	8779	8781	8785									
[]_#-[]	1528 #												
[]_#-[]*2	1615 #	5050											
[]_#-[]*.5	1616 #												
[]_#-[]-.25)*.5 LONG	1617 #	4399	5708										
[]_#-[]-.25)*2 LONG	1618 #	4476											
[]_#(AC[] .AND. [])*.5	1613 #	4175	4177	5547	5638	5693							
[]_#(MEM. AND. [])*.5	1827 #	2424											
[]_#(Q+1)*.5	1614 #	5830											
[]_#([]+#)*2	1628 #												
[]_#([]+1)*2	1629 #	6642											
[]_#([]+[])*.5 LONG	1630 #	5577											
[]_#([]+[])*2 LONG	1631 #	4583											
[]_#([]+[]+.25)*.5 LO	1634 #												
[]_#([]-[])*.5 LONG	1632 #												
[]_#([]-[])*2 LONG	1633 #	4580											
[]_#([] .AND. #)*.5	1620 #	4943	4950	7499									
[]_#([] .AND. #)*2	1621 #												
[]_#([] .AND. NOT. #)*.5	1622 #												
[]_#([] .AND. NOT. #)*2	1623 #												
[]_#([] .AND. [])*.5	1624 #	4169											
[]_#([] .AND. [])*2	1625 #												
[]_#([] .OR. #)*.5	1626 #												
[]_#([] .OR. #)*2	1627 #												
[]_# +SIGN	1657 #	2422	2457	5290	5294	5319	5352	5413					
[]_# +SIGN*.5	1662 #	5551	5653	5704									
[]_# -1	1530 #	4321	6389	7361									
[]_# -2	1531 #												
[]_# -AC	1546 #	4625											
[]_# -AC[]	1547 #	6268											
[]_# -Q	1532 #												
[]_# -Q*.5	1534 #	5377											
[]_# -Q*2	1533 #	4449											
[]_# -SIGN	1658 #	2420	2458	5291	5295	5320	5353	5414					
[]_# -SIGN*.5	1663 #	5553	5654	5706									
[]_# -WORK[]	1773 #	5996	6939	6953									
[]_# -[]	1535 #	2577	2947	4518	4536	4543	4637	4641	4645	4647	5280	5355	
	5390	5499	5515	5544	5696	5776	5797	6265					

Produced on Advanced Information Services Electronic Laser Printer. PKO/IES6, DTN: 223-7881

[]_PC WITH FLAGS	1573 #	3689	3806	3812	4216	6964	7471						
[]_Q	1574 #	4124	4132	4352	4407	4424	4430	4439	5607	5614	5634	5657	
	5716	5718											
[]_Q*.5	1575 #	4187	5371	5376	5646								
[]_Q*2	1576 #												
[]_Q*2 LONG	1577 #	4435	4445										
[]_Q+1	1578 #	5101											
[]_RAM	1579 #												
[]_TIME	1580 #	7292	7293	7294									
[]_VMA	1581 #	7682	7925	8195	8210	8726							
[]_VMA FLAGS	1843 #	7440	7487	7579	7582	7618	7629	7635	7861				
[]_VMA IO READ	1844 #	7440	7487	7579	7582	7861							
[]_VMA IO WRITE	1845 #	7618	7629	7635									
[]_WORK[]	1771 #	4967	4977	5176	6060	6061	6062	6063	6127	6338	6348	6381	
	6382	6403	6579	6615	6692	6812	6823	6930	7050	7084	7152	7209	
	7211	7213	7215	7217	7246	7255	7260	7266	7298	7302	7323	7655	
	7660	7913	7918	7922	7937	7955	7957	7958	7960	8199	8288	8291	
	8375	8378	8452	8454	8499	8539	8540	8541	8547	8736	8742	8751	
	8755	8767											
[]_WORK[]+1	1774 #	5992	6025	6080	6178	6356	6819	6825					
[]_XR	1582 #	7658											
[]_[]	1583 #	2295	3019	3454	3457	3525	3526	3552	3579	3594	3653	3707	
	3825	3830	4118	4146	4170	4186	4316	4325	4340	4346	4375	4730	
	5038	5286	5287	5288	5326	5346	5354	5563	5564	5571	5572	5645	
	5687	5690	5732	5733	5974	5994	6040	6075	6335	6346	6354	6421	
	6521	6754	6845	6868	6887	6944	7010	7011	7174	7187	7194	7199	
	7297	7329	7475	7557	7559	7566	7591	8261	8349	8496	8520	8817	
[]_[] SWAP	1584 #	2389	2569	2716	2718	2721	2724	2727	2730	3269	3274	3593	
	3770	3827	3841	3959	4811	4937	4953	5053	5093	5102	5175	6400	
	6457	6462	6463	6757	7085	7337	7350	7360	8402				
[]_[] XWD O	1585 #	6155	6270	7708	8225	8227	8241						
[]_[]*.5	1586 #	2152	2426	2427	2984	3005	3006	3010	3093	3096	4168	4225	
	4387	4415	4546	4547	4548	4554	4815	4891	4940	4959	5361	5369	
	5436	5485	5487	5503	5504	5523	5524	5529	5639	5672	5841	6464	
	6467	6471	6473	6553	6661	6668	6731	7178	7180	7188	7195	7197	
	7502	7520	7595	7601	8264								
[]_[]*.5 LONG	1587 #	2385	3071	3099	3143	4120	4148	4227	4248	4261	4389	4391	
	4403	4419	4434	4441	4493	4541	5041	5208	5228	5229	5230	5244	
	5299	5322	5327	5402	5404	5434	5602	5605	5613	5616	5618	5632	
	5667	5722	5748	5751	5785	5787	5833						
[]_[]*2	1588 #	2150	2972	2995	3008	3011	3038	4123	4150	4206	5059	5243	
	5356	5364	5439	6202	6748	7132	7147	7149	7447	7483	7627	7633	
	7724	8326	8329	8363	8502								
[]_[]*2 LONG	1589 #	2991	3072	3075	3076	3077	3102	3106	3109	3144	3147	3148	
	4473	4553	4556	4558	4904	5223	5224	5481	5493	5501	5505	5579	
	5584	5586	5598	5600	5601	5630	5671	5675	5724	5741	5763	5780	
	5808												
[]_[]*4	1590 #	6458											
[]_[]+#	1591 #	3725	3754	6417	6426	7448	7493	8296	8304	8790			
[]_[]+.25	1592 #	5837											
[]_[]+0	1593 #												
[]_[]+1	1594 #	2180	2184	2414	2417	3404	3614	3635	4736	5142	5166	5251	
	5448	5951	6052	6070	6099	6242	6310	6322	6329	6349	6370	6384	
	6439	6502	6526	6620	6847	6889	7256	7722	8769	8830			
[]_[]+1000001	1595 #	3818	3833	5116									

.NOT.[]	1517 #	4330	7363	7438								
2T	1855 #	4409	4422	5289	5559	6133	6203	6267	6555	6692	6737	7302
	7659											
3T	1856 #	2188	2413	2423	2578	3111	3285	3370	3404	3420	3510	3600
	3684	3719	3727	3752	3756	3773	3778	3783	3820	3834	4058	4072
	4175	4177	4196	4230	4367	4371	4421	4427	4428	4429	4446	4450
	4471	4479	4560	4625	4642	4700	4715	4786	4804	4900	4944	4951
	5031	5047	5051	5117	5119	5138	5162	5249	5283	5317	5358	5389
	5390	5393	5425	5428	5482	5484	5486	5488	5547	5548	5551	5553
	5652	5653	5654	5698	5702	5704	5706	5714	5797	5803	5963	5981
	5983	6031	6035	6098	6205	6219	6233	6265	6268	6353	6356	6388
	6389	6417	6423	6426	6435	6460	6474	6501	6521	6522	6644	6651
	6695	6754	6844	6851	6872	6886	6890	6893	6895	6931	6986	7105
	7111	7117	7154	7419	7444	7448	7456	7457	7458	7493	7497	7499
	7666	8296	8304	8526	8543	8761	8791					
4T	1857 #	3016	3487	3501	4084	4089	4096	4099	4101	4991	4996	5977
	6007	6184	6189	6214	6255	6428	6476	6616	6739	6940	7919	8216
	8289	8292	8376	8379								
5T	1858 #	7688	7705									
7-BIT DPB	4881 #	4885	4886	4887	4888	4889						
7-BIT LDB	4783 #	4791	4792	4793	4794	4795						
(D) MACRO%												
AC	2124 #	2547	2548	2552	2553	2557	2558	2562	2563	2584	2600	2601
	2605	2606	2610	2611	2615	2616	2620	2621	2625	2626	2630	2631
	2635	2636	2642	2643	2647	2648	2652	2653	2657	2658	2662	2663
	2667	2668	2672	2673	2677	2678	2767	2797	2798	2807	2808	2817
	2818	2827	2828	2832	2833	2849	2850	2859	2860	2869	2870	2879
	2880	2889	2890	2899	2900	2910	2911	2920	2921	2930	2931	2940
	2941	4050	4051	4064	4065	4112	4113	4682	8188			
AC DISP	2138 #	3516	6996	6997	7204							
B	2126 #	2800	2810	2820	2830	2835	2852	2862	2872	2882	2892	2902
	2913	2923	2933	2943	4053	4067	4115					
DAC	2128 #	2766	4079	4080	4137	4138	4163	4304	4305	4309	4310	4381
	5623	5684										
DBL AC	2113 #	2775										
DBL B	2129 #	4140	4307	4312								
DBL FL-R	2119 #	5534	5535	5623	5684							
DBL R	2112 #	2766	2767	4079	4080	4163	4381					
FL-AC	2130 #	5260	5263	5264	5268	5271	5272	5305	5309	5310	5334	5338
	5339	5384	5385	5419	5420							
FL-BOTH	2132 #	5262	5266	5270	5274	5307	5312	5336	5341			
FL-I	2118 #	5264	5272	5310	5339							
FL-MEM	2131 #	5261	5265	5269	5273	5306	5311	5335	5340			
FL-R	2116 #	5260	5263	5268	5271	5305	5309	5334	5338	5419	5420	
FL-RW	2117 #	5261	5262	5265	5266	5269	5270	5273	5274	5306	5307	5311
	5312	5335	5336	5340	5341							
I	2104 #	2843	2955	3175	3176	3177	3178	3179	3180	3181	3182	3184
	3185	3193	3194	3195	3196	3197	3198	3199	3200	3211	3212	3213
	3214	3215	3216	3217	3218	3228	3229	3230	3231	3232	3233	3234
	3235	3350	3351	3352	3353	3354	3355	3356	3357	3462	3463	3464
	3465	3466	3467	3468	3469	3477	3478	3479	3480	3481	3482	3483
	3491	3492	3493	3494	3495	3496	3497	3504	3505	3516	3518	3669
	3671	3672	3799	3800	3801	3802	3852	3853	3854	3855	3856	3857
	3858	3859	3863	3864	3865	3866	3867	3868	3869	3870	3871	3872
	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884

Produced on Advanced Information Services Electronic Laser Printer, PKO/ISS, DTN: 223-7881

		3885	3886	3887	3888	3889	3890	3891	3892	3893	3894	3898	3899
		3900	3901	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914
		3915	5109	5385	5849	5850	5851	5852	5853	5854	5855	5857	5858
		5859	5860	5862	5863	5864	5865	5867	5868	5869	5870	5871	5872
		5873	5874	5908	7737	7738	7739	7741	7742	7744	7745	7747	7748
		7749	7750	7751	7752	7753	7754	7756	7757	7758	7759	7760	7761
		7762	7763	7765	7766	7767	7768	7769	7770	7771	7772	7774	7775
		7776	7777	7778	7779	7780	7781	7783	7784	7785	7786	7787	7788
		7789	7790										
I-PF		2105 #	2548	2553	2558	2563	2601	2606	2611	2616	2621	2626	2631
		2636	2643	2648	2653	2658	2663	2668	2673	2678	2797	2798	2808
		2818	2828	2833	2850	2860	2870	2880	2889	2890	2900	2911	2921
		2931	2940	2941	2956	3476	3490	3766	4051	4065	4113	4138	4305
		4310											
IOT		2120 #	6996	6997	7204	7525	7526	7527	7528	7536	7537	7538	7539
		7550	7551	7552	7553	7818	7819	8188					
IR		2111 #	3670										
IW		2110 #	2799	2800	2891	2892	2942	2943					
M		2125 #	2549	2554	2559	2564	2602	2607	2612	2617	2622	2627	2632
		2637	2644	2649	2654	2659	2664	2669	2674	2679	2799	2809	2819
		2829	2834	2844	2845	2851	2861	2871	2881	2891	2901	2912	2922
		2932	2942	4052	4066	4114	4139	4306	4311	7204			
R		2106 #	2584	2842	3186	3187	3188	3189	3190	3191	3202	3203	3204
		3205	3206	3207	3208	3209	3220	3221	3222	3223	3224	3225	3226
		3227	3237	3238	3239	3240	3241	3242	3243	3244	3359	3360	3361
		3362	3363	3364	3365	3366	3376	3377	3378	3379	3380	3381	3382
		3383	3647	4682	4683	4684	4685	4686	5384				
R-PF		2107 #	2547	2552	2557	2562	2600	2605	2610	2615	2620	2625	2630
		2635	2642	2647	2652	2657	2662	2667	2672	2677	2807	2817	2827
		2832	2849	2859	2869	2879	2899	2910	2920	2930	4050	4064	4112
		4137	4304	4309									
ROUND		2133 #	5263	5264	5265	5266	5271	5272	5273	5274	5309	5310	5311
		5312	5338	5339	5340	5341	5384	5420					
RW		2109 #	2550	2555	2560	2565	2602	2603	2607	2608	2613	2618	2623
		2628	2633	2638	2644	2645	2649	2650	2655	2660	2665	2670	2675
		2680	2809	2810	2819	2820	2829	2830	2834	2835	2851	2852	2861
		2862	2871	2872	2881	2882	2901	2902	2912	2913	2922	2923	2932
		2933	3393	3394	3395	3396	3397	3398	3399	3400	3409	3410	3411
		3412	3413	3414	3415	3416	4052	4053	4066	4067	4114	4115	4139
		4140	4306	4307	4311	4312							
S		2127 #	2550	2555	2560	2565	2603	2608	2613	2618	2623	2628	2633
		2638	2645	2650	2655	2660	2665	2670	2675	2680			
		2114 #	2952	2953	2954								
SH		2115 #	2957	2958									
SHC		3310 #	3350	3359	3376	3393	3409	3462	3476	3490			
SJC-		3314 #	3354	3363	3380	3397	3413	3466	3480	3494			
SJCA		3312 #	3352	3361	3378	3395	3411	3464	3478	3492	5850		
SJCE		3317 #	3357	3366	3383	3400	3416	3469	3483	3497	5855		
SJCG		3315 #	3355	3364	3381	3398	3414	3467	3481	3495	3504	5853	
SJCGE		3311 #	3351	3360	3377	3394	3410	3463	3477	3491	3505	5849	
SJCL		3313 #	3353	3362	3379	3396	3412	3465	3479	3493	5851		
SJCLE		3316 #	3356	3365	3382	3399	3415	3468	3482	3496	5854		
SJCN		3166 #	3211	3212	3220	3221							
TC-		3168 #	3215	3216	3224	3225							
TCA		3167 #	3213	3214	3222	3223							
TCE													

Produced on Advanced Information Services Electronic Laser Printer. PGO/JES6. DTN: 222-7981

	8623	8624	8625	8627	8740	8746	8794	8795	8795	8795	8796	8818
	8825	8827	8831	8831	8831	8834	8834	8834				
(U) MICROCODE OPTION(INH^V@^H	1240 #											
OPT	1242 #	7039										
(U) MICROCODE OPTION(KIP^W@^H	1264 #											
OPT	1268 #	7043										
(U) MICROCODE OPTION(KLP^W#^H	1270 #											
OPT	1272 #	7044										
(U) MICROCODE OPTION(NOC^V#^H	1246 #											
OPT	1250 #	7040										
(U) MICROCODE OPTION(NON^W	1252 #											
OPT	1254 #	7041										
(U) MICROCODE OPTION(UBA^W ^H	1258 #											
OPT	1260 #	7042										
(U) MICROCODE OPTIONS	1233 #											
(U) MICROCODE VERSION	1277 #											
UCV	1278 #	7045										
(D) MODE	1338 #											
(U) MULTI PREC	965 #	4400	4452	4467	4470	4477	4591	5545	5578	5698	5709	
(U) MULTI SHIFT	967 #	2973	2985	3007	3012	4817	4822	4898	4906	5243	5244	
(U) PHYSICAL	1134 #	3965	3979	3999	4026	6972	6980	7382	7383	7386	7395	7396
	7399	7440	7454	7487	7496	7579	7582	7618	7629	7635	7844	7847
	7861	7930	7933	7940	7943	7946	7949	7952	8294	8302	8308	8388
	8625	8795	8831	8834								
(U) PI.CLR	1193 #	7332										
(U) PI.CO1	1182 #											
(U) PI.CO2	1183 #											
(U) PI.DIR	1192 #	7338										
(U) PI.IP1	1174 #	7515										
(U) PI.IP2	1175 #											
(U) PI.IP3	1176 #											
(U) PI.IP4	1177 #											
(U) PI.IP5	1178 #											
(U) PI.IP6	1179 #											
(U) PI.IP7	1180 #											
(U) PI.MBZ	1191 #	7334										
(U) PI.ON	1181 #	7343	7345									
(U) PI.REQ	1194 #	7340										
(U) PI.SC1	1199 #											
(U) PI.SC2	1200 #											
(U) PI.SC3	1201 #											
(U) PI.SC4	1202 #											
(U) PI.SC5	1203 #											
(U) PI.SC6	1204 #											
(U) PI.SC7	1205 #											
(U) PI.TCF	1196 #	7348										
(U) PI.TCN	1195 #	7346										
(U) PI.TSF	1197 #	7344										
(U) PI.TSN	1198 #	7342										
(U) PI.ZER	1173 #											
(U) PXCT	1138 #											
BIS-DST-EA	1144 #	6887	6889	6895	6896	6898						
BIS-SRC-EA	1142 #											
CURRENT	1139 #	2228	2229	2231	2232	2251	2291	2297	2405	2430	3020	3326
	3338	3453	3454	3457	3458	3525	3526	3550	3552	3574	3589	3594

Produced on Advanced Information Services Electronic Laser Printer, PFOJ155, DTN: 237981

		3709	4218	4693	4702	7352	7623	7664	7859				
D1		1141 #	2321	2363	2415	2418	2786	2790	3731	3736	5124	5132	5146
		5151	5247										
D2		1145 #	3683	3720	3752	4751	4755	5096	5169	5253	6893	6894	
E1		1140 #	2330	2335	2341	3570	3656						
E2		1143 #	4700	4719	4759	4763	4767	5919	5928	5929	5931	6850	
(U) RAMADR		677 #											
AC#		678 #	2357	2370	2384	2521	2588	2689	2691	2722	2728	2814	2824
		2856	2866	2886	2896	2906	2937	2965	2968	2982	2988	3000	3003
		3016	3078	3090	3111	3276	3291	3294	3299	3300	3370	3389	3429
		3432	3435	3438	3441	3444	3447	3450	3473	3487	3501	3509	3679
		3710	3712	3717	3750	3772	3775	3780	3808	3828	3840	4057	4071
		4087	4098	4101	4118	4119	4145	4147	4194	4209	4316	4317	4326
		4407	4409	4451	4453	4624	4625	4629	4707	4713	4717	4993	5065
		5104	5113	5114	5178	5201	5203	5283	5316	5346	5348	5411	5412
		5440	5446	5548	5552	5554	5558	5650	5701	5802	5803	5958	6049
		6086	6088	6096	6122	6158	6260	6292	6294	6306	6315	6385	6452
		6606	6679	6931	6955	7533	7544	7547	7561	7563	7732	7834	8202
AC*#		679 #	2173	2174	2181	2371	2383	2489	2772	2781	3022	3043	3087
		3117	4084	4096	4175	4177	4179	4187	4189	4196	4198	4210	4211
		4212	4213	4214	4215	4327	4411	4433	4437	4440	4454	4455	4457
		4460	4479	4481	4483	4484	4610	4611	4614	4615	4616	4619	4620
		4621	4627	4628	5547	5638	5691	5693	5699	5758	5766	5804	5811
		5955	5963	5973	5976	6006	6009	6022	6029	6036	6046	6084	6094
		6114	6129	6146	6161	6163	6189	6192	6194	6201	6202	6203	6205
		6211	6213	6216	6222	6224	6252	6254	6255	6259	6266	6267	6268
		6272	6273	6280	6282	6287	6293	6309	6320	6332	6341	6347	6358
		6361	6363	6367	6380	6386	6387	6510	6541	6546	6551	6588	6590
		6694	6695	6807	6814	6821	6840	6842	6849	6869	6890	6926	6928
		6934	6936	6940	6943	6948							
RAM		682 #	2183	2185	6423	6428	6433	6435					
VMA		681 #	2161	2269	2339	2353	2362	2366	2394	2405	2411	2424	2495
		2527	2533	2538	2541	2591	2787	3406	3549	3573	3588	3609	3613
		3617	3678	3695	3701	3723	3740	3746	3759	3817	3845	3967	3969
		3975	4002	4041	4737	4765	4771	4892	5134	5157	5160	5206	5248
		5912	5931	6351	6545	6591	6594	6622	6898	6913	6983	7105	7222
		7226	7230	7234	7238	7278	7308	7310	7317	7371	7439	7444	7455
		7492	7497	7588	7621	7668	7713	7716	7718	7728	7730	7827	7841
		7846	7866	7914	7931	7932	7934	7935	7938	7939	7941	7942	7944
		7945	7947	7948	7950	7951	7953	7954	7956	8221	8245	8310	8392
		8628	8740	8746	8797	8825	8827						
XR#		680 #	2279	2289	2301	2316	2329	2345	3554	3556	3564	4744	4747
		4749	4757	5923	5925	5928	6893	6895	6899	7657	7658	7666	
#		683 #	2160	2168	2170	2171	2176	2177	3663	4690	4949	4962	4967
		4972	4976	4977	4982	4988	5014	5027	5035	5040	5046	5176	5922
		5935	5938	5968	5985	5992	5996	6013	6025	6030	6041	6045	6056
		6057	6058	6060	6061	6062	6063	6073	6080	6104	6120	6127	6132
		6133	6140	6143	6156	6174	6178	6181	6185	6295	6297	6338	6348
		6353	6354	6356	6359	6372	6374	6378	6381	6382	6403	6553	6554
		6577	6579	6615	6620	6630	6692	6698	6725	6729	6732	6737	6738
		6745	6812	6819	6823	6825	6914	6930	6939	6945	6953	6954	6957
		6965	7050	7072	7084	7152	7157	7209	7211	7213	7215	7217	7223
		7227	7231	7235	7239	7246	7253	7255	7260	7264	7266	7285	7288
		7298	7299	7302	7318	7319	7323	7403	7404	7652	7655	7660	7672
		7674	7686	7692	7694	7839	7911	7913	7918	7922	7926	7936	7937

Produced on Advanced Information Services Electronic Laser Printer, PKO/IES6, DTM, 223-2881

	7955	7957	7958	7960	8198	8199	8208	8209	8211	8212	8219	8229
	8231	8233	8235	8239	8240	8288	8291	8318	8320	8351	8358	8375
	8378	8381	8383	8397	8404	8442	8452	8454	8483	8499	8539	8540
	8541	8547	8567	8568	8574	8725	8736	8742	8751	8755	8767	
(D) READ	1349 #	2547	2548	2550	2552	2553	2555	2557	2558	2560	2562	2563
	2565	2584	2600	2601	2602	2603	2605	2606	2607	2608	2610	2611
	2613	2615	2616	2618	2620	2621	2623	2625	2626	2628	2630	2631
	2633	2635	2636	2638	2642	2643	2644	2645	2647	2648	2649	2650
	2652	2653	2655	2657	2658	2660	2662	2663	2665	2667	2668	2670
	2672	2673	2675	2677	2678	2680	2766	2767	2797	2798	2807	2808
	2809	2810	2817	2818	2819	2820	2827	2828	2829	2830	2832	2833
	2834	2835	2842	2849	2850	2851	2852	2859	2860	2861	2862	2869
	2870	2871	2872	2879	2880	2881	2882	2889	2890	2899	2900	2901
	2902	2910	2911	2912	2913	2920	2921	2922	2923	2930	2931	2932
	2933	2940	2941	2952	2953	2954	2956	2957	2958	3186	3187	3188
	3189	3190	3191	3202	3203	3204	3205	3206	3207	3208	3209	3220
	3221	3222	3223	3224	3225	3226	3227	3237	3238	3239	3240	3241
	3242	3243	3244	3359	3360	3361	3362	3363	3364	3365	3366	3376
	3377	3378	3379	3380	3381	3382	3383	3393	3394	3395	3396	3397
	3398	3399	3400	3409	3410	3411	3412	3413	3414	3415	3416	3476
	3490	3647	3670	3766	4050	4051	4052	4053	4064	4065	4066	4067
	4079	4080	4112	4113	4114	4115	4137	4138	4139	4140	4163	4304
	4305	4306	4307	4309	4310	4311	4312	4381	4682	4683	4684	4685
	4686	5260	5261	5262	5263	5264	5265	5266	5268	5269	5270	5271
	5272	5273	5274	5305	5306	5307	5309	5310	5311	5312	5334	5335
	5336	5338	5339	5340	5341	5384	5419	5420	5534	5535	5623	5684
	5908											
(U) READ CYCLE	1126 #	2228	2229	2231	2232	2251	2291	2297	2302	2308	2331	2335
	2364	2405	2416	2419	2430	3020	3326	3338	3453	3454	3457	3458
	3525	3526	3527	3532	3536	3539	3552	3566	3569	3594	3616	3636
	3709	3719	3753	3843	4000	4044	4218	4693	4702	4750	4754	4758
	4762	5095	5170	5253	5928	5929	5953	6100	6350	6377	6450	6455
	6511	6631	6747	6893	6894	6895	6896	6971	6979	7103	7221	7225
	7229	7233	7237	7276	7280	7316	7352	7440	7454	7487	7496	7505
	7579	7582	7623	7646	7667	7824	7859	7861	8294	8302	8307	8387
	8522	8624	8831									
(D) ROUND	1337 #	5263	5264	5265	5266	5271	5272	5273	5274	5309	5310	5311
	5312	5338	5339	5340	5341	5384	5420					
(U) RSRC	611 #											
AB	613 #	3663	7652									
AQ	612 #											
DO	619 #	2759	6155	6270	7440	7487	7579	7582	7618	7629	7635	7708
	7861	8225	8227	8241								
DA	617 #	2155	2158	2191	2202	2238	2242	2245	2248	2257	2260	2263
	2266	2517	3042	3580	3943	3983	4031	4395	4406	4800	5880	5882
	5884	5886	5888	5890	5892	5894	5896	6580	6592	6618	6658	6803
	7028	7031	7094	7249	7292	7293	7294	7428	7429	7430	7431	7432
	7433	7434	7436	7469	7501	8267	8346	8406	8424	8528	8750	
DQ	618 #											
OA	616 #	2320	2761	3689	3806	3812	4216	5221	5551	5553	5653	5654
	5704	5706	5926	6964	7471							
OB	615 #	2738	2740									
OQ	614 #	3628	3642	3960	3986	5225	5915	5959	6159	7459	8805	
(U) S#	1001 #	2175	2378	2397	2397	2412	2412	2428	2966	2969	2973	2983
	2985	3001	3004	3007	3012	3024	3039	3041	3068	3085	3094	3097

Produced on Advanced Information Services Electronic Laser Printer. PLO/RES. DTN: 222-7881

	3140	4119	4147	4184	4204	4224	4337	4434	4442	4704	4786	4809
	4814	4816	4818	4821	4884	4885	4886	4887	4888	4889	4897	4901
	4903	4905	4938	4954	4965	4978	4994	5039	5054	5213	5215	5233
	5235	5243	5244	5287	5289	5321	5326	5328	5354	5355	5368	5369
	5391	5395	5397	5398	5402	5410	5425	5428	5430	5481	5485	5487
	5490	5493	5501	5503	5504	5505	5506	5524	5529	5559	5570	5585
	5587	5628	5639	5643	5646	5656	5662	5668	5679	5717	5718	5728
	5742	5749	5752	5764	5781	5786	5788	5809	5834	5836	5842	5980
	5981	5983	6149	6294	6462	6464	6471	6520	6525	6527	6650	6652
	6654	6655	6661	6671	6854	7129	7147	7178	7188	7195	7407	7595
	7627	7681	7697	8262	8313	8361	8497					
(U) SCAD	978 #											
A	986 #	2175	3024	4119	4147	4184	4204	4224	4337	4434	4442	4700
	4714	4791	4792	4793	4794	4795	4900	4938	4954	4965	4978	4994
	5039	5054	5213	5215	5233	5235	5321	5368	5391	5395	5397	5398
	5402	5628	5639	5643	5656	5662	5717	5718	6294	6462	6464	6471
	6520	6650	6651	6661	6851	6872	6891	7129	7147	7178	7188	7195
	7407	7595	7627	7681	7697	8262	8313	8361	8497			
A*2	979 #	6643										
A+B	983 #	2378	2397	2397	2412	2412	2969	2973	2983	2985	3004	3007
	3012	3039	3041	3085	3094	4816	4821	4885	4886	4887	4888	4889
	4897	4905	5243	5244	5287	5289	5317	5326	5328	5369	5410	5412
	5424	5428	5481	5485	5487	5490	5493	5501	5503	5504	5505	5506
	5524	5529	5559	5570	5585	5587	5651	5668	5679	5742	5749	5752
	5764	5781	5786	5788	5809	5834	5836	5842	5980	5981	5983	6521
	6525	6653	6654	6655	6671	6959						
A-1	985 #	2185	2989	2991	3067	3069	3071	3075	3101	3108	3139	3141
	3143	3147	4250	4262	4268	4276	4286	4294	4549	4552	4592	4941
	4960	5060	5297	5299	5405	5429	5432	5435	5439	5603	5606	5613
	5631	6323	6467	6473	6668	7133	7149	7180	7197	7387	7400	7602
	7633	7688	7701	8265	8327	8330	8364	8503				
A-B	982 #	2966	3001	3068	3097	3140	4692	4697	4711	4732	4786	4809
	4814	4818	4884	4903	5348	5354	5355	5430	5702	5728	6033	6043
	6843	6846	6886	6888								
A-B-1	981 #	5283	5288	5549	5569							
A .AND. B	984 #	4704	4901	6149	6527	6652	6854					
A .OR. B	980 #	2428	5646									
(U) SCADA	987 #											
BYTE1	991 #	4692	4697	4700	4711	4791	4900	6033	6521	6651	6843	6851
	6886	6891	6959									
BYTE2	992 #	4792										
BYTE3	993 #	4793										
BYTE4	994 #	4794										
BYTE5	995 #	4715	4795	6644	6873							
PTR44	990 #	4732	6043	6846	6888							
S#	989 #	2175	2378	2397	2397	2412	2412	2428	2966	2969	2973	2983
	2985	3001	3004	3007	3012	3024	3039	3041	3068	3085	3094	3097
	3140	4119	4147	4184	4204	4224	4337	4434	4442	4704	4786	4809
	4814	4816	4818	4821	4884	4885	4886	4887	4888	4889	4897	4901
	4903	4905	4938	4954	4965	4978	4994	5039	5054	5213	5215	5233
	5235	5243	5244	5287	5289	5321	5326	5328	5354	5355	5368	5369
	5391	5395	5397	5398	5402	5410	5424	5428	5430	5481	5485	5487
	5490	5493	5501	5503	5504	5505	5506	5524	5529	5559	5570	5585
	5587	5628	5639	5643	5646	5656	5662	5668	5679	5717	5718	5728
	5742	5749	5752	5764	5781	5786	5788	5809	5834	5836	5842	5980

	5981	5983	6149	6294	6462	6464	6471	6520	6525	6527	6650	6652
	6654	6655	6661	6671	6854	7129	7147	7178	7188	7195	7407	7595
	7627	7681	7697	8262	8313	8361	8497					
SC	988 #	2185	2989	2991	3067	3069	3071	3075	3101	3108	3139	3141
	3143	3147	4250	4262	4268	4276	4286	4294	4549	4552	4592	4941
	4960	5060	5283	5288	5297	5299	5317	5348	5405	5412	5429	5432
	5435	5439	5549	5569	5603	5606	5613	5631	5651	5702	6323	6467
	6473	6653	6668	7133	7149	7180	7197	7387	7400	7602	7633	7688
	7701	8265	8327	8330	8364	8503						
(U) SCADB	996 #											
EXP	998 #	2397	2397	2412	2412	5283	5287	5289	5317	5348	5412	5549
	5559	5570	5651	5702								
FE	997 #	2428	2966	2969	2973	2983	2985	3001	3004	3007	3012	3039
	3041	3094	3097	4704	4786	4809	4814	4816	4821	4885	4886	4887
	4888	4889	4897	4901	4903	4905	5243	5244	5288	5326	5328	5354
	5355	5369	5425	5428	5430	5481	5485	5487	5490	5493	5501	5503
	5504	5505	5506	5524	5529	5569	5585	5587	5646	5668	5679	5728
	5742	5749	5752	5764	5781	5786	5788	5809	5834	5836	5842	5981
	5983	6149	6521	6525	6527	6652	6653	6654	6655	6671	6854	
SHIFT	999 #	2378	3068	3085	3140	5410						
SIZE	1000 #	4692	4697	4711	4732	4818	4884	5980	6033	6043	6843	6846
	6886	6888	6959									
(U) SETFOV	1082 #	5350	5362	5715								
(U) SETFPD	1088 #	4232	4746	6389								
(U) SETNDV	1083 #	4339	4374	4405	4491	4998	5350	5362	5715			
(U) SETOV	1080 #	4132	4133	4157	4219	4339	4374	4405	4491	4998	5350	5362
	5427	5715										
(U) SHSTYLE	825 #											
ASHC	830 #	2991	3100	3103	3107	3110	4268	4276	4286	4294	4400	4404
	4473	4477	4493	5299	5402	5404	5434	5578	5580	5584	5586	5602
	5606	5613	5616	5618	5630	5671	5675	5710	5724	5742	5748	5751
	5764	5781	5785	5787	5809	5834						
DIV	832 #	4549	4552	4553	4596	4600	5481	5493	5501	5505	5598	5600
	5601											
LSHC	831 #	3071	3075	4581	4584							
NORM	826 #	2425	2985	4227	4261	4434	4590	5208	5223	5224	5228	5229
	5230	5244	5632	5722								
ONES	828 #	4820	4823	4896	4899	5983						
ROT	829 #	3007	3012									
ROTC	833 #	3143	3147									
ZERO	827 #											
(U) SKIP	903 #											
ACO	911 #	2484	3388	4689								
ADEQO	923 #	3017	3332	3344	3435	3447	4122	4127	4329	4331	4335	4349
	4394	4423	4453	4534	4612	4617	4622	4638	4645	4963	4992	5346
	5372	5374	5447	5489	5513	5669	5720	5762	5807	6137	6190	6266
	6269	6558	6595	6616	6739	7296	8289	8292	8358	8376	8379	8483
	8567											
ADLEQO	907 #	2188	3987	3992	5389	5394	5508	5673	5676	5744	5747	5750
	5753	5813	5913	5956	6095	6097	6179	6199	6383	6399	6453	6488
	6491	6523	6987	7111	7118	7154	7457	7458	7649	7651	7665	8249
	8315	8322	8325	8340	8345	8394	8413	8417	8422	8433	8437	8439
	8459	8512	8531	8533	8571	8576	8598	8601	8603	8606	8609	8630
	8640	8694	8712	8753								
ADREQO	908 #	3086	3592	3607	5120	6324	6397	6752	7029	7032	7057	7059

Produced on Advanced Information Services Electronic Laser Printer, PKCJES6, DTN: 223-7781

	7062	7064	7066	7332	7334	7338	7340	7342	7344	7346	7348	7445
	7498	7518	7590	7617	8200	8464	8492	8698	8701	8706		
CRYO	906 #	3685	3728	3757	5358	5482	5484	5486	5488			
CRY1	919 #	4085	4097	5798	6214	6220	6233	6424	6435			
CRY2	914 #	5771	5831									
DPO	915 #	2398	2413	2572	2750	2757	3040	3112	3329	3341	3432	3444
	3774	3777	3782	4104	4129	4151	4154	4209	4217	4318	4354	4356
	4361	4366	4370	4392	4409	4421	4428	4443	4462	4480	4532	4539
	4554	4995	5139	5163	5249	5285	5289	5318	5348	5356	5390	5412
	5443	5550	5652	5699	5703	5714	5727	5964	5977	5994	6007	6014
	6032	6074	6076	6098	6115	6118	6124	6164	6256	6261	6298	6307
	6340	6357	6429	6479	6505	6696	6726	6768	6822	6941	7106	7251
	8276	8374	8597									
DP18	916 #	2377	2736	2743	5937	6184	6388	6477	6800	6801	7484	
EXECUTE	925 #	7853										
FPD	910 #	4171	4692	4697	4711	6288						
INT	912 #	5143	6034	8213	8408	8626	8734	8765	8816			
IOLGL	904 #	2434	3537	3540								
IOT	917 #											
JFCL	918 #	3599										
KERNEL	909 #	3530	3534	3535	3542	3651	6966					
LE	913 #	3335	3347	3438	3450	4388	5022	5366	5690	6337	7262	7919
LLE	905 #	7658										
SC	924 #	2185	2989	2991	3067	3069	3071	3075	3092	3101	3108	3139
	3141	3143	3147	4250	4262	4268	4276	4286	4294	4549	4552	4592
	4941	4960	5060	5297	5299	5405	5429	5432	5435	5439	5552	5554
	5603	5606	5613	5631	6323	6467	6473	6659	6668	7133	7149	7180
	7197	7387	7400	7602	7633	7701	8265	8327	8330	8364	8503	
TRAP CYCLE	921 #	8752										
TXXX	920 #	3285										
-1 MS	928 #	5135	6029	7300	8403							
-CONTINUE	927 #	7855										
-IO BUSY	926 #	7683	7689	7698	7706							
(U) SPEC	789 #	2569	2575	2689	2691	2724	2730	2738	2740	2759	2761	2804
	2814	2824	2856	2866	2886	2896	2907	2917	2947	4124	4130	4153
	4155	8518	8818									
APR EN	803 #	2166	7071	7156								
APR FLAGS	801 #	7076	7080	7270								
ASHOV	810 #	2991	3107	3110								
CLR IO BUSY	793 #	7576	7614									
CLR IO LATCH	792 #	7661	7669	7673	7676	7685	7691	7700	7705			
CLRCLK	791 #	7247	7301	8453	8737							
CLRCSH	802 #	7382	7383	7386								
EXPTST	811 #	5490	5506	5836								
FLAGS	812 #	2217	2218	2578	3404	3420	3487	3501	3557	3558	3561	3562
	3598	3619	3620	3690	3702	3758	3788	3792	3807	3813	4004	4005
	4006	4058	4072	4089	4099	4102	4132	4133	4157	4207	4219	4229
	4232	4339	4374	4405	4491	4640	4642	4707	4724	4746	4998	5350
	5362	5427	5715	6389	6390	6988	7460	7475	8756			
INHCRY18	807 #	3510	3680	3727	3756	3776	3781	5030	5105			
LDACBLK	813 #	2163	7126	7143								
LDINST	814 #	2270	3655	6984	7867							
LDPAGE	794 #	7377	8534									
LDPI	809 #	3608	7363	7438	7920							
LDPXCT	796 #	3660										

Produced on Advanced Information Services Electronic Laser Printer, PNO1E56, DTN: 223-7881

LOADIR	808 #	5918											
LOADXR	799 #	2341	3550	3574	3589	4700	4719	4767	5919	5931	6850	6887	
	6889	6898	7664										
MEMCLR	804 #	2161	3609	7439	7841	7914	8245	8740	8746				
NICOND	795 #	2228	2229	2231	2232	2292	2298	2522	3043	3089	3117	3326	
	3338	3429	3453	3454	3457	3458	3525	3526	3552	3594	3711	3787	
	3794	4218	4482	4485	4693	5440	5446	5805	7352	7623	8202		
PREV	798 #	7825	7833										
PXCT OFF	806 #	2240	2243	2246	2258	2261	2264	2278					
SWEEP	805 #	7395	7396	7399									
WAIT	797 #	2410											
#	790 #	7849	7852	7915									
(U) STATE	1004 #												
BLT	1006 #	5106											
COMP-DST	1013 #	6123											
CVTDB	1012 #	6170											
DST	1009 #	6085											
DSTF	1011 #	6023											
EDIT-DST	1015 #	6142	6352	6379	6598	6624	8773	8777					
EDIT-S+D	1016 #	6564											
EDIT-SRC	1014 #	6116	6503	6560	8771	8779	8781	8785					
MAP	1007 #	8201											
SIMPLE	1005 #												
SRC	1008 #	5972	6001	6071	8775								
SRC+DST	1010 #	5998	6077										
(U) SWITCH%													
FULL	398	1295	1297	2172	2192	2197	2201	4166	4234	4237	4384	4494	
	4497	4565	4630	4916	5067	5071	7150	7158					
INHCST	7	401	1241	1243	1245	8004	8006	8353	8356	8357	8360	8478	
	8481	8482	8485	8556	8565	8566	8582						
KIPAGE	9	410	418	420	422	1265	1267	1269	3947	3957	4009	4020	
	7160	7167	7173	7175	8270	8274	8279	8282	8646	8692	8801	8804	
	8808	8813											
KLPAGE	413	417	419	1271	1273	1275	3948	3952	3958	3972	7161	7166	
	7169	7171	8269	8278	8284	8487	8550	8644	8800	8806	8823	8828	
NOCST	404	425	427	1247	1249	1251	8348	8423	8427	8441	8444	8445	
	8449	8475	8494	8551	8583								
NONSTD	10	429	1253	1255	1257								
SIM	395	2210	2215	2222									
UBABLT	8	407	416	423	1259	1261	1263	5181	5255	7797	7799	7801	
(U) T	941 #												
2T	944 #	4409	4422	5289	5559	6133	6203	6267	6555	6692	6737	7302	
	7659												
3T	945 #	2188	2413	2423	2578	3111	3285	3370	3404	3420	3510	3600	
	3684	3719	3727	3752	3756	3773	3778	3783	3820	3834	4058	4072	
	4175	4177	4196	4230	4367	4371	4421	4427	4428	4429	4446	4450	
	4471	4479	4560	4625	4642	4700	4715	4786	4804	4900	4944	4951	
	5031	5047	5051	5117	5119	5138	5162	5249	5283	5317	5358	5389	
	5390	5393	5425	5428	5482	5484	5486	5488	5547	5548	5551	5553	
	5652	5653	5654	5698	5702	5704	5706	5714	5797	5803	5963	5981	
	5983	6031	6035	6098	6205	6219	6233	6265	6268	6353	6356	6388	
	6389	6417	6423	6426	6435	6460	6474	6501	6521	6522	6644	6651	
	6695	6754	6844	6851	6872	6886	6890	6893	6895	6931	6986	7105	
	7111	7117	7154	7419	7444	7448	7456	7457	7458	7493	7497	7499	
	7666	8296	8304	8526	8543	8761	8791						

4T	946 #	3016	3487	3501	4084	4089	4096	4099	4101	4991	4996	5977
	6007	6184	6189	6214	6255	6428	6476	6616	6739	6940	7919	8216
	8289	8292	8376	8379								
5T	947 #	7688	7705									
(D) TEST	1350 #	2549	2549	2550	2550	2554	2554	2555	2555	2559	2559	2560
	2560	2564	2564	2565	2565	2584	2602	2602	2603	2603	2607	2607
	2608	2608	2612	2612	2613	2613	2617	2617	2618	2618	2622	2622
	2623	2623	2627	2627	2628	2628	2632	2632	2633	2633	2637	2637
	2638	2638	2644	2644	2645	2645	2649	2649	2650	2650	2654	2654
	2655	2655	2659	2659	2660	2660	2664	2664	2665	2665	2669	2669
	2670	2670	2674	2674	2675	2675	2679	2679	2680	2680	2776	2799
	2799	2800	2800	2809	2809	2810	2810	2819	2819	2820	2820	2829
	2829	2830	2830	2834	2834	2835	2835	2844	2844	2845	2845	2851
	2851	2852	2852	2861	2861	2862	2862	2871	2871	2872	2872	2881
	2881	2882	2882	2891	2891	2892	2892	2901	2901	2902	2902	2912
	2912	2913	2913	2922	2922	2923	2923	2932	2932	2933	2933	2942
	2942	2943	2943	3393	3393	3396	3396	3397	3398	3399	3400	3409
	3410	3411	3412	3413	3414	3415	3416	4052	4052	4053	4053	4066
	4066	4067	4067	4114	4114	4115	4115	4139	4139	4140	4140	4306
	4306	4307	4307	4311	4311	4312	4312	4683	4685	5261	5261	5262
	5262	5265	5265	5266	5266	5269	5269	5270	5270	5273	5273	5274
	5274	5306	5306	5307	5307	5311	5311	5312	5312	5335	5335	5336
	5336	5340	5340	5341	5341	7204						
(U) TRAP1	1097 #	4132	4133	4157	4219	4339	4374	4405	4491	4998	5350	5362
	5427	5715										
(U) TRAP2	1096 #	3702	3758	3788	3792							
(U) VECTOR CYCLE	1168 #	7488										
(D) VMA	1352 #	2548	2553	2558	2563	2601	2606	2611	2616	2621	2626	2631
	2636	2643	2648	2653	2658	2663	2668	2673	2678	2797	2798	2808
	2818	2828	2833	2850	2860	2870	2880	2889	2890	2900	2911	2921
	2931	2940	2941	2952	2953	2954	2956	2957	2958	3476	3490	3516
	3766	4051	4065	4113	4138	4305	4310					
(U) WAIT	1157 #	2228	2228	2229	2229	2231	2231	2232	2232	2251	2251	2268
	2291	2291	2297	2297	2302	2308	2321	2331	2335	2339	2352	2361
	2364	2365	2393	2404	2405	2405	2416	2419	2423	2430	2430	2494
	2526	2532	2537	2540	2589	2589	2590	2785	2785	2787	2791	2791
	3020	3020	3326	3326	3338	3338	3405	3405	3406	3453	3453	3454
	3454	3457	3457	3458	3458	3525	3525	3526	3526	3527	3532	3536
	3539	3548	3552	3552	3566	3569	3572	3587	3594	3594	3613	3616
	3617	3630	3630	3633	3633	3636	3641	3641	3677	3682	3682	3694
	3700	3709	3709	3719	3722	3732	3732	3737	3737	3739	3745	3753
	3759	3815	3815	3816	3826	3826	3843	3844	3965	3966	3968	3974
	3979	3979	4000	4001	4026	4026	4036	4036	4040	4044	4218	4218
	4693	4693	4702	4702	4730	4730	4736	4736	4737	4750	4754	4758
	4762	4764	4770	4891	4891	4892	5095	5123	5123	5131	5131	5133
	5147	5147	5152	5152	5156	5159	5170	5205	5246	5246	5248	5253
	5912	5928	5929	5931	5953	6100	6350	6351	6377	6450	6455	6511
	6543	6543	6545	6589	6589	6591	6594	6622	6631	6747	6893	6894
	6895	6896	6898	6912	6968	6971	6976	6979	6983	7103	7104	7221
	7222	7225	7226	7229	7230	7233	7234	7237	7238	7276	7277	7280
	7305	7305	7308	7309	7309	7310	7316	7317	7352	7352	7370	7370
	7371	7443	7454	7455	7470	7470	7491	7496	7497	7505	7585	7587
	7620	7621	7623	7623	7646	7667	7668	7713	7716	7718	7726	7728
	7730	7824	7826	7832	7832	7844	7846	7847	7847	7859	7859	7865
	7930	7931	7932	7933	7933	7934	7935	7938	7939	7940	7940	7941

	7942	7943	7943	7944	7945	7946	7946	7947	7948	7949	7949	7950
	7951	7952	7952	7953	7954	7956	8294	8302	8307	8309	8387	8391
	8544	8579	8624	8627	8795	8796	8825	8827	8831	8834	8834	
(U) WORK	1020 #											
ACO	1061 #											
AC1	1062 #											
AC2	1063 #											
AC3	1064 #											
ADJBPW	1039 #	5014	5035									
ADJP	1034 #	4949	4976									
ADJPTR	1036 #	4690	4967	5029								
ADJQ1	1037 #	4972	4990									
ADJR2	1038 #	4982	5049									
ADJS	1035 #	4962	4977	5040								
APR	1041 #	2168	7050	7072	7084	7152	7157	8199				
BADWO	1021 #	8239										
BADW1	1022 #	8240										
BDH	1051 #	6297	6372	6381	6403							
BDL	1052 #	6295	6338	6378	6382							
CBR	1031 #	7211	7227	8358	8483	8567	8568					
CMS	1048 #	6120	6132	6133	6143							
CSTM	1032 #	7215	7235	8574								
DDIV SGN	1065 #											
DECHI	1070 #	2177	2179	6426								
DECLO	1069 #	2176	2178	6417								
DIV	1024 #											
DVSOR H	1066 #											
DVSOR L	1067 #											
EO	1043 #	5922	6348	6620	6630							
E1	1044 #	5935	5938	6374	6553	6554	6729	6732	6737	6745		
FILL	1047 #	6104	6127	6140	6615	6692	6698	6812	6914			
FSIG	1049 #	6577	6579									
HSBADR	1040 #	2160	7217	7239	7913	7918						
MSK	1046 #	5985	6156	6738								
MUL	1023 #											
PERIOD	1057 #	7266	7318	7323								
PTA.E	1073 #	7404	8288	8375	8381							
PTA.U	1074 #	7403	8291	8378	8383							
PUR	1033 #	7213	7231	8351	8442							
SBR	1030 #	7209	7223	8318	8320	8397						
SLEN	1045 #	5968	5992	5996	6013	6025	6041	6045	6073	6080	6174	6178
	6181	6185	6353	6354	6356	6359	6725	6819	6823	6825	6939	6953
	6957											
SV.ARX	1027 #	6057	6062	7839	7911	7922	7937	7960	8212	8539	8767	
SV.AR	1026 #	6030	6060	7936	7957	8208	8547					
SV.AR1	1076 #	8404	8454									
SV.BRX	1029 #	5176	6058	6063	6930	6945	6954	8209	8541			
SV.BR	1028 #	6056	6061	8219	8229	8231	8233	8235	8540	8725		
SV.VMA	1025 #	7686	7692	7694	7926	7955	7958	8198	8211	8499	8742	8751
TIMEO	1055 #	2170	7255	7288	7298							
TIME1	1056 #	2171	7246	7253	7285	7299	7302	8452	8736			
TRAPPC	1075 #	6965	8755									
TTG	1058 #	7260	7264	7319								
YSAVE	1072 #	3663	7652	7655	7660	7672	7674					
(D) WRITE	1353 #											

Cross Reference Listing

(U) WRITE CYCLE

1128 #	2569	2575	2589	2689	2691	2724	2730	2738	2740	2759	2761
2785	2791	2804	2814	2824	2856	2866	2886	2896	2907	2917	2947
3405	3630	3633	3641	3682	3732	3737	3815	3826	3965	3979	4026
4036	4058	4072	4124	4130	4153	4155	4730	4736	4891	5123	5131
5147	5152	5246	5444	5448	5494	5516	6543	6589	7305	7309	7370
7470	7618	7629	7635	7832	7844	7847	7930	7933	7940	7943	7946

(U) WRITE TEST

1127 #	2569	2575	2589	2689	2691	2724	2730	2738	2740	2759	2761
2785	2791	2804	2814	2824	2856	2866	2886	2896	2907	2917	2947
3405	3630	3633	3641	3682	3732	3737	3815	3826	3965	3979	4026
4036	4058	4072	4124	4130	4153	4155	4730	4736	4891	5123	5131
5147	5152	5246	5444	5448	5494	5516	6543	6589	7305	7309	7370
7440	7454	7470	7487	7496	7579	7582	7618	7629	7635	7832	7844
7847	7861	7930	7933	7940	7943	7946	7949	7952	8249	8294	8302

(U) WRU CYCLE

(U) #	1164 #	7441	975 #	2149	2151	2153	2155	2158	2188	2202	2238	2242	2245	2257
	2260	2263	2420	2422	2435	2457	2458	2745	2747	2752	2754	2771		
	3042	3113	3115	3529	3538	3541	3543	3544	3545	3578	3583	3586		
	3592	3606	3607	3627	3640	3726	3755	3818	3833	3919	3921	3923		
	3925	3927	3929	3931	3933	3940	3943	3983	3988	3990	3993	3995		
	4023	4031	4038	4105	4106	4358	4395	4396	4406	4414	4459	4512		
	4538	4545	4636	4801	4945	4947	4952	4970	4974	4980	4984	5013		
	5018	5052	5056	5063	5116	5212	5217	5226	5232	5237	5290	5291		
	5294	5295	5319	5320	5324	5352	5353	5389	5393	5413	5414	5551		
	5553	5633	5653	5654	5673	5676	5704	5706	5880	5882	5884	5886		
	5888	5890	5892	5894	5896	5913	5916	5940	5956	5960	6095	6097		
	6155	6160	6167	6172	6179	6183	6199	6241	6257	6258	6264	6270		
	6290	6303	6305	6312	6317	6324	6327	6383	6397	6399	6406	6431		
	6437	6453	6465	6468	6476	6488	6491	6508	6523	6581	6592	6601		
	6603	6619	6638	6659	6664	6673	6676	6752	6774	6778	6782	6786		
	6790	6794	6804	6810	6908	6919	6986	6991	7000	7001	7002	7006		
	7007	7008	7009	7014	7015	7016	7019	7020	7021	7022	7023	7024		
	7025	7026	7028	7031	7037	7052	7053	7056	7057	7059	7062	7064		
	7066	7078	7088	7091	7095	7109	7114	7117	7121	7136	7153	7154		
	7155	7181	7190	7198	7218	7240	7249	7268	7283	7335	7336	7379		
	7381	7392	7394	7421	7422	7423	7424	7425	7426	7427	7428	7429		
	7430	7431	7432	7433	7434	7437	7448	7457	7458	7473	7493	7500		
	7590	7598	7604	7617	7649	7651	7665	7708	7794	7803	7805	7807		
	7809	7811	7813	7849	7852	7863	7915	8193	8197	8200	8225	8227		
	8241	8251	8254	8260	8268	8297	8305	8316	8323	8340	8347	8395		
	8407	8410	8418	8425	8432	8435	8461	8465	8468	8493	8506	8510		
	8514	8516	8529	8530	8532	8537	8572	8593	8599	8604	8610	8698		
	8701	8704	8707	8716	8721	8729	8791							

Produced on Advanced Information Services Electronic Laser Printer, PKO1156, DTN: 223-7881

; Location / Line Number Index

; Dcode	Loc'n	0	1	2	3	4	5	6	7
D 0000		3905	5849	5850	5851	5852	5853	5854	5855
D 0010		5857	5858	5859	5860	5862	5863	5864	5865
D 0020		5867	5868	5869	5870	5871	5872	5873	5874
D 0030		3852	3853	3854	3855	3856	3857	3858	3859
D 0040		3863	3864	3865	3866	3867	3868	3869	3870
D 0050		3871	3872	3873	3874	3875	3876	3877	3878
D 0060		3879	3880	3881	3882	3883	3884	3885	3886
D 0070		3887	3888	3889	3890	3891	3892	3893	3894
D 0100		3898	3899	3900	3901	3906	3766	3907	3908
D 0110		5534	5535	5623	5684	4079	4080	4163	4381
D 0120		2766	2767	5419	5908	2775	2776	5420	5384
D 0130		3909	3910	5385	4682	4683	4684	4685	4686
D 0140		5260	3911	5261	5262	5263	5264	5265	5266
D 0150		5268	3912	5269	5270	5271	5272	5273	5274
D 0160		5305	3913	5306	5307	5309	5310	5311	5312
D 0170		5334	3914	5335	5336	5338	5339	5340	5341
D 0200		2547	2548	2549	2550	2552	2553	2554	2555
D 0210		2557	2558	2559	2560	2562	2563	2564	2565
D 0220		4112	4113	4114	4115	4137	4138	4139	4140
D 0230		4304	4305	4306	4307	4309	4310	4311	4312
D 0240		2952	2953	2954	2955	2956	2957	2958	3915
D 0250		2584	5109	3504	3505	3516	3518	3647	8188
D 0260		3669	3670	3671	3672	3799	3800	3801	3802
D 0270		4050	4051	4052	4053	4064	4065	4066	4067
D 0300		3350	3351	3352	3353	3354	3355	3356	3357
D 0310		3359	3360	3361	3362	3363	3364	3365	3366
D 0320		3462	3463	3464	3465	3466	3467	3468	3469
D 0330		3376	3377	3378	3379	3380	3381	3382	3383
D 0340		3476	3477	3478	3479	3480	3481	3482	3483
D 0350		3393	3394	3395	3396	3397	3398	3399	3400
D 0360		3490	3491	3492	3493	3494	3495	3496	3497
D 0370		3409	3410	3411	3412	3413	3414	3415	3416
D 0400		2797	2798	2799	2800	2807	2808	2809	2810
D 0410		2817	2818	2819	2820	2827	2828	2829	2830
D 0420		2832	2833	2834	2835	2842	2843	2844	2845
D 0430		2849	2850	2851	2852	2859	2860	2861	2862
D 0440		2869	2870	2871	2872	2879	2880	2881	2882
D 0450		2889	2890	2891	2892	2899	2900	2901	2902
D 0460		2910	2911	2912	2913	2920	2921	2922	2923
D 0470		2930	2931	2932	2933	2940	2941	2942	2943
D 0500		2600	2601	2602	2603	2605	2606	2607	2608
D 0510		2610	2611	2612	2613	2615	2616	2617	2618
D 0520		2620	2621	2622	2623	2625	2626	2627	2628
D 0530		2630	2631	2632	2633	2635	2636	2637	2638
D 0540		2642	2643	2644	2645	2647	2648	2649	2650
D 0550		2652	2653	2654	2655	2657	2658	2659	2660
D 0560		2662	2663	2664	2665	2667	2668	2669	2670
D 0570		2672	2673	2674	2675	2677	2678	2679	2680

Produced on Advanced Information Services Electronic Laser Printer, PK011556, DTN: 225-7881

; T10KL.MCR[10,1141]

15:34 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 310

; Dcode Loc'n 0 1 2 3 4 5 6 7
Location / Line Number Index

D 0600	3175	3176	3177	3178	3179	3180	3181	3182
D 0610	3184	3185	3186	3187	3188	3189	3190	3191
D 0620	3193	3194	3195	3196	3197	3198	3199	3200
D 0630	3202	3203	3204	3205	3206	3207	3208	3209
D 0640	3211	3212	3213	3214	3215	3216	3217	3218
D 0650	3220	3221	3222	3223	3224	3225	3226	3227
D 0660	3228	3229	3230	3231	3232	3233	3234	3235
D 0670	3237	3238	3239	3240	3241	3242	3243	3244
D 0700	6996	6997	7204	7737	7818	7819	7738	7739
D 0710	7525	7526	7536	7537	7550	7551	7741	7742
D 0720	7527	7528	7538	7539	7552	7553	7744	7745
D 0730	7747	7748	7749	7750	7751	7752	7753	7754
D 0740	7756	7757	7758	7759	7760	7761	7762	7763
D 0750	7765	7766	7767	7768	7769	7770	7771	7772
D 0760	7774	7775	7776	7777	7778	7779	7780	7781
D 0770	7783	7784	7785	7786	7787	7788	7789	7790

; Ucode Loc'n 0 1 2 3 4 5 6 7
 Location / Line Number Index

U 0000	2149:	7912:	2150	2158:	7853:	7855:	5209=	5212=
U 0010	2182=	7913:	2183=	2151	3285=	3288=	3291=	3294=
U 0020	3971=	4121=	3977=	3980=	7914=	4122=	7915=	3983=
U 0030	3559=	4149=	3563=	4490=	3567=	4150=	3571=	4491=
U 0040	2354:	2358:	2362:	2370:	2380:	2383:	2390:	2394:
U 0050	2407:	2413:	2434:	2152	4412=	4414=	4448=	4449=
U 0060	3631=	2153	5042=	5443=	3634=	3636=	5049=	7581=
U 0070	4752=	2156	4756=	5444=	4760=	4416=	4763=	7582=
U 0100	2160	2241=	2244=	2247=	7840=	2248=	2161	2252=
U 0110	2163	2259=	2262=	2265=	7842=	2266=	7844=	2271=
U 0120	4176=	4178=	4269=	4273=	4179=	2166	4277=	4281=
U 0130	3829=	2168	5689=	5691=	3832=	3836=	2170	7047:
U 0140	2191=	2194=	4287=	4291=	5368=	5369=	4295=	4299=
U 0150	3551=	2202=	3552=	5599=	5371=	5373=	5375=	5376=
U 0160	4332=	4335=	2171	5323=	4338=	4339=	4341=	5325=
U 0170	5925=	4185=	5927=	5600=	5928=	4186=	5929=	5377=
U 0200	5782=	2292=	5784=	2298=	5786=	2304=	5788=	2309=
U 0210	5790=	2317=	2173	2321=	2174	2331=	5791=	2335=
U 0220	3709=	3711=	3713=	2175	2187=	2188=	2176	7617=
U 0230	6893=	6845=	6894=	6846=	6895=	2177	6896=	7618=
U 0240	4691=	4692=	2178	5670=	2179	4693=	6534=	6535=
U 0250	3326=	3329=	3332=	3335=	3338=	3341=	3344=	3347=
U 0260	2231=	2232=	5523=	5671=	4720=	4723=	5524=	4724=
U 0270	3429=	3432=	3435=	3438=	3441=	3444=	3447=	3450=
U 0300	3586=	3590=	5528=	3592=	2399=	2401=	5529=	3593=
U 0310	4195=	4197=	5832=	2180	4198=	2184	5834=	3594=
U 0320	3606=	3607=	6693=	2185	2218	3608=	6694=	5508=
U 0330	4357=	4359=	2220	4362=	3967=	3610=	3969=	5510=
U 0340	4788=	4791=	4792=	2279	4793=	4794=	2341	4795=
U 0350	4730=	4731=	4732=	4733=	4434=	4474=	4435=	4475=
U 0360	4884=	4885=	4886=	2346	4887=	4888=	2364	4889=
U 0370	4514=	2367	2416=	2419=	4515=	4516=	4517=	4518=
U 0400	4036:	2372	2435=	2436=	7349=	7350=	2384	5526=
U 0410	4810=	4444=	4811=	4445=	4706=	7352=	4708=	5527=
U 0420	5481=	5482=	5483=	5484=	5485=	5486=	5487=	5488=
U 0430	5489=	2386	2421	5582=	2457=	2458=	5490=	5583=
U 0440	5501=	6321=	5502=	6323=	5503=	2422	5504=	6324=
U 0450	5505=	2425	2426	5616=	2539=	2542=	5506=	5617=
U 0460	6178=	6180=	5836=	5840=	6182=	6184=	5843=	2428
U 0470	7472=	2431	5606=	5608=	7474=	7476=	5609=	5610=
U 0500	5993=	5995=	2991=	2995=	5997=	5999=	2592	6001=
U 0510	6379=	2722	2728	5729=	3021=	3022=	6380=	5730=
U 0520	5743=	5744=	5746=	5747=	5749=	5750=	5752=	5753=
U 0530	5755=	2787	2792	6818=	3040=	3041=	5756=	6820=
U 0540	6074=	6076=	6078=	4205=	3071=	3072=	6081=	4207=
U 0550	5013=	5015=	2907	5019=	3075=	3076=	5936=	5937=
U 0560	6212=	6215=	6371=	4228=	2973	6216=	6372=	4230=
U 0570	5214=	5216=	2986	5217=	5972=	5973=	5975=	5978=

; Ucode	Loc'n	0	1	2	3	4	5	6	7
U 0600		6218=	6221=	3088=	3089=	2989	6222=	6578=	6579=
U 0610		5234=	5236=	3005	5237=	3095=	3097=	6170=	6173=
U 0620		6229=	6230=	6234=	5639=	3101=	3103=	6235=	5643=
U 0630		5794=	3007	5796=	5798=	3108=	3110=	6377=	6378=
U 0640		6358=	6360=	3114=	3116=	3143=	3144=	3008	6361=
U 0650		6022=	6024=	3010	6026=	6462=	6463=	6464=	6465=
U 0660		4971=	6477=	4972=	6480=	4981=	6482=	4982=	6484=
U 0670		5202=	6486=	5203=	6489=	5285=	6492=	5286=	6494=
U 0700		6555=	3013	6559=	6561=	6562=	6565=	3024	6567=
U 0710		6502=	6504=	6506=	6509=	6510=	6512=	6731=	6732=
U 0720		5427=	6769=	5428=	6771=	5429=	6775=	5430=	6779=
U 0730		5644=	6783=	5656=	6787=	5645=	6791=	5657=	6795=
U 0740		6600=	6602=	3147=	3149=	3389=	3390=	6604=	3042
U 0750		6040=	6042=	6044=	6045=	6808=	6811=	6813=	6815=
U 0760		6625=	6626=	3453=	3454=	3457=	3458=	6627=	3043
U 0770		7420=	7421=	7422=	7423=	7424=	7425=	7426=	7427=
U 1000		8314=	8317=	8319=	8321=	3578=	3579=	3069	8323=
U 1010		8331=	8332=	5663=	7562=	3077	8333=	5668=	7564=
U 1020		6084=	6085=	5914=	6086=	3583=	3584=	5916=	7590=
U 1030		6117=	6119=	6120=	3078	3627=	3628=	3092	7591=
U 1040		8377=	8380=	8382=	8384=	3640=	3641=	3112	8388=
U 1050		6126=	6128=	6129=	3117	5717=	7659=	5718=	7661=
U 1060		8217=	8219=	3141	8223=	5721=	8225=	5722=	8227=
U 1070		8229=	8231=	8233=	8235=	6193=	6195=	6196=	3299
U 1100		8766=	8768=	8770=	8772=	8774=	8776=	8778=	8780=
U 1110		8782=	8783=	8784=	8786=	6727=	6728=	6729=	3300
U 1120		7281=	7284=	3405	7286=	7303=	7305=	7306=	3406
U 1130		7687=	7690=	7693=	7695=	8393=	8396=	8398=	8403=
U 1140		8422=	8426=	3554	8432=	8738=	8739=	8740=	3575
U 1150		8569=	8570=	3657=	3661=	8572=	3580	3697=	3701=
U 1160		3733=	3737=	3758=	3761=	3779=	3784=	3787=	3789=
U 1170		3793=	3794=	3946=	3959=	3962=	3965=	3991=	3993=
U 1200		3996=	4000=	4090=	4091=	4100=	4102=	4105=	4106=
U 1210		4123=	4124=	4129=	4130=	4132=	4133=	4153=	4154=
U 1220		4155=	4157=	4210=	4211=	4218=	4219=	4320=	4322=
U 1230		4347=	4350=	4368=	4372=	4374=	4376=	4390=	4392=
U 1240		4394=	4395=	4404=	4405=	4420=	4421=	4423=	4424=
U 1250		4427=	4428=	4429=	4430=	4438=	4439=	4454=	4455=
U 1260		4464=	4465=	4482=	4483=	4533=	4534=	4535=	4536=
U 1270		4542=	4543=	4552=	4553=	4556=	4557=	4581=	4584=
U 1300		4598=	4602=	4614=	4615=	4619=	4620=	4624=	4625=
U 1310		4640=	4642=	4646=	4647=	4942=	4945=	4961=	4963=
U 1320		4966=	4967=	4997=	4998=	5032=	5034=	5061=	5064=
U 1330		5097=	5099=	5125=	5132=	5136=	5139=	5141=	5143=
U 1340		5148=	5153=	5165=	5166=	5250=	5251=	5290=	5291=
U 1350		5294=	5295=	5299=	5300=	5319=	5320=	5349=	5350=
U 1360		5352=	5353=	5358=	5359=	5361=	5362=	5365=	5366=
U 1370		5390=	5391=	5394=	5395=	5397=	5398=	5405=	5406=

:
; Ucode Loc'n 0 1 2 3 4 5 6 7
Location / Line Number Index

U 1400	2228:	2229:	2569:	2572:	2575:	2579:	2589:	2689:
U 1410	2691:	2716:	2718:	2721:	2724:	2727:	2730:	2736:
U 1420	2738:	2740:	2743:	2750:	2745:	2747:	2752:	2754:
U 1430	2757:	3404:	2759:	2761:	2771:	8541	2772:	3420:
U 1440	3473:	2804:	2814:	2824:	2839:	2856:	2866:	2876:
U 1450	2886:	2896:	2906:	2917:	2927:	2937:	2947:	4085:
U 1460	7543:	7547:	3017:	7544:	3067:	3068:	3086:	5912:
U 1470	3139:	3140:	3269:	3271:	3274:	3277:	3370:	3388:
U 1500	8544	8548	8567	8593	2485:	2490:	2496:	8595
U 1510	8597	8626	8640	8698	2517:	2522:	2528:	2534:
U 1520	3525:	3526:	3528:	3529:	3530:	3533:	3534:	3535:
U 1530	3537:	3538:	3540:	3541:	3542:	3543:	3544:	3545:
U 1540	3601:	3651:	3501:	3678:	3691:	3720:	3753:	3511:
U 1550	3806:	3771:	3813:	8194:	3826:	3840:	3941:	4031:
U 1560	4058:	4072:	8722	8725	8732	2781:	4168:	2786:
U 1570	5318:	4145:	8734	8743	5346:	8746	5280:	5283:
U 1600	4316:	4325:	8756	8795	8816	8832	8834	
U 1610	4689:	3487:	2967:	2970:	7532:	4097:	5389:	7533:
U 1620	4697:	5410:	2981:	2983:	4701:	4702:	5426:	4386:
U 1630	4711:	5628:	3002:	3004:	4713:	5544:	5687:	5547:
U 1640	5113:	4118:	5115:	5118:	7558:			7560:
U 1650	7794:	7800:	7803:	7805:	7807:	7809:	7811:	7813:
U 1660	3931:	3919:	3921:	3923:	3925:	3933:	3927:	3929:
U 1670								
U 1700	7037:	7000:	7001:	7002:	7050:	7084:	7003:	7004:
U 1710	7006:	7007:	7008:	7009:	7332:	7329:	7010:	7011:
U 1720	7014:	7187:	7377:	7103:	7147:	7178:	7015:	7016:
U 1730	7019:	7020:	7021:	7022:	7023:	7024:	7025:	7026:
U 1740	5880:	5882:	5884:	5886:	5888:		5890:	5892:
U 1750	5894:	5896:			7825:	7833:		
U 1760	7209:	7211:	7213:	7215:	7292:	7324:	7217:	7218:
U 1770	7221:	7225:	7229:	7233:	7276:	7316:	7237:	7240:
U 2000	5413=	5414=	5435=	5436=	5439=	5440=	5446=	5447=
U 2010	5448=	5450=	5493=	5494=	5497=	5499=	5512=	5513=
U 2020	5514=	5515=	5552=	5554=	5557=	5559=	5563=	5564=
U 2030	5571=	5572=	5613=	5614=	5631=	5632=	5653=	5654=
U 2040	5674=	5675=	5677=	5679=	5694=	5695=	5704=	5705=
U 2050	5713=	5714=	5715=	5716=	5732=	5733=	5765=	5766=
U 2060	5770=	5771=	5775=	5777=	5802=	5803=	5810=	5811=
U 2070	5820=	5821=	5939=	5941=	5961=	5964=	5966=	5967=
U 2100	5957=	5983=	3613	5984=	5958=	3616	6009=	6011=
U 2110	6015=	6016=	6030=	6032=	6035=	6037=	6048=	6050=
U 2120	6059=	6095=	6060=	6097=	6162=	6096=	6163=	6098=
U 2130	6099=	6100=	6103=	6105=	6113=	6114=	6140=	6141=
U 2140	6147=	6150=	6165=	6168=	6186=	6190=	6200=	6201=
U 2150	6204=	6206=	6240=	6241=	6253=	6254=	6257=	6258=
U 2160	6261=	6262=	6267=	6269=	6270=	6272=	6280=	6281=
U 2170	6291=	6292=	6301=	6303=	6296=	6305=	6298=	6307=

Produced on Advanced Information Services Electronic Laser Printer, PKO/JES6, DTN: 233-7981

Ucode	Loc'n	0	1	2	3	4	5	6	7
U 2200		6309=	6310=	6315=	6318=	6328=	6329=	6333=	6334=
U 2210		6339=	6340=	6342=	6346=	6384=	6385=	6389=	6390=
U 2220		6396=	6397=	6398=	6399=	6400=	6401=	6418=	6419=
U 2230		6422=	6552=	6424=	6553=	6425=	6426=	6432=	6433=
U 2240		6438=	6440=	6454=	8339=	6459=	6460=	6455=	8340=
U 2250		6467=	6468=	6473=	6474=	6500=	6501=	6525=	6528=
U 2260		6542=	6543=	6572=	6573=	6593=	6595=	6617=	6619=
U 2270		6621=	6622=	6639=	6641=	6662=	6666=	6668=	6670=
U 2300		6697=	6698=	6751=	6752=	6756=	6758=	6802=	6804=
U 2310		6824=	6826=	6841=	6842=	6851=	6855=	6870=	6873=
U 2320		6875=	6887=	6876=	6888=	6927=	6928=	6935=	6936=
U 2330		6943=	6944=	6946=	6947=	6973=	6981=	6990=	6991=
U 2340		7058=	7059=	7060=	7061=	7063=	7064=	7065=	7066=
U 2350		7068=	7069=	7112=	7115=	7122=	7127=	7134=	7137=
U 2360		7149=	7152=	7155=	7156=	7180=	7181=	7188=	7191=
U 2370		7197=	7198=	7254=	7255=	7257=	7931=	7259=	7932=
U 2400		7265=	7266=	7297=	7298=	7333=	7334=	7335=	7336=
U 2410		7339=	7340=	7341=	7342=	7343=	7344=	7345=	7346=
U 2420		7347=	7348=	7380=	7382=	7388=	7389=	7393=	7395=
U 2430		7401=	7403=	7442=	7445=	7446=	7447=	7458=	7459=
U 2440		7469=	7470=	7486=	7488=	7490=	7492=	7494=	7496=
U 2450		7500=	7501=	7515=	7516=	7519=	7520=	7577=	7578=
U 2460		7586=	7589=	7596=	7598=	7602=	7604=	7615=	7616=
U 2470		7622=	7623=	7628=	7630=	7634=	7636=	7647=	7648=
U 2500		7651=	7653=	7656=	7657=	7666=	7667=	7673=	7674=
U 2510		7701=	7702=	7707=	7708=	7848=	7849=	7860=	7861=
U 2520		7864=	7868=	7921=	7923=	7934=	7936=	7935=	7937=
U 2530		7938=	7941=	7939=	7942=	7944=	7947=	7945=	7948=
U 2540		7950=	7953=	7951=	7954=	7955=	8350=	7956=	8351=
U 2550		8201=	8202=	8252=	8255=	8266=	8268=	8290=	8292=
U 2560		8295=	8298=	8303=	8305=	8325=	8327=	8345=	8347=
U 2570		8359=	8416=	8365=	8374=	8361=	8418=	8405=	8408=
U 2600		8412=	8413=	8436=	8438=	8439=	8443=	8463=	8465=
U 2610		8453=	3618	8455=	8484=	8470=	8474=	3620	8493=
U 2620		8498=	8500=	8504=	8507=	8514=	8516=	8532=	8534=
U 2630		8537=	8538=	8575=	8576=	8580=	8581=	8600=	8601=
U 2640		8605=	8606=	8611=	8612=	8615=	8616=	8629=	8630=
U 2650		8638=	8639=	8641=	8642=	8694=	8697=	8700=	8701=
U 2660		8705=	8707=	8711=	8713=	8717=	8718=	8727=	8730=
U 2670		8750=	8752=	8754=	8755=	8758=	8759=	8761=	3622
U 2700		8792=	8793=	8798=	8805=	8818=	8819=	8826=	8827=
U 2710		3642	3664	3686	3704	3723	3728	3742	3747
U 2720		3757	3774	3809	3815	3817	3821	3827	3841
U 2730		3843	3846	3943	3985	3986	3988	4002	4007
U 2740		4024	4027	4039	4041	4045	4093	4104	4119
U 2750		4127	4146	4147	4151	4169	4171	4188	4189
U 2760		4200	4209	4212	4213	4214	4215	4217	4223
U 2770		4224	4225	4232	4248	4251	4263	4318	4326

Location / Line Number Index

	Ucode	Loc'n	0	1	2	3	4	5	6	7
U 3000			4327	4329	4352	4354	4388	4397	4398	4401
U 3010			4406	4407	4409	4433	4436	4437	4440	4446
U 3020			4452	4453	4457	4459	4460	4462	4467	4468
U 3030			4470	4471	4477	4480	4485	4493	4532	4538
U 3040			4539	4545	4546	4547	4548	4549	4554	4558
U 3050			4559	4562	4592	4610	4612	4617	4622	4627
U 3060			4628	4629	4636	4638	4645	4716	4717	4736
U 3070			4737	4745	4747	4768	4772	4805	4814	4817
U 3100			4818	4819	4822	4823	4824	4891	4892	4895
U 3110			4898	4899	4900	4901	4903	4906	4909	4910
U 3120			4911	4913	4938	4948	4949	4952	4954	4956
U 3130			4975	4976	4977	4978	4984	4992	5022	5036
U 3140			5039	5040	5052	5054	5057	5066	5093	5100
U 3150			5101	5103	5105	5106	5120	5135	5158	5160
U 3160			5163	5171	5175	5177	5179	5207	5220	5222
U 3170			5223	5224	5226	5228	5229	5230	5232	5241
U 3200			5243	5244	5247	5248	5249	5254	5287	5288
U 3210			5289	5297	5321	5326	5329	5354	5355	5356
U 3220			5379	5402	5411	5412	5432	5516	5545	5550
U 3230			5565	5569	5570	5573	5578	5580	5585	5587
U 3240			5588	5601	5603	5618	5633	5634	5638	5646
U 3250			5652	5658	5672	5673	5676	5696	5700	5703
U 3260			5706	5707	5710	5725	5759	5762	5772	5800
U 3270			5805	5807	5813	5918	5920	5922	5923	5931
U 3300			5969	5980	5981	5986	6007	6013	6014	6017
U 3310			6029	6046	6053	6056	6057	6058	6061	6062
U 3320			6064	6070	6071	6088	6089	6115	6122	6123
U 3330			6124	6132	6133	6134	6135	6136	6137	6142
U 3340			6143	6156	6158	6160	6174	6199	6202	6225
U 3350			6243	6246	6248	6256	6259	6264	6266	6273
U 3360			6282	6293	6294	6313	6335	6337	6347	6348
U 3370			6350	6351	6352	6353	6354	6357	6363	6368
U 3400			6375	6381	6382	6383	6386	6387	6388	6403
U 3410			6407	6427	6429	6434	6435	6456	6457	6471
U 3420			6520	6521	6523	6545	6546	6582	6588	6589
U 3430			6590	6591	6609	6630	6632	6642	6645	6650
U 3440			6651	6652	6653	6654	6656	6659	6671	6674
U 3450			6676	6678	6679	6696	6737	6740	6745	6747
U 3460			6822	6844	6847	6849	6868	6886	6889	6891
U 3470			6898	6899	6913	6915	6920	6930	6931	6939
U 3500			6941	6949	6951	6953	6954	6955	6957	6960
U 3510			6964	6966	6984	6987	7028	7029	7031	7032
U 3520			7052	7053	7054	7056	7057	7062	7071	7073
U 3530			7076	7078	7081	7086	7089	7092	7093	7095
U 3540			7097	7106	7118	7130	7139	7140	7144	7153
U 3550			7154	7157	7174	7184	7194	7195	6908	6909
U 3560			7199	7222	7223	7226	7227	7230	7231	7234
U 3570			7235	7238	7239	7247	7249	7250	7251	7260

; T10KL.MCR[10,1141]

15:34 27-JULY-1984

MICRO 31(254)

KS10 MICROCODE V124, 27-JUL-84 Page 316

; Ucode Loc'n 0 1 2 3 4 5 6 7

Location / Line Number Index

U 3600	7262	7267	7268	7271	7278	7289	7293	7294
U 3610	7296	7300	7308	7309	7310	7317	7318	7320
U 3620	7337	7338	7360	7361	7362	7364	7370	7371
U 3630	7378	7383	7396	7405	7408	7428	7429	7430
U 3640	7431	7432	7433	7434	7437	7438	7439	7449
U 3650	7451	7454	7456	7457	7461	7484	7498	7502
U 3660	7505	7518	7567	7620	7649	7664	7665	7669
U 3670	7676	7683	7698	7714	7716	7718	7720	7722
U 3700	7724	7726	7728	7730	7732	7828	7835	7846
U 3710	7919	7925	7927	7930	7933	7940	7943	7946
U 3720	7949	7952	7957	7958	7959	7961	8195	8197
U 3730	8198	8199	8200	8209	8210	8211	8213	8239
U 3740	6094:	6451:	6155:	6288:	5954:	8240	8241	8245
U 3750	8248	6452:	8249	8260	5955:	8262	8277	8308
U 3760	8310	8433	8459	8508	8511	8512	8518	8520
U 3770	8525	8527	8529	8530	8531	8539	8540	8208:

No errors detected

End of microcode assembly

319 pages of listing

Used 48.24 seconds, 118 pages of core

Symbol table: 31P

Text strings: 9P

Loc'n assignment: 18P

Cross reference: 53P

