

# EAE

BASIC LOGIC TEST  
MD-11-DZKEB-A

EP-DZKEB-A-DL-A

NOV 1976

COPYRIGHT © 1976

**digital**

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames. The first column on the left contains frames with text, likely serving as a table of contents or index. The remaining columns contain frames with data, which appears to be organized in a grid format. The data is too small to read clearly but seems to consist of multiple columns of text or numbers.

.REM%

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZKEB-A
PRODUCT NAME:	EAE BASIC LOGIC TEST
DATE:	AUGUST 1976
MAINTAINER:	DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) BY DIGITAL EQUIPMENT CORPORATION 1970,1976

41  
40  
39  
38  
37  
36  
35  
34  
33  
32  
31  
30  
29  
28  
27  
26  
25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97

1. ABSTRACT  
THIS TEST IS TO BE USED AS AN EAE LOGIC TEST FOR THE PDP-11 WITH THE EAE OPTION. IT TESTS ALL THE FUNCTIONS OF THE EAE WITH SPECIFIC NUMBER COMBINATIONS.

2. REQUIREMENTS  
2.1 EQUIPMENT  
PDP-11 STANDARD COMPUTER WITH EAE OPTION WITH OR WITHOUT THE HARDWARE SWITCH REGISTER

2.2 STORAGE  
2.2.1 PROGRAM STORAGE - THE ROUTINE USES 8K MEMORY

3. LOADING PROCEDURE  
3.1 METHOD  
PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE  
4.1 CONTROL SWITCH SETTING  
STARTING AT SA 200 ALL SWITCHES SHOULD BE SET AS INDICATED. \*\*\*IF SOFTWARE SWITCH REGISTER IS SELECTED THE FOLLOWING WILL BE PRINTED;  
SWR=XXXXXX NEW=  
REFER TO SECTION 5.1.2 FOR MORE INFORMATION\*\*\*

4.2 STARTING ADDRESS OR ADDRESSES  
SA=200

4.3 PROGRAM AND/OR OPERATOR ACTION  
LOAD PROGRAM INTO MEMORY.  
LOAD STARTING ADDRESS  
LOAD ADDRESS.  
SET SWITCHES (SEE 5) ALL DOWN FOR WORSE CASE  
PRESS START.  
NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING IS PRINTED:  
SWR=XXXXXX NEW=  
(REFER TO SECTION 5.1 FOR OPERATOR OPTIONS)

THE PROGRAM WILL LOOP AND BELL WILL RING ONCE PER PASS OF THE PROGRAM. A MINIMUM OF TWO PASSES SHOULD

001

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MAY11 27(732) 03-NOV-76 15:27 PAGE 4

98

ALWAYS BE RUN.

99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 .. ALL SWITCHES DOWN WILL TEST ALL OF THE EAE AND PRINT OUT ON ERRORS AND CONTINUE IN TEST. (BELL WILL RING AT COMPLETION OF A PASS)

5.1.2 SWITCH SETTINGS ARE

- SW15 = 1 OR UP ... HALT ON ERROR
- SW14 = 1 OR UP ... SCOPE LOOP
- SW13 = 1 OR UP ... INHIBIT PRINTOUT
- SW12 = 1 OR UP ... INHIBIT TRACE TRAPPING
- SW11 = 1 OR UP ... INHIBIT ITERATION LOOP
- SW10 = 1 OR UP ... BELL ON ERROR
- 0 OR DOWN . BELL ON PASS COMPLETE
- SW01 = 1 OR UP ... INHIBIT MULTIPLY/DIVIDE TEST
- SW00 = 1 OR UP ... INHIBIT SHIFT/NORMALIZE TEST

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW=''' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
  - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
  - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190

## 5.2. SUBROUTINE ABSTRACTS

## 5.2.1 BEGIN SA 200

5.2.2 SCOPE  
-----

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 4000 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS. NOTE: SUPPORTS CONT-G ROUTINE

5.2.3 HLT  
---

IS A ROUTINE THAT PRINTS-OUT AN ADDRESS THAT TAGS THE FAILING SUBTEST, THE AC, MQ, AND SC AT THE TIME OF THE FAILURE. SUPPORTS CONT-G ROUTINE.

5.2.4 TRTRAP  
-----

THIS ROUTINE WILL ALLOW THE TRACE BIT TRAP TO BE SET AFTER FIRST LOOP OF THE PROGRAM. UNDER NORMAL TESTING THE TRACE BIT WILL BE SET ON ALTERNATE LOOPS OF THE PROGRAM. WHEN SET IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTION. THIS SEQUENCE IS CONTINUED TILL THE END OF THE PROGRAM LOOP IS REACHED.

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  
2395.2.5 TRAPCATCHER  
\*\*\*\*\*

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0, DESIGNED TO DETECT, AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

THE PRINCIPAL OF THIS ROUTINE IS: THE VECTOR ENTRANCE ADDRESS POINTS TO THE NEXT SEQUENTIAL WORD WHICH CONTAINS A HALT (0000). (THIS LOCATION IS ALSO THE STATUS FOR THAT VECTOR ENTRANCE, BUT THIS HAS NO EFFECT ON IT ALSO BEING THE NEXT INSTRUCTION).

IF A HALT OCCURS IN THE TRAP OR INTERRUPT VECTOR AREA, REGISTER SIX SHOULD BE EXAMINED TO DETERMINE ITS CONTENTS, THEN USE REGISTER SIX CONTENTS AS AN ADDRESS TO DETERMINE THE LOCATION THE PROGRAM WAS AT, WHEN THE INTERRUPT OR TRAP OCCURRED. (MEMORY AS SPECIFIED BY R6 CONTAINS THE PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE THE TRAP OCCURRED).

## 5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS WORSE CASE TESTING. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, HIT CONTINUE WITH SW14 UP TO LOOP ON ERROR, AND SW13 UP TO DELETE PRINTOUTS.

## 6. ERRORS

## 6.1 ERROR PRINTOUT

ARE IN A FOUR WORD FORMAT, THE 1ST IS PC+2 OF THE DETECTED ERROR, THE SECOND IS THE AC, THE THIRD IS THE MQ, AND THE LAST IS THE SC. THE LISTING WILL REFLECT THE TRUE ANSWER.

## 6.2 ERROR RECOVERY

RESTART AT 200

240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295

7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

NONE

8. MISCELLANEOUS

THIS PROGRAM SHOULD BE RUN IN CONJUNCTION WITH MAINDEC-11-DZKEC-A (EAE RANDOM EXERCIZER).

8.1 EXECUTION TIME

ABOUT 40 SECONDS WITH ALL SWITCHES DOWN

9. PROGRAM DESCRIPTION

THIS PROGRAM IS A STRAIGHT LINE TEST OF THE EAE FUNCTIONS STARTING WITH A TEST OF THE REGISTERS. THE TEST IS ACTUALLY A CLUSTER OF SUB-TESTS SEPERATED BY 'SCOPE'. THESE SUB-TESTS ARE EXECUTED 4000 TIMES BEFORE GOING ON TO THE NEXT TEST. SW11 INHIBITS THIS SO THAT EACH SUB-TEST IS EXECUTED ONLY ONCE PER PASS. SW14 CAUSES THE CURRENT SUB-TEST TO BE LOOPED ON.

THE PROGRAM STARTS OFF BY CHECKING THE REGISTERS FOR WRITABILITY. THE NEXT SECTION CHECKS OUT THE LOGICAL SHIFT INSTRUCTION. THIS SECTION IS THE BULK OF THE PROGRAM BECAUSE IT IS THE MOST BASIC TEST OF THE SHIFT REGISTER. THIS SECTION CHECKS THE LOGICAL SHIFTS FROM 0-16 TIMES OF 0'S, 1'S, AND ALTERNATE 0'S AND 1'S PLUS SPECIAL CASES OF BOTH LEFT AND RIGHT SHIFTING. THE REST OF THE PROGRAM TESTS SPECIAL CASES OF ARITHMETIC SHIFT, NORMALIZE, MULTIPLY, AND DIVIDE.

10. LISTING

FOLLOWING

11. FLOW CHART(S)

NA

%

\*\*\*\*\*  
\*\*\*\*\*  
;TITLE MAINDEC-11-DZKEB-A  
;COPYRIGHT (C) 1970,1976  
;DIGITAL EQUIPMENT CORP.



296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322

```

;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY BOB BRAIN
;*
;*
;*MODIFIED BY ALAN BOSTICK, JULY 1976 TO SUPPORT
;*SOFTWARE SWITCH REGISTER.
;*ALSO ALLOWS DYNAMIC LOADING OF SOFTWARE SWITCH
;*REGISTER FROM TTY
;*****
;
;      SWITCH SETTINGS
;
;      SWITCH          USE
;      -----
;      00             INHIBIT SHIFT AND NORMALIZE TEST
;      01             INHIBIT MULTIPLY AND DIVIDE TEST
;      10             0 - BELL ON PASS COMPLETE
;                   1 - BELL ON ERROR
;      11             INHIBIT ITERATIONS
;      12             INHIBIT TRACE TRAP
;      13             INHIBIT TYPEOUT
;      14             LOOP ON TEST
;      15             HALT ON ERROR
;*****
;*****

```

```

323      000000      . = 0
324      000030      . = 30
325      000030      016334      PRINT
326      000032      000340      340
327      000034      000034      . = 34
328      000034      017202      SCOPEC
329      000036      000000      0
330      104000      HLT = EMT
331      000040      177570      SWR: 177570
332      177776      PSW= 177776
333      000240      NOP= 240
334      104400      SCOPE= TRAP
335      000176      . = 176
336      000000      SWREG: 0
337      000200      . = 200
338      000200      012706      017334      MOV #BUFF,%6 ;SET UP STACK FOR SCOPE LOOP
339      000204      005005      CLR %5
340      000206      012705      020023      MOV #SHEAD,%5 ;TYPE PROGRAM NAME
341      000212      004767      017634      JSR %7,TTOUT
342      000216      012705      017774      MOV #SMAIN,%5 ; TYPE MAINDEC NUMBER
343      000222      004767      017624      JSR %7,TTOUT
344      000226      000167      000022      JMP SUSWR
345      000232      177300      DIV: 177300
346      000234      177302      AC: 177302
347      000236      177304      MQ: 177304
348      000240      177306      MUL: 177306
349      000242      177310      SC: 177310
350      000244      177311      SR: 177311
351      000246      177312      NOR: 177312
352      000250      177314      LSH: 177314
353      000252      177316      ASH: 177316
354
355
356
357
358      ; *****
359      ; TEST FOR HARDWARE SWITCH REGISTER
360      ; *****
361      000254      013746      000006      SUSWR: MOV @#6,-(6) ;SAVE VECTORS
362      000260      013746      000004      MOV @#4,-(6)
363      000264      012737      000304      000004      MOV #64$,@#4 ;SET UP FOR TIMEOUT
364      000272      022777      177777      177540      CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
365      000300      001402      BEQ 65$
366      000302      000404      BR 66$
367      000304      022626      64$: CMP (6)+,(6)+ ;ADJUST STACK
368      000306      012767      000176      177524      65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
369      000314      012637      000004      66$: MOV (6)+,@#4 ;RESTORE VECTORS
370      000320      012637      000006      MOV (6)+,@#6
371      000324      022767      000176      177506      CMP #SWREG,SWR ;IS SWREG USED
372      000332      001002      BNE BEGIN
373      000334      004767      017062      JSR %7,CNTLU ;ALLOW SWREG TO BE LOADED
374      000340      012767      000340      016716      BEGIN: MOV #BEGIN,RETURN ;SET UP RESTART OF PROGRAM
375      000346      005077      177662      CLR @AC
376      000352      005077      177660      CLR @MQ
377      000356      005003      CLR %3

```

```

378 ;:*****
379 ;: REGISTER TEST FOR WRITABILITY
380 ;:*****
381 000360 REGWRI:
382 000360 104400 SCOPE ;CHECK SR AND SC FLOPS FOR 0 STATE
383 000362 012777 000000 177652 MOV #0,ASC
384 000370 105777 177646 TST ASC
385 000374 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT
386 000376 104000 HLT ;CALL ERROR ROUTINE
387 000400 64$:
388
389 000400 104400 SCOPE ;CHECK SR AND SC FLOPS FOR 1 STATE
390 000402 012777 140477 177632 MOV #140477,ASC
391 000410 017746 177626 MOV ASC, -(6)
392 000414 042716 037300 BIC #37300, (6)
393 000420 022726 140477 CMP #140477, (6)+
394 000424 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT
395 000426 104000 HLT ;CALL ERROR ROUTINE
396 000430 65$:
397 000430 005077 177606 CLR ASC
398
399 000434 104400 SCOPE ;CHECK MQ FLOPS FOR 0 STATE
400 000436 012777 000000 177572 MOV #0,AMQ
401 000444 005777 177566 TST AMQ
402 000450 001401 BEQ 66$ ;IF NO.ERROR SKIP HLT
403 000452 104000 HLT ;CALL ERROR ROUTINE
404 000454 66$:
405
406 000454 122777 000036 177562 CMPB #36,ASR ;CHECK STATUS 36
407 000462 001401 BEQ 67$ ;IF NO.ERROR SKIP HLT
408 000464 104000 HLT ;CALL ERROR ROUTINE
409 000466 67$:
410
411 000466 104400 SCOPE ;CHECK MQ FLOPS FOR 1 STATE
412 000470 012777 177777 177540 MOV #-1,AMQ
413 000476 022777 177777 177532 CMP #-1,AMQ
414 000504 001401 BEQ 68$ ;IF NO.ERROR SKIP HLT
415 000506 104000 HLT ;CALL ERROR ROUTINE
416 000510 68$:
417
418 000510 122777 000042 177526 CMPB #42,ASR ;CHECK STATUS 42
419 000516 001401 BEQ 69$ ;IF NO.ERROR SKIP HLT
420 000520 104000 HLT ;CALL ERROR ROUTINE
421 000522 69$:
422 000522 104400 SCOPE ;CHECK AC FLOPS FOR 0 STATE
423 000524 012777 000000 177502 MOV #0,ASC
424 000532 005777 177476 TST ASC
425 000536 001401 BEQ 70$ ;IF NO.ERROR SKIP HLT
426 000540 104000 HLT ;CALL ERROR ROUTINE
427 000542 70$:
428
429 000542 122777 000020 177474 CMPB #20,ASR ;CHECK STATUS 20
430 000550 001401 BEQ 71$ ;IF NO.ERROR SKIP HLT
431 000552 104000 HLT ;CALL ERROR ROUTINE
432 000554 71$:
433

```

L01

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 12

```

434 000554 104400          SCOPE                      ;CHECK AC FLOPS FOR 1 STATE
435 000556 012777 177777 177450      MOV          #-1, @AC
436 000564 022777 177777 177442      CMP          #-1, @AC
437 000572 001401          BEQ          72$
438 000574 104000          HLT
439 000576          72$:
440 000576 122777 000042 177440      CMPB        #42, @SR
441 000604 001401          BEQ          73$
442 000606 104000          HLT
443 000610          73$:
444
445 000610 104400          SCOPE                      ;CHECK AC AND MQ WITH ALL NUMBERS
446 000612 005067 016520          CLR          CP
447 000616 005267 016514          CP1:        INC          CP
448 000622 001420          BEQ          OUTPC
449 000624 016777 016506 177404      MOV          CP, @MQ
450 000632 026777 016500 177376      CMP          CP, @MQ
451 000640 001401          BEQ          64$
452 000642 104000          HLT
453 000644          64$:
454 000644 016777 016466 177362      MOV          CP, @AC
455 000652 026777 016460 177354      CMP          CP, @AC
456 000660 001756          BEQ          CP1
457 000662 104000          HLT
458 000664 012767 004000 016370  OUTPC:  MOV          #4000, SCOPEF
459 000672 104400          SCOPE
460 000674 012777 177777 177334      MOV          #-1, @MQ
461 000702 022777 177777 177324      CMP          #-1, @AC
462 000710 001401          BEQ          64$
463 000712 104000          HLT
464 000714          64$:
465
466 000714 005077 177316          CLR          @MQ
467 000720 005777 177310          TST         @AC
468 000724 001401          BEQ          65$
469 000726 104000          HLT
470 000730          65$:
471
472 000730 112777 177777 177300      MOVB        #-1, @MQ
473 000736 022777 177777 177272      CMP          #-1, @MQ
474 000744 001401          BEQ          66$
475 000746 104000          HLT
476 000750          66$:
477 000750 022777 177777 177256      CMP          #-1, @AC

```

```

;CHECK AC FLOPS FOR 1 STATE
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;CHECK STATUS 42
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;CHECK AC AND MQ WITH ALL NUMBERS
;FINISHED WHEN CP=0
;LOAD MQ
;TEST MQ
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;LOAD AC
;TEST AC
;HALT IF AC.NE.CP
;NO ITERATIONS
;TEST OF SIGN EXTENTION
;LOAD MQ WITH -1
;TEST FOR SIGN EXTENTION
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;CHECK FOR ZERO SIGN EXTENTION
;CHECK FOR ZERO AC
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;TEST OF BYTE SIGN EXTENTION
;CHECK FOR SIGN EXTENTION IN MQ
; IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;CHECK FOR SIGN EXTENTION IN AC

```

```

478 000756 001401          BEQ      67$          ; IF NO.ERROR SKIP HLT
479 000760 104000          HLT                               ; CALL ERROR ROUTINE
480 000762                67$:
481
482 000762 105077 177250    CLRB    2MQ          ; CHECK FOR BYTE ZERO SIGN EXTENTION
483 000766 005777 177244    TST     2MQ          ; CHECK FOR ZERO MQ
484 000772 001401          BEQ      68$          ; IF NO.ERROR SKIP HLT
485 000774 104000          HLT                               ; CALL ERROR ROUTINE
486 000776                68$:
487 000776 005777 177232    TST     2AC          ; CHECK FOR ZERO AC
488 001002 001401          BEQ      69$          ; IF NO.ERROR SKIP HLT
489 001004 104000          HLT                               ; CALL ERROR ROUTINE
490 001006                69$:
491
492 001006 012777 100000 177222  MOV     #100000,2MQ    ; LOAD MQ WITH LARGEST NUMBER
493 001014 022777 177777 177212  CMP     #-1,2AC       ; DID IT SIGN EXTEND
494 001022 001401          BEQ      70$          ; IF NO.ERROR SKIP HLT
495 001024 104000          HLT                               ; CALL ERROR ROUTINE
496 001026                70$:
497 001026 112777 000200 177202  MOVB   #200,2MQ      ; LOAD MQ WITH LARGEST BYTE
498 001034 022777 177777 177172  CMP     #-1,2AC       ; DID IT SIGN EXTEND
499 001042 001401          BEQ      71$          ; IF NO.ERROR SKIP HLT
500 001044 104000          HLT                               ; CALL ERROR ROUTINE
501 001046                71$:
502 001046 022777 177600 177162  CMP     #177600,2MQ   ; DID IT SIGN EXTEND
503 001054 001401          BEQ      72$          ; IF NO.ERROR SKIP HLT
504 001056 104000          HLT                               ; CALL ERROR ROUTINE
505 001060                72$:
506
507 001060 012777 000077 177154  MOV     #77,2SC       ; LOAD SC WITH -1
508 001066 022777 000077 177152  CMP     #77,2NOR      ; CHECK FOR SIGN EXTENTION
509 001074 001401          BEQ      73$          ; IF NO.ERROR SKIP HLT
510 001076 104000          HLT                               ; CALL ERROR ROUTINE
511 001100                73$:
512
513 001100 005077 177136    CLR     2SC          ; CLEAR SC
514 001104 005777 177136    TST     2NOR         ; CHECK NOR
515 001110 001401          BEQ      74$          ; IF NO.ERROR SKIP HLT
516 001112 104000          HLT                               ; CALL ERROR ROUTINE
517 001114                74$:
518
519
520 ; *****
521 ; AT THIS POINT, ALL THE REGISTERS CAN HANDLE DATA OF
522 ; ANY FORM
523 ; *****
523 001114 004767 016230    JSR     %7,CKSWR      ; CHECK FOR CONT-G
524 001120 032777 000001 176712  BIT     #1,2SWR
525 001126 001402          BEQ     MQ0L
526 001130 000167 011742    JMP     .DIV
527

```

```

528      ;*****
529      ; LOGICAL SHIFT TEST SECTION
530      ; TEST MQ SHIFT OF 0'S LEFT
531      ;*****
532      001134      104400      MQOL:      SCOPE
533      001134      005077      177072      CLR      @AC
534      001136      005077      177070      CLR      @MQ
535      001142      012700      177777      MOV      #-1,%0
536      001146      005200      LOOP:      INC      %0
537      001152      005077      177062      CLR      @SC      ;CLEAR SR AND SC
538      001160      010077      177064      MOV      %0,@LSH      ;SHIFT R0 TIMES LEFT
539      001164      005777      177046      TST      @MQ      ;TEST MQ FOR '0'
540      001170      001401      BEQ      64$      ;IF NO.ERROR SKIP HLT
541      001172      104000      HLT      ;CALL ERROR ROUTINE
542      001174      122777      000036      177042      64$:      CMPB     #36,@SR      ;CHECK STATUS REGISTER
543      001202      001401      BEQ      65$      ;IF NO.ERROR SKIP HLT
544      001204      104000      HLT      ;CALL ERROR ROUTINE
545      001206      022700      000020      65$:      CMP      #-16,%0      ;LAST ONE
546      001212      001357      BNE      LOOP
547
548
549
550
551      ;*****
552      ; TEST AC SHIFT OF 0'S RIGHT
553      ;*****
554
555      001214      104400      SCOPE
556      001216      012700      000001      MOV      #1,%0
557      001222      005300      LOOP1:  DEC      %0
558      001224      005077      177012      CLR      @SC      ;CLEAR SR AND SC
559      001230      010077      177014      MOV      %0,@LSH      ;SHIFT R0 TIMES RIGHT
560      001234      005777      176774      TST      @AC      ;TEST AC FOR '0'
561      001240      001401      BEQ      64$      ;IF NO.ERROR SKIP HLT
562      001242      104000      HLT      ;CALL ERROR ROUTINE
563      001244      122777      000036      176772      64$:      CMPB     #36,@SR      ;CHECK STATUS REGISTER
564      001252      001401      BEQ      65$      ;IF NO.ERROR SKIP HLT
565      001254      104000      HLT      ;CALL ERROR ROUTINE
566      001256      022700      177760      65$:      CMP      #-16,%0      ;LAST ONE
567      001262      001357      BNE      LOOP1
568
569

```

```

570
571
572
573
574 001264
575 001264 104400
576 001266 012777 177777 176742
577 001274 005077 176734
578 001300 005077 176744
579 001304 027727 176726 177777
580 001312 001401
581 001314 104000
582 001316
583 001316 122777 000020 176720
584 001324 001401
585 001326 104000
586 001330
587 001330 104400
588 001332 012777 177777 176676
589 001340 005077 176670
590 001344 012777 177777 176676
591 001352 122777 077777 176656
592 001360 001401
593 001362 104000
594 001364
595 001364 122777 000023 176652
596 001372 001401
597 001374 104000
598 001376
599 001376 104400
600 001400 012777 177777 176630
601 001406 005077 176622
602 001412 012777 177776 176630
603 001420 022777 037777 176610
604 001426 001401
605 001430 104000
606 001432
607 001432 122777 000023 176604
608 001440 001401
609 001442 104000
610 001444
611
612 001444 104400
613 001446 012777 177777 176562
614 001454 005077 176554
615 001460 012777 177775 176562
616 001466 022777 017777 176542
617 001474 001401
618 001476 104000
619 001500
620 001500 122777 000023 176536
621 001506 001401
622 001510 104000
623 001512
624 001512 104400
625 001514 012777 177777 176514

```

```

*****
TEST MQ SHIFT OF 1'S RIGHT
*****

```

```

MQ1R:
SCOPE
MOV # -1, MQ ;SET MQ=-1
CLR JAC ;CLEAR AC
CLR JLSH ;INITIALIZE SHIFT BY 0
CMP JMQ, # -1 ;COMPARE UNSHIFTED MQ TO -1
BEQ 64S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

64S:
CMPB #20, JSR ;CHECK STATUS
BEQ 65S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

65S:
SCOPE
MOV # -1, MQ ;RESET MQ
CLR JAC
MOV #177777, JLSH ;SHIFT 177777 TIMES RIGHT
CMP #77777, JMQ ;CHECK MQ
BEQ 66S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

66S:
CMPB #23, JSR ;CHECK STATUS
BEQ 67S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

67S:
SCOPE
MOV # -1, MQ ;RESET MQ
CLR JAC
MOV #177776, JLSH ;SHIFT 177776 TIMES RIGHT
CMP #37777, JMQ ;CHECK MQ
BEQ 68S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

68S:
CMPB #23, JSR ;CHECK STATUS
BEQ 69S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

69S:
SCOPE
MOV # -1, MQ ;RESET MQ
CLR JAC
MOV #177775, JLSH ;SHIFT 177775 TIMES RIGHT
CMP #17777, JMQ ;CHECK MQ
BEQ 70S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

70S:
CMPB #23, JSR ;CHECK STATUS
BEQ 71S ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE

71S:
SCOPE
MOV # -1, MQ ;RESET MQ

```

626	001522	005077	176506		CLR	0AC	
627	001526	012777	177774	176514	MOV	#177774, 0LSH	;SHIFT 177774 TIMES RIGHT
628	001534	022777	007777	176474	CMP	#7777, 0MQ	;CHECK MQ
629	001542	001401			BEQ	72S	;IF NO.ERROR SKIP HLT
630	001544	104000			HLT		;CALL ERROR ROUTINE
631	001546						72S:
632	001546	122777	000023	176470	CMPB	#23, 0SR	;CHECK STATUS
633	001554	001401			BEQ	73S	;IF NO.ERROR SKIP HLT
634	001556	104000			HLT		;CALL ERROR ROUTINE
635	001560						73S:
636							
637	001560	104400			SCOPE		
638	001562	012777	177777	176446	MOV	#-1, 0MQ	;RESET MQ
639	001570	005077	176440		CLR	0AC	
640	001574	012777	177773	176446	MOV	#177773, 0LSH	;SHIFT 177773 TIMES RIGHT
641	001602	000241			CLC		
642	001604	022777	003777	176424	CMP	#3777, 0MQ	;CHECK MQ
643	001612	001401			BEQ	74S	;IF NO.ERROR SKIP HLT
644	001614	104000			HLT		;CALL ERROR ROUTINE
645	001616						74S:
646	001616	122777	000023	176420	CMPB	#23, 0SR	;CHECK STATUS
647	001624	001401			BEQ	75S	;IF NO.ERROR SKIP HLT
648	001626	104000			HLT		;CALL ERROR ROUTINE
649	001630						75S:
650							
651	001630	104400			SCOPE		
652	001632	012777	177777	176376	MOV	#-1, 0MQ	;RESET MQ
653	001640	005077	176370		CLR	0AC	
654	001644	012777	177772	176376	MOV	#177772, 0LSH	;SHIFT 177772 TIMES RIGHT
655	001652	022777	001777	176356	CMP	#1777, 0MQ	;CHECK MQ
656	001660	001401			BEQ	76S	;IF NO.ERROR SKIP HLT
657	001662	104000			HLT		;CALL ERROR ROUTINE
658	001664						76S:
659	001664	122777	000023	176352	CMPB	#23, 0SR	;CHECK STATUS
660	001672	001401			BEQ	77S	;IF NO.ERROR SKIP HLT
661	001674	104000			HLT		;CALL ERROR ROUTINE
662	001676						77S:
663	001676	104400			SCOPE		
664	001700	012777	177777	176330	MOV	#-1, 0MQ	;RESET MQ
665	001706	005077	176322		CLR	0AC	
666	001712	012777	177771	176330	MOV	#177771, 0LSH	;SHIFT 177771 TIMES RIGHT
667	001720	022777	000777	176310	CMP	#777, 0MQ	;CHECK MQ
668	001726	001401			BEQ	78S	;IF NO.ERROR SKIP HLT
669	001730	104000			HLT		;CALL ERROR ROUTINE
670	001732						78S:
671	001732	122777	000023	176304	CMPB	#23, 0SR	;CHECK STATUS



672	001740	001401			BEQ	79\$			: IF NO.ERROR SKIP HLT
673	001742	104000			HLT				:CALL ERROR ROUTINE
674	001744				79\$:				
675	001744	104400			SCOPE				
676	001746	012777	177777	176262	MOV	#-1,AMQ			:RESET MQ
677	001754	005077	176254		CLR	QAC			
678	001760	012777	177770	176262	MOV	#177770,ALSH			:SHIFT 177770 TIMES RIGHT
679	001766	022777	000377	176242	CMP	#377,AMQ			:CHECK MQ
680	001774	001401			BEQ	80\$			: IF NO.ERROR SKIP HLT
681	001776	104000			HLT				:CALL ERROR ROUTINE
682	002000				80\$:				
683	002000	122777	000023	176236	CMPB	#23,ASR			:CHECK STATUS
684	002006	001401			BEQ	81\$			: IF NO.ERROR SKIP HLT
685	002010	104000			HLT				:CALL ERROR ROUTINE
686	002012				81\$:				
687									
688	002012	104400			SCOPE				
689	002014	012777	177777	176214	MOV	#-1,AMQ			:RESET MQ
690	002022	005077	176206		CLR	QAC			
691	002026	012777	177767	176214	MOV	#177767,ALSH			:SHIFT 177767 TIMES RIGHT
692	002034	022777	000177	176174	CMP	#177,AMQ			:CHECK MQ
693	002042	001401			BEQ	82\$			: IF NO.ERROR SKIP HLT
694	002044	104000			HLT				:CALL ERROR ROUTINE
695	002046				82\$:				
696	002046	122777	000023	176170	CMPB	#23,ASR			:CHECK STATUS
697	002054	001401			BEQ	83\$			: IF NO.ERROR SKIP HLT
698	002056	104000			HLT				:CALL ERROR ROUTINE
699	002060				83\$:				
700									
701	002060	104400			SCOPE				
702	002062	012777	177777	176146	MOV	#-1,AMQ			:RESET MQ
703	002070	005077	176140		CLR	QAC			
704	002074	012777	177766	176146	MOV	#177766,ALSH			:SHIFT 177766 TIMES RIGHT
705	002102	022777	000077	176126	CMP	#77,AMQ			:CHECK MQ
706	002110	001401			BEQ	84\$			: IF NO.ERROR SKIP HLT
707	002112	104000			HLT				:CALL ERROR ROUTINE
708	002114				84\$:				
709	002114	122777	000023	176122	CMPB	#23,ASR			:CHECK STATUS
710	002122	001401			BEQ	85\$			: IF NO.ERROR SKIP HLT
711	002124	104000			HLT				:CALL ERROR ROUTINE
712	002126				85\$:				
713									
714	002126	104400			SCOPE				
715	002130	012777	177777	176100	MOV	#-1,AMQ			:RESET MQ
716	002136	005077	176072		CLR	QAC			
717	002142	012777	177765	176100	MOV	#177765,ALSH			:SHIFT 177765 TIMES RIGHT
718	002150	022777	000037	176060	CMP	#37,AMQ			:CHECK MQ
719	002156	001401			BEQ	86\$			: IF NO.ERROR SKIP HLT
720	002160	104000			HLT				:CALL ERROR ROUTINE
721	002162				86\$:				
722	002162	122777	000023	176054	CMPB	#23,ASR			:CHECK STATUS

E02

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 18

723	002170	001401			BEG	87\$			; IF NO.ERROR SKIP HLT
724	002172	104000			HLT				; CALL ERROR ROUTINE
725	002174					87\$:			
726	002174	104400			SCOPE				
727	002176	012777	177777	176032	MOV	#-1,AMQ			; RESET MQ
728	002204	005077	176024		CLR	DAC			
729	02210	012777	177764	176032	MOV	#177764,ALSH			; SHIFT 177764 TIMES RIGHT
730	002216	022777	000017	176012	CMP	#17,AMQ			; CHECK MQ
731	002224	001401			BEG	88\$			; IF NO.ERROR SKIP HLT
732	002226	104000			HLT				; CALL ERROR ROUTINE
733	002230					88\$:			
734	002230	122777	000023	176006	CMPB	#23,ASR			; CHECK STATUS
735	002236	001401			BEG	89\$			; IF NO.ERROR SKIP HLT
736	002240	104000			HLT				; CALL ERROR ROUTINE
737	002242					89\$:			
738									
739	002242	104400			SCOPE				
740	002244	012777	177777	175764	MOV	#-1,AMQ			; RESET MQ
741	002252	005077	175756		CLR	DAC			
742	002256	012777	177763	175764	MOV	#177763,ALSH			; SHIFT 177763 TIMES RIGHT
743	002264	022777	000007	175744	CMP	#7,AMQ			; CHECK MQ
744	002272	001401			BEG	90\$			; IF NO.ERROR SKIP HLT
745	002274	104000			HLT				; CALL ERROR ROUTINE
746	002276					90\$:			
747	002276	122777	000023	175740	CMPB	#23,ASR			; CHECK STATUS
748	002304	001401			BEG	91\$			; IF NO.ERROR SKIP HLT
749	002306	104000			HLT				; CALL ERROR ROUTINE
750	002310					91\$:			
751									
752	002310	104400			SCOPE				
753	002312	012777	177777	175716	MOV	#-1,AMQ			; RESET MQ
754	002320	005077	175710		CLR	DAC			
755	002324	012777	177762	175716	MOV	#177762,ALSH			; SHIFT 177762 TIMES RIGHT
756	002332	022777	000003	175676	CMP	#3,AMQ			; CHECK MQ
757	002340	001401			BEG	92\$			; IF NO.ERROR SKIP HLT
758	002342	104000			HLT				; CALL ERROR ROUTINE
759	002344					92\$:			
760	002344	122777	000023	175672	CMPB	#23,ASR			; CHECK STATUS
761	002352	001401			BEG	93\$			; IF NO.ERROR SKIP HLT
762	002354	104000			HLT				; CALL ERROR ROUTINE
763	002356					93\$:			
764									
765	002356	104400			SCOPE				
766	002360	012777	177777	175650	MOV	#-1,AMQ			; RESET MQ
767	002366	005077	175642		CLR	DAC			
768	002372	012777	177761	175650	MOV	#177761,ALSH			; SHIFT 177761 TIMES RIGHT
769	002400	022777	000001	175630	CMP	#1,AMQ			; CHECK MQ
770	002406	001401			BEG	94\$			; IF NO.ERROR SKIP HLT
771	002410	104000			HLT				; CALL ERROR ROUTINE
772	002412					94\$:			
773	002412	122777	000023	175624	CMPB	#23,ASR			; CHECK STATUS
774	002420	001401			BEG	95\$			; IF NO.ERROR SKIP HLT
775	002422	104000			HLT				; CALL ERROR ROUTINE
776	002424					95\$:			

```

777
778
779
780 002424
781 002424 104400
782 002426 005077 175604
783 002432 012777 177777 175574
784 002440 005077 175604
785 002444 027727 175564 177777
786 002452 001401
787 002454 104000
788 002456
789 002456 122777 000350 175560
790 002464 001401
791 002466 104000
792 002470
793 002470 104400
794 002472 012777 177777 175534
795 002500 012777 000001 175542
796 002506 022777 177776 175520
797 002514 001401
798 002516 104000
799 002520
800 002520 122777 000311 175516
801 002526 001401
802 002530 104000
803 002532
804
805
806 002532 104400
807 002534 012777 177777 175472
808 002542 012777 000002 175500
809 002550 022777 177774 175456
810 002556 001401
811 002560 104000
812 002562
813 002562 122777 000311 175454
814 002570 001401
815 002572 104000
816 002574
817
818
819 002574 104400
820 002576 012777 177777 175430
821 002604 012777 000003 175436
822 002612 022777 177770 175414
823 002620 001401
824 002622 104000
825 002624
826 002624 122777 000311 175412
827 002632 001401
828 002634 104000
829 002636
830 002636 104400
831 002640 012777 177777 175366
832 002646 012777 000004 175374

```

;\*\*\*\*\*  
 ; TEST AC SHIFT OF 1'S LEFT  
 ;\*\*\*\*\*  
 ACIL:

```

SCOPE
CLR      2MQ
MOV      #-1,2AC      ;SET AC=-1
CLR      2LSH         ;INITIALIZE SHIFT BY 0
CMP      2AC, #-1     ;COMPARE UNSHIFTED AC TO -1
BEQ      64$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

64$:
CMPB     #350,2SR     ;CHECK STATUS
BEQ      65$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

65$:
SCOPE
MOV      #-1,2AC      ;SET AC=-1
MOV      #1,2LSH     ;SHIFT 1 TIMES LEFT
CMP      #177776,2AC ;CHECK AC
BEQ      66$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

66$:
CMPB     #311,2SR     ;CHECK STATUS
BEQ      67$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

67$:
SCOPE
MOV      #-1,2AC      ;SET AC=-1
MOV      #2,2LSH     ;SHIFT 2 TIMES LEFT
CMP      #177774,2AC ;CHECK AC
BEQ      68$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

68$:
CMPB     #311,2SR     ;CHECK STATUS
BEQ      69$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

69$:
SCOPE
MOV      #-1,2AC      ;SET AC=-1
MOV      #3,2LSH     ;SHIFT 3 TIMES LEFT
CMP      #177770,2AC ;CHECK AC
BEQ      70$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

70$:
CMPB     #311,2SR     ;CHECK STATUS
BEQ      71$         ;IF NO.ERROR SKIP HLT
HLT      ;CALL ERROR ROUTINE

71$:
SCOPE
MOV      #-1,2AC      ;SET AC=-1
MOV      #4,2LSH     ;SHIFT 4 TIMES LEFT

```

833	002654	022777	177760	175352	CMP	#177760, @AC		;CHECK AC
834	002662	001401			BEQ	72\$		; IF NO.ERROR SKIP HLT
835	002664	104000			HLT			;CALL ERROR ROUTINE
836	002666						72\$:	
837	002666	122777	000311	175350	CMPB	#311, @SR		;CHECK STATUS
838	002674	001401			BEQ	73\$		; IF NO.ERROR SKIP HLT
839	002676	104000			HLT			;CALL ERROR ROUTINE
840	002700						73\$:	
841								
842								
843	002700	104400			SCOPE			
844	002702	012777	177777	175324	MOV	#-1, @AC		;SET AC=-1
845	002710	012777	000005	175332	MOV	#5, @LSH		;SHIFT 5 TIMES LEFT
846	002716	022777	177740	175310	CMP	#177740, @AC		;CHECK AC
847	002724	001401			BEQ	74\$		; IF NO.ERROR SKIP HLT
848	002726	104000			HLT			;CALL ERROR ROUTINE
849	002730						74\$:	
850	002730	122777	000311	175306	CMPB	#311, @SR		;CHECK STATUS
851	002736	001401			BEQ	75\$		; IF NO.ERROR SKIP HLT
852	002740	104000			HLT			;CALL ERROR ROUTINE
853	002742						75\$:	
854								
855	002742	104400			SCOPE			
856	002744	012777	177777	175262	MOV	#-1, @AC		;SET AC=-1
857	002752	012777	000006	175270	MOV	#6, @LSH		;SHIFT 6 TIMES LEFT
858	002760	022777	177700	175246	CMP	#177700, @AC		;CHECK AC
859	002766	001401			BEQ	76\$		; IF NO.ERROR SKIP HLT
860	002770	104000			HLT			;CALL ERROR ROUTINE
861	002772						76\$:	
862	002772	122777	000311	175244	CMPB	#311, @SR		;CHECK STATUS
863	003000	001401			BEQ	77\$		; IF NO.ERROR SKIP HLT
864	003002	104000			HLT			;CALL ERROR ROUTINE
865	003004						77\$:	
866								
867								
868	003004	104400			SCOPE			
869	003006	012777	177777	175220	MOV	#-1, @AC		;SET AC=-1
870	003014	012777	000007	175226	MOV	#7, @LSH		;SHIFT 7 TIMES LEFT
871	003022	022777	177600	175204	CMP	#177600, @AC		;CHECK AC
872	003030	001401			BEQ	78\$		; IF NO.ERROR SKIP HLT
873	003032	104000			HLT			;CALL ERROR ROUTINE
874	003034						78\$:	
875	003034	122777	000311	175202	CMPB	#311, @SR		;CHECK STATUS

876	003042	001401			BEQ	79\$			; IF NO.ERROR SKIP HLT
877	003044	104000			HLT				; CALL ERROR ROUTINE
878	003046						79\$:		
879	003046	104400			SCOPE				
880	003050	012777	177777	175156	MOV	#-1, @AC			; SET AC=-1
881	003056	012777	000010	175164	MOV	#10, @LSH			; SHIFT 10 TIMES LEFT
882	003064	022777	177400	175142	CMP	#177400, @AC			; CHECK AC
883	003072	001401			BEQ	80\$			; IF NO.ERROR SKIP HLT
884	003074	104000			HLT				; CALL ERROR ROUTINE
885	003076						80\$:		
886	003076	122777	000311	175140	CMPB	#311, @SR			; CHECK STATUS
887	003104	001401			BEQ	81\$			; IF NO.ERROR SKIP HLT
888	003106	104000			HLT				; CALL ERROR ROUTINE
889	003110						81\$:		
890									
891									
892	003110	104400			SCOPE				
893	003112	012777	177777	175114	MOV	#-1, @AC			; SET AC=-1
894	003120	012777	000011	175122	MOV	#11, @LSH			; SHIFT 11 TIMES LEFT
895	003126	022777	177000	175100	CMP	#177000, @AC			; CHECK AC
896	003134	001401			BEQ	82\$			; IF NO.ERROR SKIP HLT
897	003136	104000			HLT				; CALL ERROR ROUTINE
898	003140						82\$:		
899	003140	122777	000311	175076	CMPB	#311, @SR			; CHECK STATUS
900	003146	001401			BEQ	83\$			; IF NO.ERROR SKIP HLT
901	003150	104000			HLT				; CALL ERROR ROUTINE
902	003152						83\$:		
903									
904									
905	003152	104400			SCOPE				
906	003154	012777	177777	175052	MOV	#-1, @AC			; SET AC=-1
907	003162	012777	000012	175060	MOV	#12, @LSH			; SHIFT 12 TIMES LEFT
908	003170	022777	176000	175036	CMP	#176000, @AC			; CHECK AC
909	003176	001401			BEQ	84\$			; IF NO.ERROR SKIP HLT
910	003200	104000			HLT				; CALL ERROR ROUTINE
911	003202						84\$:		
912	003202	122777	000311	175034	CMPB	#311, @SR			; CHECK STATUS
913	003210	001401			BEQ	85\$			; IF NO.ERROR SKIP HLT
914	003212	104000			HLT				; CALL ERROR ROUTINE
915	003214						85\$:		
916									
917	003214	104400			SCOPE				
918	003216	012777	177777	175010	MOV	#-1, @AC			; SET AC=-1
919	003224	012777	000013	175016	MOV	#13, @LSH			; SHIFT 13 TIMES LEFT
920	003232	022777	174000	174774	CMP	#174000, @AC			; CHECK AC
921	003240	001401			BEQ	86\$			; IF NO.ERROR SKIP HLT
922	003242	104000			HLT				; CALL ERROR ROUTINE
923	003244						86\$:		
924	003244	122777	000311	174772	CMPB	#311, @SR			; CHECK STATUS

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 22

925	003252	001401			BEQ	87\$						; IF NO.ERROR SKIP HLT
926	003254	104000			HLT							; CALL ERROR ROUTINE
927	003256						87\$:					
928	003256	104400			SCOPE							
929	003260	012777	177777	174746	MOV	#-1, @AC						; SET AC=-1
930	003266	012777	000014	174754	MOV	#14, @LSH						; SHIFT 14 TIMES LEFT
931	003274	022777	170000	174732	CMP	#170000, @AC						; CHECK AC
932	003302	001401			BEQ	88\$						; IF NO.ERROR SKIP HLT
933	003304	104000			HLT							; CALL ERROR ROUTINE
934	003306						88\$:					
935	003306	122777	000311	174730	CMPB	#311, @SR						; CHECK STATUS
936	003314	001401			BEQ	89\$						; IF NO.ERROR SKIP HLT
937	003316	104000			HLT							; CALL ERROR ROUTINE
938	003320						89\$:					
939												
940												
941	003320	104400			SCOPE							
942	003322	012777	177777	174704	MOV	#-1, @AC						; SET AC=-1
943	003330	012777	000015	174712	MOV	#15, @LSH						; SHIFT 15 TIMES LEFT
944	003336	022777	160000	174670	CMP	#160000, @AC						; CHECK AC
945	003344	001401			BEQ	90\$						; IF NO.ERROR SKIP HLT
946	003346	104000			HLT							; CALL ERROR ROUTINE
947	003350						90\$:					
948	003350	122777	000311	174666	CMPB	#311, @SR						; CHECK STATUS
949	003356	001401			BEQ	91\$						; IF NO.ERROR SKIP HLT
950	003360	104000			HLT							; CALL ERROR ROUTINE
951	003362						91\$:					
952												
953												
954	003362	104400			SCOPE							
955	003364	012777	177777	174642	MOV	#-1, @AC						; SET AC=-1
956	003372	012777	000016	174650	MOV	#16, @LSH						; SHIFT 16 TIMES LEFT
957	003400	022777	140000	174626	CMP	#140000, @AC						; CHECK AC
958	003406	001401			BEQ	92\$						; IF NO.ERROR SKIP HLT
959	003410	104000			HLT							; CALL ERROR ROUTINE
960	003412						92\$:					
961	003412	122777	000311	174624	CMPB	#311, @SR						; CHECK STATUS
962	003420	001401			BEQ	93\$						; IF NO.ERROR SKIP HLT
963	003422	104000			HLT							; CALL ERROR ROUTINE
964	003424						93\$:					
965												
966												
967	003424	104400			SCOPE							
968	003426	012777	177777	174600	MOV	#-1, @AC						; SET AC=-1
969	003434	012777	000017	174606	MOV	#17, @LSH						; SHIFT 17 TIMES LEFT
970	003442	022777	100000	174564	CMP	#100000, @AC						; CHECK AC
971	003450	001401			BEQ	94\$						; IF NO.ERROR SKIP HLT
972	003452	104000			HLT							; CALL ERROR ROUTINE
973	003454						94\$:					
974	003454	122777	000311	174562	CMPB	#311, @SR						; CHECK STATUS
975	003462	001401			BEQ	95\$						; IF NO.ERROR SKIP HLT
976	003464	104000			HLT							; CALL ERROR ROUTINE
977	003466						95\$:					

```

978 ;*****
979 ; TEST MQ SHIFT RIGHT OF ALTERNATE 1'S AND 0'S
980 ;*****
981
982 003466      104400      MQ10R:
983 003466      012777      125252      174540      SCOPE
984 003470      005077      174532      MOV          #125252, @MQ      ; SET MQ=125252
985 003476      005077      174542      CLR          @AC              ; CLEAR AC
986 003502      027727      174524      125252      CLR          @LSH           ; INITIALIZE SHIFT BY 0
987 003506      001401      BEQ          @MQ, #125252    ; COMPARE MQ
988 003514      104000      HLT                               ; IF NO.ERROR SKIP HLT
989 003516      104000      HLT                               ; CALL ERROR ROUTINE
990 003520
991 003520      122777      000020      174516      64$:      CMPB         #20, @SR
992 003526      001401      BEQ          @SR           ; CHECK STATUS
993 003530      104000      HLT                               ; IF NO.ERROR SKIP HLT
994 003532      HLT                               ; CALL ERROR ROUTINE
995
996
997 003532      104400      SCOPE
998 003534      012777      125252      174474      MOV          #125252, @MQ      ; SET MQ=125252
999 003542      012777      177777      174500      MOV          #177777, @LSH    ; SHIFT 177777 TIMES
1000 003550      105777      174470      TSTB         @SR           ; CHECK STATUS - NO CARRY
1001 003554      001401      BEQ          @SR           ; IF NO.ERROR SKIP HLT
1002 003556      104000      HLT                               ; CALL ERROR ROUTINE
1003 003560
1004 003560      022777      152525      174450      66$:      CMP          #152525, @MQ    ; COMPARE MQ
1005 003566      001401      BEQ          @MQ           ; IF NO.ERROR SKIP HLT
1006 003570      104000      HLT                               ; CALL ERROR ROUTINE
1007 003572
1008
1009 003572      104400      SCOPE
1010 003574      012777      125252      174434      MOV          #125252, @MQ      ; SET MQ=125252
1011 003602      012777      177776      174440      MOV          #177776, @LSH    ; SHIFT 177776 TIMES
1012 003610      122777      000001      174426      CMPB         #1, @SR        ; CHECK STATUS - WITH CARRY
1013 003616      001401      BEQ          @SR           ; IF NO.ERROR SKIP HLT
1014 003620      104000      HLT                               ; CALL ERROR ROUTINE
1015 003622
1016 003622      022777      165252      174406      68$:      CMP          #165252, @MQ    ; COMPARE MQ
1017 003630      001401      BEQ          @MQ           ; IF NO.ERROR SKIP HLT
1018 003632      104000      HLT                               ; CALL ERROR ROUTINE
1019 003634
1020
1021 003634      104400      SCOPE
1022 003636      012777      125252      174372      MOV          #125252, @MQ      ; SET MQ=125252
1023 003644      012777      177775      174376      MOV          #177775, @LSH    ; SHIFT 177775 TIMES
1024 003652      105777      174366      TSTB         @SR           ; CHECK STATUS - NO CARRY
1025 003656      001401      BEQ          @SR           ; IF NO.ERROR SKIP HLT
1026 003660      104000      HLT                               ; CALL ERROR ROUTINE
1027 003662
1028 003662      022777      172525      174346      70$:      CMP          #172525, @MQ    ; COMPARE MQ
1029 003670      001401      BEQ          @MQ           ; IF NO.ERROR SKIP HLT
1030 003672      104000      HLT                               ; CALL ERROR ROUTINE
1031 003674
1032 003674      104400      SCOPE
1033 003676      012777      125252      174332      MOV          #125252, @MQ      ; SET MQ=125252

```

1034	003704	012777	177774	174336	MOV	#177774, @LSH	; SHIFT 177774 TIMES
1035	003712	122777	000001	174324	CMPB	#1, @SR	; CHECK STATUS - WITH CARRY
1036	003720	001401			BEQ	72\$	; IF NO.ERROR SKIP HLT
1037	003722	104000			HLT		; CALL ERROR ROUTINE
1038	003724				72\$:		
1039	003724	022777	175252	174304	CMP	#175252, @MQ	; COMPARE MQ
1040	003732	001401			BEQ	73\$	; IF NO.ERROR SKIP HLT
1041	003734	104000			HLT		; CALL ERROR ROUTINE
1042	003736				73\$:		
1043	003736	104400			SCOPE		
1044	003740	012777	125252	174270	MOV	#125252, @MQ	; SET MQ=125252
1045	003746	012777	177773	174274	MOV	#177773, @LSH	; SHIFT 177773 TIMES
1046	003754	105777	174264		TSTB	@SR	; CHECK STATUS - NO CARRY
1047	003760	001401			BEQ	74\$	; IF NO.ERROR SKIP HLT
1048	003762	104000			HLT		; CALL ERROR ROUTINE
1049	003764				74\$:		
1050	003764	022777	176525	174244	CMP	#176525, @MQ	; COMPARE MQ
1051	003772	001401			BEQ	75\$	; IF NO.ERROR SKIP HLT
1052	003774	104000			HLT		; CALL ERROR ROUTINE
1053	003776				75\$:		
1054					SCOPE		
1055	003776	104400			MOV	#125252, @MQ	; SET MQ=125252
1056	004000	012777	125252	174230	MOV	#177772, @LSH	; SHIFT 177772 TIMES
1057	004006	012777	177772	174234	CMPB	#1, @SR	; CHECK STATUS - WITH CARRY
1058	004014	122777	000001	174222	BEQ	76\$	; IF NO.ERROR SKIP HLT
1059	004022	001401			HLT		; CALL ERROR ROUTINE
1060	004024	104000			76\$:		
1061	004026				CMP	#177252, @MQ	; COMPARE MQ
1062	004026	022777	177252	174202	BEQ	77\$	; IF NO.ERROR SKIP HLT
1063	004034	001401			HLT		; CALL ERROR ROUTINE
1064	004036	104000			77\$:		
1065	004040				SCOPE		
1066					MOV	#125252, @MQ	; SET MQ=125252
1067	004040	104400			MOV	#177771, @LSH	; SHIFT 177771 TIMES
1068	004042	012777	125252	174166	TSTB	@SR	; CHECK STATUS - NO CARRY
1069	004050	012777	177771	174172	BEQ	78\$	; IF NO.ERROR SKIP HLT
1070	004056	105777	174162		HLT		; CALL ERROR ROUTINE
1071	004062	001401			78\$:		
1072	004064	104000			CMP	#177525, @MQ	; COMPARE MQ
1073	004066						
1074	004066	022777	177525	174142			



L02

1075	004074	001401			BEQ	79\$		:IF NO.ERROR SKIP HLT
1076	004076	104000			HLT			:CALL ERROR ROUTINE
1077	004100					79\$:		
1078	004100	104400			SCOPE			
1079	004102	012777	125252	174126	MOV	#125252, @MQ		:SET MQ=125252
1080	004110	012777	177770	174132	MOV	#177770, @LSH		:SHIFT 177770 TIMES
1081	004116	122777	000001	174120	CMPB	#1, @SR		:CHECK STATUS - WITH CARRY
1082	004124	001401			BEQ	80\$		:IF NO.ERROR SKIP HLT
1083	004126	104000			HLT			:CALL ERROR ROUTINE
1084	004130					80\$:		
1085	004130	022777	177652	174100	CMP	#177652, @MQ		:COMPARE MQ
1086	004136	001401			BEQ	81\$		:IF NO.ERROR SKIP HLT
1087	004140	104000			HLT			:CALL ERROR ROUTINE
1088	004142					81\$:		
1089								
1090	004142	104400			SCOPE			
1091	004144	012777	125252	174064	MOV	#125252, @MQ		:SET MQ=125252
1092	004152	012777	177767	174070	MOV	#177767, @LSH		:SHIFT 177767 TIMES
1093	004160	105777	174060		TSTB	@SR		:CHECK STATUS - NO CARRY
1094	004164	001401			BEQ	82\$		:IF NO.ERROR SKIP HLT
1095	004166	104000			HLT			:CALL ERROR ROUTINE
1096	004170					82\$:		
1097	004170	022777	177725	174040	CMP	#177725, @MQ		:COMPARE MQ
1098	004176	001401			BEQ	83\$		:IF NO.ERROR SKIP HLT
1099	004200	104000			HLT			:CALL ERROR ROUTINE
1100	004202					83\$:		
1101								
1102	004202	104400			SCOPE			
1103	004204	012777	125252	174024	MOV	#125252, @MQ		:SET MQ=125252
1104	004212	012777	177766	174030	MOV	#177766, @LSH		:SHIFT 177766 TIMES
1105	004220	122777	000001	174016	CMPB	#1, @SR		:CHECK STATUS - WITH CARRY
1106	004226	001401			BEQ	84\$		:IF NO.ERROR SKIP HLT
1107	004230	104000			HLT			:CALL ERROR ROUTINE
1108	004232					84\$:		
1109	004232	022777	177752	173776	CMP	#177752, @MQ		:COMPARE MQ
1110	004240	001401			BEQ	85\$		:IF NO.ERROR SKIP HLT
1111	004242	104000			HLT			:CALL ERROR ROUTINE
1112	004244					85\$:		
1113								
1114	004244	104400			SCOPE			
1115	004246	012777	125252	173762	MOV	#125252, @MQ		:SET MQ=125252
1116	004254	012777	177765	173766	MOV	#177765, @LSH		:SHIFT 177765 TIMES
1117	004262	105777	173756		TSTB	@SR		:CHECK STATUS - NO CARRY
1118	004266	001401			BEQ	86\$		:IF NO.ERROR SKIP HLT
1119	004270	104000			HLT			:CALL ERROR ROUTINE
1120	004272					86\$:		
1121	004272	022777	177765	173736	CMP	#177765, @MQ		:COMPARE MQ

1122	004300	001401			BEQ	87\$			; IF NO.ERROR SKIP HLT
1123	004302	104000			HLT				; CALL ERROR ROUTINE
1124	004304					87\$:			
1125	004304	104400			SCOPE				
1126	004306	012777	125252	173722	MOV	#125252, @MQ			; SET MQ=125252
1127	004314	012777	177764	173726	MOV	#177764, @LSH			; SHIFT 177764 TIMES
1128	004322	122777	000001	173714	CMPB	#1, @SR			; CHECK STATUS - WITH CARRY
1129	004330	001401			BEQ	88\$			; IF NO.ERROR SKIP HLT
1130	004332	104000			HLT				; CALL ERROR ROUTINE
1131	004334					88\$:			
1132	004334	022777	177772	173674	CMP	#177772, @MQ			; COMPARE MQ
1133	004342	001401			BEQ	89\$			; IF NO.ERROR SKIP HLT
1134	004344	104000			HLT				; CALL ERROR ROUTINE
1135	004346					89\$:			
1136									
1137	004346	104400			SCOPE				
1138	004350	012777	125252	173660	MOV	#125252, @MQ			; SET MQ=125252
1139	004356	012777	177763	173664	MOV	#177763, @LSH			; SHIFT 177763 TIMES
1140	004364	105777	173654		TSTB	@SR			; CHECK STATUS - NO CARRY
1141	004370	001401			BEQ	90\$			; IF NO.ERROR SKIP HLT
1142	004372	104000			HLT				; CALL ERROR ROUTINE
1143	004374					90\$:			
1144	004374	022777	177775	173634	CMP	#177775, @MQ			; COMPARE MQ
1145	004402	001401			BEQ	91\$			; IF NO.ERROR SKIP HLT
1146	004404	104000			HLT				; CALL ERROR ROUTINE
1147	004406					91\$:			
1148									
1149	004406	104400			SCOPE				
1150	004410	012777	125252	173620	MOV	#125252, @MQ			; SET MQ=125252
1151	004416	012777	177762	173624	MOV	#177762, @LSH			; SHIFT 177762 TIMES
1152	004424	122777	000001	173612	CMPB	#1, @SR			; CHECK STATUS - WITH CARRY
1153	004432	001401			BEQ	92\$			; IF NO.ERROR SKIP HLT
1154	004434	104000			HLT				; CALL ERROR ROUTINE
1155	004436					92\$:			
1156	004436	022777	177776	173572	CMP	#177776, @MQ			; COMPARE MQ
1157	004444	001401			BEQ	93\$			; IF NO.ERROR SKIP HLT
1158	004446	104000			HLT				; CALL ERROR ROUTINE
1159	004450					93\$:			
1160									
1161	004450	104400			SCOPE				
1162	004452	012777	125252	173556	MOV	#125252, @MQ			; SET MQ=125252
1163	004460	012777	177761	173562	MOV	#177761, @LSH			; SHIFT 177761 TIMES
1164	004466	105777	173552		TSTB	@SR			; CHECK STATUS - NO CARRY
1165	004472	001401			BEQ	94\$			; IF NO.ERROR SKIP HLT
1166	004474	104000			HLT				; CALL ERROR ROUTINE
1167	004476					94\$:			
1168	004476	022777	177777	173532	CMP	#177777, @MQ			; COMPARE MQ
1169	004504	001401			BEQ	95\$			; IF NO.ERROR SKIP HLT
1170	004506	104000			HLT				; CALL ERROR ROUTINE
1171	004510					95\$:			

```

1172 ;*****
1173 ; TEST AC SHIFT LEFT OF ALTERNATE 1'S AND 0'S
1174 ;*****
1175 AC10L:
1176 004510 104400 SCOPE
1177 004512 005077 173520 CLR 0M0
1178 004516 012777 125252 173510 MOV #125252,0AC ;SET AC=125252
1179 004524 005077 173520 CLR 0LSH ;INITALIZE SHIFT BY 0
1180 004530 027727 173500 125252 CMP 0AC,#125252 ;COMPARE AC
1181 004536 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT
1182 004540 104000 HLT ;CALL ERROR ROUTINE
1183 004542
1184 004542 122777 000310 173474 64$: CMPB #310,0SR ;CHECK STATUS
1185 004550 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT
1186 004552 104000 HLT ;CALL ERROR ROUTINE
1187 004554 65$:
1188
1189 004554 104400 SCOPE
1190 004556 012777 125252 173450 MOV #125252,0AC ;SET AC=125252
1191 004564 012777 000001 173456 MOV #1,0LSH ;SHIFT 1 TIMES
1192 004572 122777 000211 173444 CMPB #211,0SR ;CHECK STATUS - WITH CARRY
1193 004600 001401 BEQ 66$ ;IF NO.ERROR SKIP HLT
1194 004602 104000 HLT ;CALL ERROR ROUTINE
1195 004604 66$:
1196 004604 022777 052524 173422 CMP #52524,0AC ;COMPARE AC
1197 004612 001401 BEQ 67$ ;IF NO.ERROR SKIP HLT
1198 004614 104000 HLT ;CALL ERROR ROUTINE
1199 004616 67$:
1200
1201
1202 004616 104400 SCOPE
1203 004620 012777 125252 173406 MOV #125252,0AC ;SET AC=125252
1204 004626 012777 000002 173414 MOV #2,0LSH ;SHIFT 2 TIMES
1205 004634 122777 000110 173402 CMPB #110,0SR ;CHECK STATUS - NO CARRY
1206 004642 001401 BEQ 68$ ;IF NO.ERROR SKIP HLT
1207 004644 104000 HLT ;CALL ERROR ROUTINE
1208 004646 68$:
1209 004646 022777 125250 173360 CMP #125250,0AC ;COMPARE AC
1210 004654 001401 BEQ 69$ ;IF NO.ERROR SKIP HLT
1211 004656 104000 HLT ;CALL ERROR ROUTINE
1212 004660 69$:
1213
1214
1215 004660 104400 SCOPE
1216 004662 012777 125252 173344 MOV #125252,0AC ;SET AC=125252
1217 004670 012777 000003 173352 MOV #3,0LSH ;SHIFT 3 TIMES
1218 004676 122777 000211 173340 CMPB #211,0SR ;CHECK STATUS - WITH CARRY
1219 004704 001401 BEQ 70$ ;IF NO.ERROR SKIP HLT
1220 004706 104000 HLT ;CALL ERROR ROUTINE
1221 004710 70$:
1222 004710 022777 052520 173316 CMP #52520,0AC ;COMPARE AC

```

1223	004716	001401			BEQ	71\$			; IF NO.ERROR SKIP HLT
1224	004720	104000			HLT				; CALL ERROR ROUTINE
1225	004722					71\$:			
1226	004722	104400			SCOPE				
1227	004724	012777	125252	173302	MOV	#125252, @AC			; SET AC=125252
1228	004732	012777	000004	173310	MOV	#4, @LSH			; SHIFT 4 TIMES
1229	004740	122777	000110	173276	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
1230	004746	001401			BEQ	72\$			; IF NO.ERROR SKIP HLT
1231	004750	104000			HLT				; CALL ERROR ROUTINE
1232	004752					72\$:			
1233	004752	022777	125240	173254	CMP	#125240, @AC			; COMPARE AC
1234	004760	001401			BEQ	73\$			; IF NO.ERROR SKIP HLT
1235	004762	104000			HLT				; CALL ERROR ROUTINE
1236	004764					73\$:			
1237									
1238									
1239	004764	104400			SCOPE				
1240	004766	012777	125252	173240	MOV	#125252, @AC			; SET AC=125252
1241	004774	012777	000005	173246	MOV	#5, @LSH			; SHIFT 5 TIMES
1242	005002	122777	000211	173234	CMPB	#211, @SR			; CHECK STATUS - WITH CARRY
1243	005010	001401			BEQ	74\$			; IF NO.ERROR SKIP HLT
1244	005012	104000			HLT				; CALL ERROR ROUTINE
1245	005014					74\$:			
1246	005014	022777	052500	173212	CMP	#52500, @AC			; COMPARE AC
1247	005022	001401			BEQ	75\$			; IF NO.ERROR SKIP HLT
1248	005024	104000			HLT				; CALL ERROR ROUTINE
1249	005026					75\$:			
1250									
1251									
1252	005026	104400			SCOPE				
1253	005030	012777	125252	173176	MOV	#125252, @AC			; SET AC=125252
1254	005036	012777	000006	173204	MOV	#6, @LSH			; SHIFT 6 TIMES
1255	005044	122777	000110	173172	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
1256	005052	001401			BEQ	76\$			; IF NO.ERROR SKIP HLT
1257	005054	104000			HLT				; CALL ERROR ROUTINE
1258	005056					76\$:			
1259	005056	022777	125200	173150	CMP	#125200, @AC			; COMPARE AC

```

1260 005064 001401      BEQ      77$      ; IF NO.ERROR SKIP HLT
1261 005066 104000      HLT                                     ; CALL ERROR ROUTINE
1262          005070      77$:
1263
1264
1265 005070 104400      SCOPE
1266 005072 012777 125252 173134  MOV      #125252, @AC      ; SET AC=125252
1267 005100 012777 000007 173142  MOV      #7, @LSH        ; SHIFT 7 TIMES
1268 005106 122777 000211 173130  CMPB    #211, @SR        ; CHECK STATUS - WITH CARRY
1269 005114 001401      BEQ      78$      ; IF NO.ERROR SKIP HLT
1270 005116 104000      HLT                                     ; CALL ERROR ROUTINE
1271          005120      78$:
1272 005120 022777 052400 173106  CMP     #52400, @AC      ; COMPARE AC
1273 005126 001401      BEQ      79$      ; IF NO.ERROR SKIP HLT
1274 005130 104000      HLT                                     ; CALL ERROR ROUTINE
1275          005132      79$:
1276
1277 005132 104400      SCOPE
1278 005134 012777 125252 173072  MOV      #125252, @AC      ; SET AC=125252
1279 005142 012777 000010 173100  MOV      #10, @LSH       ; SHIFT 10 TIMES
1280 005150 122777 000110 173066  CMPB    #110, @SR        ; CHECK STATUS - NO CARRY
1281 005156 001401      BEQ      80$      ; IF NO.ERROR SKIP HLT
1282 005160 104000      HLT                                     ; CALL ERROR ROUTINE
1283          005162      80$:
1284 005162 022777 125000 173044  CMP     #125000, @AC     ; COMPARE AC
1285 005170 001401      BEQ      81$      ; IF NO.ERROR SKIP HLT
1286 005172 104000      HLT                                     ; CALL ERROR ROUTINE
1287          005174      81$:
1288
1289 005174 104400      SCOPE
1290 005176 012777 125252 173030  MOV      #125252, @AC      ; SET AC=125252
1291 005204 012777 000011 173036  MOV      #11, @LSH       ; SHIFT 11 TIMES
1292 005212 122777 000211 173024  CMPB    #211, @SR        ; CHECK STATUS - WITH CARRY
1293 005220 001401      BEQ      82$      ; IF NO.ERROR SKIP HLT
1294 005222 104000      HLT                                     ; CALL ERROR ROUTINE
1295          005224      82$:
1296 005224 022777 052000 173002  CMP     #52000, @AC     ; COMPARE AC
1297 005232 001401      BEQ      83$      ; IF NO.ERROR SKIP HLT
1298 005234 104000      HLT                                     ; CALL ERROR ROUTINE
1299          005236      83$:
1300
1301
1302 005236 104400      SCOPE
1303 005240 012777 125252 172766  MOV      #125252, @AC      ; SET AC=125252
1304 005246 012777 000012 172774  MOV      #12, @LSH       ; SHIFT 12 TIMES
1305 005254 122777 000110 172762  CMPB    #110, @SR        ; CHECK STATUS - NO CARRY
1306 005262 001401      BEQ      84$      ; IF NO.ERROR SKIP HLT
1307 005264 104000      HLT                                     ; CALL ERROR ROUTINE
1308          005266      84$:
1309 005266 022777 124000 172740  CMP     #124000, @AC     ; COMPARE AC

```

1310	005274	001401			BEQ	85\$				: IF NO.ERROR SKIP HLT
1311	005276	104000			HLT					: CALL ERROR ROUTINE
1312	005300				85\$:					
1313										
1314										
1315	005300	104400			SCOPE					
1316	005302	012777	125252	172724	MOV	#125252, JAC				: SET AC=125252
1317	005310	012777	000013	172732	MOV	#13, JLSH				: SHIFT 13 TIMES
1318	005316	122777	000211	172720	CMPB	#211, JSR				: CHECK STATUS - WITH CARRY
1319	005324	001401			BEQ	86\$				: IF NO.ERROR SKIP HLT
1320	005326	104000			HLT					: CALL ERROR ROUTINE
1321	005330				86\$:					
1322	005330	022777	050000	172676	CMP	#50000, JAC				: COMPARE AC
1323	005336	001401			BEQ	87\$				: IF NO.ERROR SKIP HLT
1324	005340	104000			HLT					: CALL ERROR ROUTINE
1325	005342				87\$:					
1326										
1327	005342	104400			SCOPE					
1328	005344	012777	125252	172662	MOV	#125252, JAC				: SET AC=125252
1329	005352	012777	000014	172670	MOV	#14, JLSH				: SHIFT 14 TIMES
1330	005360	122777	000110	172656	CMPB	#110, JSR				: CHECK STATUS - NO CARRY
1331	005366	001401			BEQ	88\$				: IF NO.ERROR SKIP HLT
1332	005370	104000			HLT					: CALL ERROR ROUTINE
1333	005372				88\$:					
1334	005372	022777	120000	172634	CMP	#120000, JAC				: COMPARE AC
1335	005400	001401			BEQ	89\$				: IF NO.ERROR SKIP HLT
1336	005402	104000			HLT					: CALL ERROR ROUTINE
1337	005404				89\$:					
1338										
1339										
1340	005404	104400			SCOPE					
1341	005406	012777	125252	172620	MOV	#125252, JAC				: SET AC=125252
1342	005414	012777	000015	172626	MOV	#15, JLSH				: SHIFT 15 TIMES
1343	005422	122777	000211	172614	CMPB	#211, JSR				: CHECK STATUS - WITH CARRY
1344	005430	001401			BEQ	90\$				: IF NO.ERROR SKIP HLT
1345	005432	104000			HLT					: CALL ERROR ROUTINE
1346	005434				90\$:					
1347	005434	022777	040000	172572	CMP	#40000, JAC				: COMPARE AC
1348	005442	001401			BEQ	91\$				: IF NO.ERROR SKIP HLT
1349	005444	104000			HLT					: CALL ERROR ROUTINE
1350	005446				91\$:					
1351										
1352	005446	104400			SCOPE					
1353	005450	012777	125252	172556	MOV	#125252, JAC				: SET AC=125252
1354	005456	012777	000016	172564	MOV	#16, JLSH				: SHIFT 16 TIMES
1355	005464	122777	000110	172552	CMPB	#110, JSR				: CHECK STATUS - NO CARRY
1356	005472	001401			BEQ	92\$				: IF NO.ERROR SKIP HLT
1357	005474	104000			HLT					: CALL ERROR ROUTINE
1358	005476				92\$:					
1359	005476	022777	100000	172530	CMP	#100000, JAC				: COMPARE AC
1360	005504	001401			BEQ	93\$				: IF NO.ERROR SKIP HLT
1361	005506	104000			HLT					: CALL ERROR ROUTINE
1362	005510				93\$:					



1419	005736				72\$:					
1420	005736	104400				SCOPE				; TEST OF LOGICAL SHIFT
1421	005740	012777	125252	172270		MOV	#125252, MQ			; LOAD MQ WITH 125252
1422	005746	012777	000000	172260		MOV	#0, AC			; LOAD AC WITH 0
1423	005754	012777	000020	172266		MOV	#16, LSH			; LOAD SHIFT COUNT (LSH) WITH 16.
1424	005762	022777	125252	172244		CMP	#125252, AC			; COMPARE AC WITH 125252
1425	005770	001401				BEQ	73\$			; IF NO.ERROR SKIP HLT
1426	005772	104000				HLT				; CALL ERROR ROUTINE
1427	005774				73\$:					
1428	005774	022777	000000	172234		CMP	#0, MQ			; COMPARE MQ WITH 0
1429	006002	001401				BEQ	74\$			; IF NO.ERROR SKIP HLT
1430	006004	104000				HLT				; CALL ERROR ROUTINE
1431	006006				74\$:					
1432	006006	122777	000110	172230		CMPB	#110, SR			; COMPARE SR WITH 110
1433	006014	001401				BEQ	75\$			; IF NO.ERROR SKIP HLT
1434	006016	104000				HLT				; CALL ERROR ROUTINE
1435	006020				75\$:					
1436										
1437										
1438										
1439	006020	104400				SCOPE				; TEST OF LOGICAL SHIFT
1440	006022	012777	125252	172206		MOV	#125252, MQ			; LOAD MQ WITH 125252
1441	006030	012777	000000	172176		MOV	#0, AC			; LOAD AC WITH 0
1442	006036	012777	000001	172204		MOV	#1, LSH			; LOAD SHIFT COUNT (LSH) WITH 1
1443	006044	022777	000001	172162		CMP	#1, AC			; COMPARE AC WITH 1
1444	006052	001401				BEQ	76\$			; IF NO.ERROR SKIP HLT
1445	006054	104000				HLT				; CALL ERROR ROUTINE
1446	006056				76\$:					
1447	006056	022777	052524	172152		CMP	#52524, MQ			; COMPARE MQ WITH 52524
1448	006064	001401				BEQ	77\$			; IF NO.ERROR SKIP HLT
1449	006066	104000				HLT				; CALL ERROR ROUTINE
1450	006070				77\$:					
1451	006070	122777	000000	172146		CMPB	#0, SR			; COMPARE SR WITH 0
1452	006076	001401				BEQ	78\$			; IF NO.ERROR SKIP HLT
1453	006100	104000				HLT				; CALL ERROR ROUTINE
1454	006102				78\$:					
1455										
1456										
1457	006102	104400				SCOPE				; TEST OF LOGICAL SHIFT
1458	006104	012777	000001	172124		MOV	#1, MQ			; LOAD MQ WITH 1
1459	006112	012777	000000	172114		MOV	#0, AC			; LOAD AC WITH 0
1460	006120	012777	000021	172122		MOV	#21, LSH			; LOAD SHIFT COUNT (LSH) WITH 21
1461	006126	022777	000002	172100		CMP	#2, AC			; COMPARE AC WITH 2
1462	006134	001401				BEQ	79\$			; IF NO.ERROR SKIP HLT
1463	006136	104000				HLT				; CALL ERROR ROUTINE
1464	006140				79\$:					
1465	006140	022777	000000	172070		CMP	#0, MQ			; COMPARE MQ WITH 0
1466	006146	001401				BEQ	80\$			; IF NO.ERROR SKIP HLT
1467	006150	104000				HLT				; CALL ERROR ROUTINE
1468	006152				80\$:					
1469	006152	122777	000010	172064		CMPB	#10, SR			; COMPARE SR WITH 10
1470	006160	001401				BEQ	81\$			; IF NO.ERROR SKIP HLT
1471	006162	104000				HLT				; CALL ERROR ROUTINE
1472	006164				81\$:					
1473										
1474	006164	104400				SCOPE				; TEST OF LOGICAL SHIFT



1475	006166	012777	000001	172042	MOV	#1, MQ	:LOAD MQ WITH 1
1476	006174	012777	000000	172032	MOV	#0, AC	:LOAD AC WITH 0
1477	006202	012777	000022	172040	MOV	#22, LSH	:LOAD SHIFT COUNT (LSH) WITH 22
1478	006210	022777	000004	172016	CMP	#4, AC	:COMPARE AC WITH 4
1479	006216	001401			BEQ	82\$	:IF NO.ERROR SKIP HLT
1480	006220	104000			HLT		:CALL ERROR ROUTINE
1481	006222					82\$:	
1482	006222	022777	000000	172006	CMP	#0, MQ	:COMPARE MQ WITH 0
1483	006230	001401			BEQ	83\$	:IF NO.ERROR SKIP HLT
1484	006232	104000			HLT		:CALL ERROR ROUTINE
1485	006234					83\$:	
1486	006234	122777	000010	172002	CMPB	#10, SR	:COMPARE SR WITH 10
1487	006242	001401			BEQ	84\$	:IF NO.ERROR SKIP HLT
1488	006244	104000			HLT		:CALL ERROR ROUTINE
1489	006246					84\$:	
1490							
1491							
1492	006246	104400			SCOPE		:TEST OF LOGICAL SHIFT
1493	006250	012777	000001	171760	MOV	#1, MQ	:LOAD MQ WITH 1
1494	006256	012777	000000	171750	MOV	#0, AC	:LOAD AC WITH 0
1495	006264	012777	000023	171756	MOV	#23, LSH	:LOAD SHIFT COUNT (LSH) WITH 23
1496	006272	022777	000010	171734	CMP	#10, AC	:COMPARE AC WITH 10
1497	006300	001401			BEQ	85\$	:IF NO.ERROR SKIP HLT
1498	006302	104000			HLT		:CALL ERROR ROUTINE
1499	006304					85\$:	
1500	006304	022777	000000	171724	CMP	#0, MQ	:COMPARE MQ WITH 0
1501	006312	001401			BEQ	86\$	:IF NO.ERROR SKIP HLT
1502	006314	104000			HLT		:CALL ERROR ROUTINE
1503	006316					86\$:	
1504	006316	122777	000010	171720	CMPB	#10, SR	:COMPARE SR WITH 10
1505	006324	001401			BEQ	87\$	:IF NO.ERROR SKIP HLT
1506	006326	104000			HLT		:CALL ERROR ROUTINE
1507	006330					87\$:	
1508							
1509							
1510	006330	104400			SCOPE		:TEST OF LOGICAL SHIFT
1511	006332	012777	000001	171676	MOV	#1, MQ	:LOAD MQ WITH 1
1512	006340	012777	000000	171666	MOV	#0, AC	:LOAD AC WITH 0
1513	006346	012777	000024	171674	MOV	#24, LSH	:LOAD SHIFT COUNT (LSH) WITH 24
1514	006354	022777	000020	171652	CMP	#20, AC	:COMPARE AC WITH 20
1515	006362	001401			BEQ	88\$	:IF NO.ERROR SKIP HLT
1516	006364	104000			HLT		:CALL ERROR ROUTINE
1517	006366					88\$:	
1518	006366	022777	000000	171642	CMP	#0, MQ	:COMPARE MQ WITH 0
1519	006374	001401			BEQ	89\$	:IF NO.ERROR SKIP HLT
1520	006376	104000			HLT		:CALL ERROR ROUTINE
1521	006400					89\$:	
1522	006400	122777	000010	171636	CMPB	#10, SR	:COMPARE SR WITH 10
1523	006406	001401			BEQ	90\$	:IF NO.ERROR SKIP HLT
1524	006410	104000			HLT		:CALL ERROR ROUTINE
1525	006412					90\$:	
1526							
1527	006412	104400			SCOPE		:TEST OF LOGICAL SHIFT
1528	006414	012777	000001	171614	MOV	#1, MQ	:LOAD MQ WITH 1
1529	006422	012777	000000	171604	MOV	#0, AC	:LOAD AC WITH 0
1530	006430	012777	000025	171612	MOV	#25, LSH	:LOAD SHIFT COUNT (LSH) WITH 25

```

1531 006436 022777 000040 171570      CMP      #40,AC      ;COMPARE AC WITH 40
1532 006444 001401                    BEQ      91$       ;IF NO.ERROR SKIP HLT
1533 006446 104000                    HLT                               ;CALL ERROR ROUTINE
1534 006450                                91$:
1535 006450 022777 000000 171560      CMP      #0,AC      ;COMPARE MQ WITH 0
1536 006456 001401                    BEQ      92$       ;IF NO.ERROR SKIP HLT
1537 006460 104000                    HLT                               ;CALL ERROR ROUTINE
1538 006462                                92$:
1539 006462 104000                    HLT
1540 006464 122777 000010 171552      CMPB     #10,SR     ;COMPARE SR WITH 10
1541 006472 001401                    BEQ      93$       ;IF NO.ERROR SKIP HLT
1542 006474 104000                    HLT                               ;CALL ERROR ROUTINE
1543 006476                                93$:
1544
1545
1546
1547 006476 104400                    SCOPE
1548 006500 012777 000001 171530      MOV      #1,AC      ;TEST OF LOGICAL SHIFT
1549 006506 012777 000000 171520      MOV      #0,AC      ;LOAD MQ WITH 1
1550 006514 012777 000026 171526      MOV      #26,LSH    ;LOAD AC WITH 0
1551 006522 022777 000100 171504      CMP      #100,AC    ;LOAD SHIFT COUNT (LSH) WITH 26
1552 006530 001401                    BEQ      94$       ;COMPARE AC WITH 100
1553 006532 104000                    HLT                               ;IF NO.ERROR SKIP HLT
1554 006534                                94$:
1555 006534 022777 000000 171474      CMP      #0,AC      ;COMPARE MQ WITH 0
1556 006542 001401                    BEQ      95$       ;IF NO.ERROR SKIP HLT
1557 006544 104000                    HLT                               ;CALL ERROR ROUTINE
1558 006546                                95$:
1559 006546 122777 000010 171470      CMPB     #10,SR     ;COMPARE SR WITH 10
1560 006554 001401                    BEQ      96$       ;IF NO.ERROR SKIP HLT
1561 006556 104000                    HLT                               ;CALL ERROR ROUTINE
1562 006560                                96$:
1563
1564
1565 006560 104400                    SCOPE
1566 006562 012777 000001 171446      MOV      #1,AC      ;TEST OF LOGICAL SHIFT
1567 006570 012777 000000 171436      MOV      #0,AC      ;LOAD MQ WITH 1
1568 006576 012777 000027 171444      MOV      #27,LSH    ;LOAD AC WITH 0
1569 006604 022777 000200 171422      CMP      #200,AC    ;LOAD SHIFT COUNT (LSH) WITH 27
1570 006612 001401                    BEQ      97$       ;COMPARE AC WITH 200
1571 006614 104000                    HLT                               ;IF NO.ERROR SKIP HLT
1572 006616                                97$:
1573 006616 022777 000000 171412      CMP      #0,AC      ;COMPARE MQ WITH 0
1574 006624 001401                    BEQ      98$       ;IF NO.ERROR SKIP HLT
1575 006626 104000                    HLT                               ;CALL ERROR ROUTINE
1576 006630                                98$:
1577 006630 122777 000010 171406      CMPB     #10,SR     ;COMPARE SR WITH 10
1578 006636 001401                    BEQ      99$       ;IF NO.ERROR SKIP HLT
1579 006640 104000                    HLT                               ;CALL ERROR ROUTINE
1580 006642                                99$:
1581 006642 104400                    SCOPE
1582 006644 012777 000001 171364      MOV      #1,AC      ;TEST OF LOGICAL SHIFT
1583 006652 012777 000000 171354      MOV      #0,AC      ;LOAD MQ WITH 1
1584 006660 012777 000030 171362      MOV      #30,LSH    ;LOAD AC WITH 0
1585 006666 022777 000400 171340      CMP      #400,AC    ;LOAD SHIFT COUNT (LSH) WITH 30
1586 006674 001401                    BEQ      100$      ;COMPARE AC WITH 400
;IF NO.ERROR SKIP HLT
    
```

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 35

1587	006676	104000			HLT				;CALL ERROR ROUTINE
1588	006700								
1589	006700	022777	000000	171330	100\$:	CMP	#0, MQ		;COMPARE MQ WITH 0
1590	006706	001401				BEQ	101\$		;IF NO.ERROR SKIP HLT
1591	006710	104000				HLT			;CALL ERROR ROUTINE
1592	006712				101\$:				
1593	006712	122777	000010	171324		CMPB	#10, SR		;COMPARE SR WITH 10
1594	006720	001401				BEQ	102\$		;IF NO.ERROR SKIP HLT
1595	006722	104000				HLT			;CALL ERROR ROUTINE
1596	006724				102\$:				
1597	006724	104400				SCOPE			;TEST OF LOGICAL SHIFT
1598	006726	012777	000001	171302		MOV	#1, MQ		;LOAD MQ WITH 1
1599	006734	012777	000000	171272		MOV	#0, AC		;LOAD AC WITH 0
1600	006742	012777	000031	171300		MOV	#31, LSH		;LOAD SHIFT COUNT (LSH) WITH 31
1601	006750	022777	001000	171256		CMP	#1000, AC		;COMPARE AC WITH 1000
1602	006756	001401				BEQ	103\$		;IF NO.ERROR SKIP HLT
1603	006760	104000				HLT			;CALL ERROR ROUTINE
1604	006762				103\$:				
1605	006762	022777	000000	171246		CMP	#0, MQ		;COMPARE MQ WITH 0
1606	006770	001401				BEQ	104\$		;IF NO.ERROR SKIP HLT
1607	006772	104000				HLT			;CALL ERROR ROUTINE
1608	006774				104\$:				
1609	006774	122777	000010	171242		CMPB	#10, SR		;COMPARE SR WITH 10
1610	007002	001401				BEQ	105\$		;IF NO.ERROR SKIP HLT
1611	007004	104000				HLT			;CALL ERROR ROUTINE
1612	007006				105\$:				
1613	007006	104400				SCOPE			;TEST OF LOGICAL SHIFT
1614	007010	012777	000001	171220		MOV	#1, MQ		;LOAD MQ WITH 1
1615	007016	012777	000000	171210		MOV	#0, AC		;LOAD AC WITH 0
1616	007024	012777	000032	171216		MOV	#32, LSH		;LOAD SHIFT COUNT (LSH) WITH 32
1617	007032	022777	002000	171174		CMP	#2000, AC		;COMPARE AC WITH 2000
1618	007040	001401				BEQ	106\$		;IF NO.ERROR SKIP HLT
1619	007042	104000				HLT			;CALL ERROR ROUTINE
1620	007044				106\$:				
1621	007044	022777	000000	171164		CMP	#0, MQ		;COMPARE MQ WITH 0
1622	007052	001401				BEQ	107\$		;IF NO.ERROR SKIP HLT
1623	007054	104000				HLT			;CALL ERROR ROUTINE
1624	007056				107\$:				
1625	007056	122777	000010	171160		CMPB	#10, SR		;COMPARE SR WITH 10
1626	007064	001401				BEQ	108\$		;IF NO.ERROR SKIP HLT
1627	007066	104000				HLT			;CALL ERROR ROUTINE
1628	007070				108\$:				
1629	007070	104400				SCOPE			;TEST OF LOGICAL SHIFT
1630	007072	012777	000001	171136		MOV	#1, MQ		;LOAD MQ WITH 1
1631	007100	012777	000000	171126		MOV	#0, AC		;LOAD AC WITH 0
1632	007106	012777	000033	171134		MOV	#33, LSH		;LOAD SHIFT COUNT (LSH) WITH 33
1633	007114	022777	004000	171112		CMP	#4000, AC		;COMPARE AC WITH 4000
1634	007122	001401				BEQ	109\$		;IF NO.ERROR SKIP HLT
1635	007124	104000				HLT			;CALL ERROR ROUTINE
1636	007126				109\$:				
1637	007126	022777	000000	171102		CMP	#0, MQ		;COMPARE MQ WITH 0
1638	007134	001401				BEQ	110\$		;IF NO.ERROR SKIP HLT
1639	007136	104000				HLT			;CALL ERROR ROUTINE
1640	007140				110\$:				
1641	007140	122777	000010	171076		CMPB	#10, SR		;COMPARE SR WITH 10
1642	007146	001401				BEQ	111\$		;IF NO.ERROR SKIP HLT

1643	007150	104000			HLT				;CALL ERROR ROUTINE
1644	007152							111\$:	
1645									
1646									
1647	007152	104400			SCOPE				;TEST OF LOGICAL SHIFT
1648	007154	012777	000001	171054	MOV	#1,AMQ			;LOAD MQ WITH 1
1649	007162	012777	000000	171044	MOV	#0,AC			;LOAD AC WITH 0
1650	007170	012777	000034	171052	MOV	#34,LSH			;LOAD SHIFT COUNT (LSH) WITH 34
1651	007176	022777	010000	171030	CMP	#10000,AC			;COMPARE AC WITH 10000
1652	007204	001401			BEQ	112\$			;IF NO.ERROR SKIP HLT
1653	007206	104000			HLT				;CALL ERROR ROUTINE
1654	007210							112\$:	
1655	007210	022777	000000	171020	CMP	#0,AMQ			;COMPARE MQ WITH 0
1656	007216	001401			BEQ	113\$			;IF NO.ERROR SKIP HLT
1657	007220	104000			HLT				;CALL ERROR ROUTINE
1658	007222							113\$:	
1659	007222	122777	000010	171014	CMPB	#10,SR			;COMPARE SR WITH 10
1660	007230	001401			BEQ	114\$			;IF NO.ERROR SKIP HLT
1661	007232	104000			HLT				;CALL ERROR ROUTINE
1662	007234							114\$:	
1663									
1664									
1665	007234	104400			SCOPE				;TEST OF LOGICAL SHIFT
1666	007236	012777	000001	170772	MOV	#1,AMQ			;LOAD MQ WITH 1
1667	007244	012777	000000	170762	MOV	#0,AC			;LOAD AC WITH 0
1668	007252	012777	000035	170770	MOV	#35,LSH			;LOAD SHIFT COUNT (LSH) WITH 35
1669	007260	022777	020000	170746	CMP	#20000,AC			;COMPARE AC WITH 20000
1670	007266	001401			BEQ	115\$			;IF NO.ERROR SKIP HLT
1671	007270	104000			HLT				;CALL ERROR ROUTINE
1672	007272							115\$:	
1673	007272	022777	000000	170736	CMP	#0,AMQ			;COMPARE MQ WITH 0
1674	007300	001401			BEQ	116\$			;IF NO.ERROR SKIP HLT
1675	007302	104000			HLT				;CALL ERROR ROUTINE
1676	007304							116\$:	
1677	007304	122777	000010	170732	CMPB	#10,SR			;COMPARE SR WITH 10
1678	007312	001401			BEQ	117\$			;IF NO.ERROR SKIP HLT
1679	007314	104000			HLT				;CALL ERROR ROUTINE
1680	007316							117\$:	
1681	007316	104400			SCOPE				;TEST OF LOGICAL SHIFT
1682	007320	012777	000001	170710	MOV	#1,AMQ			;LOAD MQ WITH 1
1683	007326	012777	000000	170700	MOV	#0,AC			;LOAD AC WITH 0
1684	007334	012777	000036	170706	MOV	#36,LSH			;LOAD SHIFT COUNT (LSH) WITH 36
1685	007342	022777	040000	170664	CMP	#40000,AC			;COMPARE AC WITH 40000
1686	007350	001401			BEQ	118\$			;IF NO.ERROR SKIP HLT
1687	007352	104000			HLT				;CALL ERROR ROUTINE
1688	007354							118\$:	
1689	007354	022777	000000	170654	CMP	#0,AMQ			;COMPARE MQ WITH 0
1690	007362	001401			BEQ	119\$			;IF NO.ERROR SKIP HLT
1691	007364	104000			HLT				;CALL ERROR ROUTINE
1692	007366							119\$:	
1693	007366	122777	000010	170650	CMPB	#10,SR			;COMPARE SR WITH 10
1694	007374	001401			BEQ	120\$			;IF NO.ERROR SKIP HLT
1695	007376	104000			HLT				;CALL ERROR ROUTINE
1696	007400							120\$:	

```

1697
1698
1699
1700 007400
1701
1702
1703 007400 104400
1704 007402 012777 000000 170626
1705 007410 012777 000000 170616
1706 007416 012777 177760 170624
1707 007424 022777 000000 170602
1708 007432 001401
1709 007434 104000
1710 007436
1711 007436 022777 000000 170572
1712 007444 001401
1713 007446 104000
1714 007450
1715 007450 122777 000036 170566
1716 007456 001401
1717 007460 104000
1718 007462
1719
1720
1721 007462 104400
1722 007464 012777 000000 170544
1723 007472 012777 177777 170534
1724 007500 012777 177760 170542
1725 007506 022777 000000 170520
1726 007514 001401
1727 007516 104000
1728 007520
1729 007520 022777 177777 170510
1730 007526 001401
1731 007530 104000
1732 007532
1733 007532 122777 000020 170504
1734 007540 001401
1735 007542 104000
1736 007544
1737 007544 104400
1738 007546 012777 000000 170462
1739 007554 012777 177777 170452
1740 007562 012777 177741 170460
1741 007570 022777 000000 170436
1742 007576 001401
1743 007600 104000
1744 007602
1745 007602 022777 000001 170426

```

ACMQ:

```

SCOPE
MOV #0,AMQ
MOV #0,AC
MOV #-16,LSH
CMP #0,AC
BEQ 64$
HLT
64$:
CMP #0,AMQ
BEQ 65$
HLT
65$:
CMPB #36,SR
BEQ 66$
HLT
66$:
SCOPE
MOV #0,AMQ
MOV #-1,AC
MOV #-16,LSH
CMP #0,AC
BEQ 67$
HLT
67$:
CMP #-1,AMQ
BEQ 68$
HLT
68$:
CMPB #20,SR
BEQ 69$
HLT
69$:
SCOPE
MOV #0,AMQ
MOV #-1,AC
MOV #-31,LSH
CMP #0,AC
BEQ 70$
HLT
70$:
CMP #1,AMQ

```

```

;*****
; TEST OF AC SHIFT INTO MQ
;*****
;TEST OF LOGICAL SHIFT
;LOAD MQ WITH 0
;LOAD AC WITH 0
;LOAD SHIFT COUNT (LSH) WITH -16.
;COMPARE AC WITH 0
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;COMPARE MQ WITH 0
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;COMPARE SR WITH 36
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;TEST OF LOGICAL SHIFT
;LOAD MQ WITH 0
;LOAD AC WITH -1
;LOAD SHIFT COUNT (LSH) WITH -16.
;COMPARE AC WITH 0
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;COMPARE MQ WITH -1
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;COMPARE SR WITH 20
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;TEST OF LOGICAL SHIFT
;LOAD MQ WITH 0
;LOAD AC WITH -1
;LOAD SHIFT COUNT (LSH) WITH -31.
;COMPARE AC WITH 0
;IF NO.ERROR SKIP HLT
;CALL ERROR ROUTINE
;COMPARE MQ WITH 1

```

L03

1746	007610	001401			BEQ	71\$			; IF NO.ERROR SKIP HLT
1747	007612	104000			HLT				; CALL ERROR ROUTINE
1748	007614				71\$:				
1749	007614	122777	000023	170422	CMPB	#23, @SR			; COMPARE SR WITH 23
1750	007622	001401			BEQ	72\$			; IF NO.ERROR SKIP HLT
1751	007624	104000			HLT				; CALL ERROR ROUTINE
1752	007626				72\$:				
1753									
1754									
1755									
1756	007626	104400			SCOPE				; TEST OF LOGICAL SHIFT
1757	007630	012777	000000	170400	MOV	#0, @MQ			; LOAD MQ WITH 0
1758	007636	012777	125252	170370	MOV	#125252, @AC			; LOAD AC WITH 125252
1759	007644	012777	177760	170376	MOV	#-16, @LSH			; LOAD SHIFT COUNT (LSH) WITH -16.
1760	007652	022777	000000	170354	CMP	#0, @AC			; COMPARE AC WITH 0
1761	007660	001401			BEQ	73\$			; IF NO.ERROR SKIP HLT
1762	007662	104000			HLT				; CALL ERROR ROUTINE
1763	007664				73\$:				
1764	007664	022777	125252	170344	CMP	#125252, @MQ			; COMPARE MQ WITH 125252
1765	007672	001401			BEQ	74\$			; IF NO.ERROR SKIP HLT
1766	007674	104000			HLT				; CALL ERROR ROUTINE
1767	007676				74\$:				
1768	007676	122777	000020	170340	CMPB	#20, @SR			; COMPARE SR WITH 20
1769	007704	001401			BEQ	75\$			; IF NO.ERROR SKIP HLT
1770	007706	104000			HLT				; CALL ERROR ROUTINE
1771	007710				75\$:				
1772									
1773									
1774									
1775	007710	104400			SCOPE				; TEST OF LOGICAL SHIFT
1776	007712	012777	000000	170316	MOV	#0, @MQ			; LOAD MQ WITH 0
1777	007720	012777	052525	170306	MOV	#52525, @AC			; LOAD AC WITH 52525
1778	007726	012777	177777	170314	MOV	#-1, @LSH			; LOAD SHIFT COUNT (LSH) WITH -1
1779	007734	022777	025252	170272	CMP	#25252, @AC			; COMPARE AC WITH 25252
1780	007742	001401			BEQ	76\$			; IF NO.ERROR SKIP HLT
1781	007744	104000			HLT				; CALL ERROR ROUTINE
1782	007746				76\$:				
1783	007746	022777	100000	170262	CMP	#100000, @MQ			; COMPARE MQ WITH 100000
1784	007754	001401			BEQ	77\$			; IF NO.ERROR SKIP HLT
1785	007756	104000			HLT				; CALL ERROR ROUTINE
1786	007760				77\$:				
1787	007760	122777	000000	170256	CMPB	#0, @SR			; COMPARE SR WITH 0

1788	007766	001401			BEQ	78\$			; IF NO.ERROR SKIP HLT
1789	007770	104000			HLT				; CALL ERROR ROUTINE
1790	007772						78\$:		
1791									
1792	007772	104400			SCOPE				; TEST OF LOGICAL SHIFT
1793	007774	012777	000000	170234	MOV	#0,2MQ			; LOAD MQ WITH 0
1794	010002	012777	100000	170224	MOV	#100000,2AC			; LOAD AC WITH 100000
1795	010010	012777	177742	170232	MOV	#177742,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177742
1796	010016	022777	000000	170210	CMP	#0,2AC			; COMPARE AC WITH 0
1797	010024	001401			BEQ	79\$			; IF NO.ERROR SKIP HLT
1798	010026	104000			HLT				; CALL ERROR ROUTINE
1799	010030						79\$:		
1800	010030	022777	000002	170200	CMP	#2,2MQ			; COMPARE MQ WITH 2
1801	010036	001401			BEQ	80\$			; IF NO.ERROR SKIP HLT
1802	010040	104000			HLT				; CALL ERROR ROUTINE
1803	010042						80\$:		
1804	010042	122777	000022	170174	CMPB	#22,2SR			; COMPARE SR WITH 22
1805	010050	001401			BEQ	81\$			; IF NO.ERROR SKIP HLT
1806	010052	104000			HLT				; CALL ERROR ROUTINE
1807	010054						81\$:		
1808									
1809	010054	104400			SCOPE				; TEST OF LOGICAL SHIFT
1810	010056	012777	000000	170152	MOV	#0,2MQ			; LOAD MQ WITH 0
1811	010064	012777	100000	170142	MOV	#100000,2AC			; LOAD AC WITH 100000
1812	010072	012777	177743	170150	MOV	#177743,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177743
1813	010100	022777	000000	170126	CMP	#0,2AC			; COMPARE AC WITH 0
1814	010106	001401			BEQ	82\$			; IF NO.ERROR SKIP HLT
1815	010110	104000			HLT				; CALL ERROR ROUTINE
1816	010112						82\$:		
1817	010112	022777	000004	170116	CMP	#4,2MQ			; COMPARE MQ WITH 4
1818	010120	001401			BEQ	83\$			; IF NO.ERROR SKIP HLT
1819	010122	104000			HLT				; CALL ERROR ROUTINE
1820	010124						83\$:		
1821	010124	122777	000022	170112	CMPB	#22,2SR			; COMPARE SR WITH 22
1822	010132	001401			BEQ	84\$			; IF NO.ERROR SKIP HLT
1823	010134	104000			HLT				; CALL ERROR ROUTINE
1824	010136						84\$:		
1825									
1826									
1827	010136	104400			SCOPE				; TEST OF LOGICAL SHIFT
1828	010140	012777	000000	170070	MOV	#0,2MQ			; LOAD MQ WITH 0
1829	010146	012777	100000	170060	MOV	#100000,2AC			; LOAD AC WITH 100000
1830	010154	012777	177744	170066	MOV	#177744,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177744
1831	010162	022777	000000	170044	CMP	#0,2AC			; COMPARE AC WITH 0
1832	010170	001401			BEQ	85\$			; IF NO.ERROR SKIP HLT
1833	010172	104000			HLT				; CALL ERROR ROUTINE
1834	010174						85\$:		
1835	010174	022777	000010	170034	CMP	#10,2MQ			; COMPARE MQ WITH 10
1836	010202	001401			BEQ	86\$			; IF NO.ERROR SKIP HLT
1837	010204	104000			HLT				; CALL ERROR ROUTINE
1838	010206						86\$:		
1839	010206	122777	000022	170030	CMPB	#22,2SR			; COMPARE SR WITH 22

1840	010214	001401			BEQ	87\$		; IF NO.ERROR SKIP HLT
1841	010216	104000			HLT			; CALL ERROR ROUTINE
1842	010220						87\$:	
1843	010220	104400			SCOPE			; TEST OF LOGICAL SHIFT
1844	010222	012777	000000	170006	MOV	#0,AMQ		; LOAD MQ WITH 0
1845	010230	012777	100000	167776	MOV	#100000,AC		; LOAD AC WITH 100000
1846	010236	012777	177745	170004	MOV	#177745,LSH		; LOAD SHIFT COUNT (LSH) WITH 177745
1847	010244	022777	000000	167762	CMP	#0,AC		; COMPARE AC WITH 0
1848	010252	001401			BEQ	88\$		; IF NO.ERROR SKIP HLT
1849	010254	104000			HLT			; CALL ERROR ROUTINE
1850	010256						88\$:	
1851	010256	022777	000020	167752	CMP	#20,AMQ		; COMPARE MQ WITH 20
1852	010264	001401			BEQ	89\$		; IF NO.ERROR SKIP HLT
1853	010266	104000			HLT			; CALL ERROR ROUTINE
1854	010270						89\$:	
1855	010270	122777	000022	167746	CMPB	#22,SR		; COMPARE SR WITH 22
1856	010276	001401			BEQ	90\$		; IF NO.ERROR SKIP HLT
1857	010300	104000			HLT			; CALL ERROR ROUTINE
1858	010302						90\$:	
1859								
1860								
1861	010302	104400			SCOPE			; TEST OF LOGICAL SHIFT
1862	010304	012777	000000	167724	MOV	#0,AMQ		; LOAD MQ WITH 0
1863	010312	012777	100000	167714	MOV	#100000,AC		; LOAD AC WITH 100000
1864	010320	012777	177746	167722	MOV	#177746,LSH		; LOAD SHIFT COUNT (LSH) WITH 177746
1865	010326	022777	000000	167700	CMP	#0,AC		; COMPARE AC WITH 0
1866	010334	001401			BEQ	91\$		; IF NO.ERROR SKIP HLT
1867	010336	104000			HLT			; CALL ERROR ROUTINE
1868	010340						91\$:	
1869	010340	104000			HLT			; HALT ON ERROR
1870	010342	022777	000040	167666	CMP	#40,AMQ		; COMPARE MQ WITH 40
1871	010350	001401			BEQ	92\$		; IF NO.ERROR SKIP HLT
1872	010352	104000			HLT			; CALL ERROR ROUTINE
1873	010354						92\$:	
1874	010354	122777	000022	167662	CMPB	#22,SR		; COMPARE SR WITH 22
1875	010362	001401			BEQ	93\$		; IF NO.ERROR SKIP HLT
1876	010364	104000			HLT			; CALL ERROR ROUTINE
1877	010366						93\$:	
1878								
1879								
1880	010366	104400			SCOPE			; TEST OF LOGICAL SHIFT
1881	010370	012777	000000	167640	MOV	#0,AMQ		; LOAD MQ WITH 0
1882	010376	012777	100000	167630	MOV	#100000,AC		; LOAD AC WITH 100000
1883	010404	012777	177747	167636	MOV	#177747,LSH		; LOAD SHIFT COUNT (LSH) WITH 177747
1884	010412	022777	000000	167614	CMP	#0,AC		; COMPARE AC WITH 0
1885	010420	001401			BEQ	94\$		; IF NO.ERROR SKIP HLT
1886	010422	104000			HLT			; CALL ERROR ROUTINE
1887	010424						94\$:	
1888	010424	022777	000100	167604	CMP	#100,AMQ		; COMPARE MQ WITH 100
1889	010432	001401			BEQ	95\$		; IF NO.ERROR SKIP HLT
1890	010434	104000			HLT			; CALL ERROR ROUTINE
1891	010436						95\$:	
1892	010436	122777	000022	167600	CMPB	#22,SR		; COMPARE SR WITH 22



1893	010444	001401				BEQ	96\$		: IF NO.ERROR SKIP HLT
1894	010446	104000				HLT			: CALL ERROR ROUTINE
1895	010450				96\$:				
1896	010450	104400				SCOPE			: TEST OF LOGICAL SHIFT
1897	010452	012777	000000	167556		MOV	#0,2MQ		: LOAD MQ WITH 0
1898	010460	012777	100000	167546		MOV	#100000,2AC		: LOAD AC WITH 100000
1899	010466	012777	177750	167554		MOV	#177750,2LSH		: LOAD SHIFT COUNT (LSH) WITH 177750
1900	010474	022777	000000	167532		CMP	#0,2AC		: COMPARE AC WITH 0
1901	010502	001401				BEQ	97\$		: IF NO.ERROR SKIP HLT
1902	010504	104000				HLT			: CALL ERROR ROUTINE
1903	010506				97\$:				
1904	010508	022777	000200	167522		CMP	#200,2MQ		: COMPARE MQ WITH 200
1905	010514	001401				BEQ	98\$		: IF NO.ERROR SKIP HLT
1906	010516	104000				HLT			: CALL ERROR ROUTINE
1907	010520				98\$:				
1908	010520	122777	000022	167516		CMPB	#22,2SR		: COMPARE SR WITH 22
1909	010526	001401				BEQ	99\$		: IF NO.ERROR SKIP HLT
1910	010530	104000				HLT			: CALL ERROR ROUTINE
1911	010532				99\$:				
1912									
1913									
1914	010532	104400				SCOPE			: TEST OF LOGICAL SHIFT
1915	010534	012777	000000	167474		MOV	#0,2MQ		: LOAD MQ WITH 0
1916	010542	012777	100000	167464		MOV	#100000,2AC		: LOAD AC WITH 100000
1917	010550	012777	177751	167472		MOV	#177751,2LSH		: LOAD SHIFT COUNT (LSH) WITH 177751
1918	010556	022777	000000	167450		CMP	#0,2AC		: COMPARE AC WITH 0
1919	010564	001401				BEQ	100\$		: IF NO.ERROR SKIP HLT
1920	010566	104000				HLT			: CALL ERROR ROUTINE
1921	010570				100\$:				
1922	010570	022777	000400	167440		CMP	#400,2MQ		: COMPARE MQ WITH 400
1923	010576	001401				BEQ	101\$		: IF NO.ERROR SKIP HLT
1924	010600	104000				HLT			: CALL ERROR ROUTINE
1925	010602				101\$:				
1926	010602	122777	000022	167434		CMPB	#22,2SR		: COMPARE SR WITH 22
1927	010610	001401				BEQ	102\$		: IF NO.ERROR SKIP HLT
1928	010612	104000				HLT			: CALL ERROR ROUTINE
1929	010614				102\$:				
1930									
1931									
1932	010614	104400				SCOPE			: TEST OF LOGICAL SHIFT
1933	010616	012777	000000	167412		MOV	#0,2MQ		: LOAD MQ WITH 0
1934	010624	012777	100000	167402		MOV	#100000,2AC		: LOAD AC WITH 100000
1935	010632	012777	177752	167410		MOV	#177752,2LSH		: LOAD SHIFT COUNT (LSH) WITH 177752
1936	010640	022777	000000	167366		CMP	#0,2AC		: COMPARE AC WITH 0
1937	010646	001401				BEQ	103\$		: IF NO.ERROR SKIP HLT
1938	010650	104000				HLT			: CALL ERROR ROUTINE
1939	010652				103\$:				
1940	010652	022777	001000	167356		CMP	#1000,2MQ		: COMPARE MQ WITH 1000
1941	010660	001401				BEQ	104\$		: IF NO.ERROR SKIP HLT
1942	010662	104000				HLT			: CALL ERROR ROUTINE
1943	010664				104\$:				
1944	010664	122777	000022	167352		CMPB	#22,2SR		: COMPARE SR WITH 22



```
2001 :*****
2002 : AT THIS POINT, THE LOGICAL SHIFT WORKS
2003 :*****
2004 :
2005 :*****
2006 : TEST OF ARITHMETIC SHIFT
2007 :
2008 : SHIFT RIGHT
2009 :*****
2010 ASR:
2011
2012
2013 011124 104400 SCOPE ;TEST OF ARITHMETIC SHIFT
2014 011126 012777 000000 167102 MOV #0,AMQ ;LOAD MQ WITH 0
2015 011134 012777 000000 167072 MOV #0,AC ;LOAD AC WITH 0
2016 011142 012777 177760 167102 MOV #-16,ASH ;LOAD SHIFT COUNT (ASH) WITH -16.
2017 011150 022777 000000 167056 CMP #0,AC ;COMPARE AC WITH 0
2018 011156 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT
2019 011160 104000 HLT ;CALL ERROR ROUTINE
2020 011162
2021 011162 022777 000000 167046 64$: CMP #0,AMQ ;COMPARE MQ WITH 0
2022 011170 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT
2023 011172 104000 HLT ;CALL ERROR ROUTINE
2024 011174
2025 011174 122777 000036 167042 65$: CMPB #36,SR ;COMPARE SR WITH 36
2026 011202 001401 BEQ 66$ ;IF NO.ERROR SKIP HLT
2027 011204 104000 HLT ;CALL ERROR ROUTINE
2028 011206
2029
2030
2031
2032 011206 104400 SCOPE ;TEST OF ARITHMETIC SHIFT
2033 011210 012777 000000 167020 MOV #0,AMQ ;LOAD MQ WITH 0
2034 011216 012777 177777 167010 MOV #-1,AC ;LOAD AC WITH -1
2035 011224 012777 177760 167020 MOV #-16,ASH ;LOAD SHIFT COUNT (ASH) WITH -16.
2036 011232 022777 177777 166774 CMP #-1,AC ;COMPARE AC WITH -1
2037 011240 001401 BEQ 67$ ;IF NO.ERROR SKIP HLT
2038 011242 104000 HLT ;CALL ERROR ROUTINE
2039 011244
2040 011244 022777 177777 166764 67$: CMP #-1,AMQ ;COMPARE MQ WITH -1
2041 011252 001401 BEQ 68$ ;IF NO.ERROR SKIP HLT
2042 011254 104000 HLT ;CALL ERROR ROUTINE
2043 011256
2044 011256 122777 000342 166760 68$: CMPB #342,SR ;COMPARE SR WITH 342
2045 011264 001401 BEQ 69$ ;IF NO.ERROR SKIP HLT
2046 011266 104000 HLT ;CALL ERROR ROUTINE
2047 011270
2048
2049
2050 011270 104400 SCOPE ;TEST OF ARITHMETIC SHIFT
2051 011272 012777 000000 166736 MOV #0,AMQ ;LOAD MQ WITH 0
2052 011300 012777 125252 166726 MOV #125252,AC ;LOAD AC WITH 125252
2053 011306 012777 177760 166736 MOV #-16,ASH ;LOAD SHIFT COUNT (ASH) WITH -16.
2054 011314 022777 177777 166712 CMP #-1,AC ;COMPARE AC WITH -1
2055 011322 001401 BEQ 70$ ;IF NO.ERROR SKIP HLT
2056 011324 104000 HLT ;CALL ERROR ROUTINE
```



```

2103
2104
2105
2106 011516
2107 011516 104400
2108 011520 012777 000000 166510
2109 011526 012777 000000 166500
2110 011534 012777 000020 166510
2111 011542 022777 000000 166464
2112 011550 001401
2113 011552 104000
2114 011554
2115 011554 022777 000000 166454
2116 011562 001401
2117 011564 104000
2118 011566
2119 011566 122777 000036 166450
2120 011574 001401
2121 011576 104000
2122 011600
2123 011600 104400
2124 011602 012777 177777 166426
2125 011610 012777 000000 166416
2126 011616 012777 000017 166426
2127 011624 022777 077777 166402
2128 011632 001401
2129 011634 104000
2130 011636
2131 011636 022777 100000 166372
2132 011644 001401
2133 011646 104000
2134 011650
2135 011650 122777 000000 166366
2136 011656 001401
2137 011660 104000
2138 011662
2139
2140 011662 104400
2141 011664 012777 177777 166344
2142 011672 012777 000000 166334
2143 011700 012777 000020 166344
2144 011706 022777 077777 166320
2145 011714 001401
2146 011716 104000
2147 011720
2148 011720 022777 000000 166310
2149 011726 001401
2150 011730 104000
2151 011732
2152 011732 122777 000211 166304

```

;\*\*\*\*\*  
;SHIFT LEFT  
;\*\*\*\*\*  
ASL:
SCOPE ;TEST OF ARITHMETIC SHIFT
MOV #0,AMQ ;LOAD MQ WITH 0
MOV #0,AC ;LOAD AC WITH 0
MOV #16,ASH ;LOAD SHIFT COUNT (ASH) WITH 16.
CMP #0,AC ;COMPARE AC WITH 0
BEQ 64\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
64\$:
CMP #0,AMQ ;COMPARE MQ WITH 0
BEQ 65\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
65\$:
CMPB #36,SR ;COMPARE SR WITH 36
BEQ 66\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
66\$:
SCOPE ;TEST OF ARITHMETIC SHIFT
MOV #-1,AMQ ;LOAD MQ WITH -1
MOV #0,AC ;LOAD AC WITH 0
MOV #15,ASH ;LOAD SHIFT COUNT (ASH) WITH 15.
CMP #77777,AC ;COMPARE AC WITH 77777
BEQ 67\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
67\$:
CMP #100000,AMQ ;COMPARE MQ WITH 100000
BEQ 68\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
68\$:
CMPB #0,SR ;COMPARE SR WITH 0
BEQ 69\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
69\$:
SCOPE ;TEST OF ARITHMETIC SHIFT
MOV #-1,AMQ ;LOAD MQ WITH -1
MOV #0,AC ;LOAD AC WITH 0
MOV #16,ASH ;LOAD SHIFT COUNT (ASH) WITH 16.
CMP #77777,AC ;COMPARE AC WITH 77777
BEQ 70\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
70\$:
CMP #0,AMQ ;COMPARE MQ WITH 0
BEQ 71\$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
71\$:
CMPB #211,SR ;COMPARE SR WITH 211

2153	011740	001401			BEQ	72\$			;IF NO.ERROR SKIP HLT
2154	011742	104000			HLT				;CALL ERROR ROUTINE
2155	011744					72\$:			
2156									
2157	011744	104400			SCOPE				;TEST OF ARITHMETIC SHIFT
2158	011746	012777	177776	166262	MOV	#-2,AMQ			;LOAD MQ WITH -2
2159	011754	012777	100000	166252	MOV	#100000,AC			;LOAD AC WITH 100000
2160	011762	012777	000017	166262	MOV	#15,ASH			;LOAD SHIFT COUNT (ASH) WITH 15.
2161	011770	022777	177777	166236	CMP	#-1,AC			;COMPARE AC WITH -1
2162	011776	001401			BEQ	73\$			;IF NO.ERROR SKIP HLT
2163	012000	104000			HLT				;CALL ERROR ROUTINE
2164	012002					73\$:			
2165	012002	022777	000000	166226	CMP	#0,AMQ			;COMPARE MQ WITH 0
2166	012010	001401			BEQ	74\$			;IF NO.ERROR SKIP HLT
2167	012012	104000			HLT				;CALL ERROR ROUTINE
2168	012014					74\$:			
2169	012014	122777	000150	166222	CMPB	#150,SR			;COMPARE SR WITH 150
2170	012022	001401			BEQ	75\$			;IF NO.ERROR SKIP HLT
2171	012024	104000			HLT				;CALL ERROR ROUTINE
2172	012026					75\$:			
2173	012026	104400			SCOPE				;TEST OF ARITHMETIC SHIFT
2174	012030	012777	125252	166200	MOV	#125252,AMQ			;LOAD MQ WITH 125252
2175	012036	012777	000000	166170	MOV	#0,AC			;LOAD AC WITH 0
2176	012044	012777	000017	166200	MOV	#15,ASH			;LOAD SHIFT COUNT (ASH) WITH 15.
2177	012052	022777	052525	166154	CMP	#52525,AC			;COMPARE AC WITH 52525
2178	012060	001401			BEQ	76\$			;IF NO.ERROR SKIP HLT
2179	012062	104000			HLT				;CALL ERROR ROUTINE
2180	012064					76\$:			
2181	012064	022777	000000	166144	CMP	#0,AMQ			;COMPARE MQ WITH 0
2182	012072	001401			BEQ	77\$			;IF NO.ERROR SKIP HLT
2183	012074	104000			HLT				;CALL ERROR ROUTINE
2184	012076					77\$:			
2185	012076	122777	000010	166140	CMPB	#10,SR			;COMPARE SR WITH 10
2186	012104	001401			BEQ	78\$			;IF NO.ERROR SKIP HLT
2187	012106	104000			HLT				;CALL ERROR ROUTINE
2188	012110					78\$:			
2189									
2190	012110	104400			SCOPE				;TEST OF ARITHMETIC SHIFT
2191	012112	012777	125252	166116	MOV	#125252,AMQ			;LOAD MQ WITH 125252
2192	012120	012777	000000	166106	MOV	#0,AC			;LOAD AC WITH 0
2193	012126	012777	000020	166116	MOV	#16,ASH			;LOAD SHIFT COUNT (ASH) WITH 16.
2194	012134	022777	025252	166072	CMP	#25252,AC			;COMPARE AC WITH 25252
2195	012142	001401			BEQ	79\$			;IF NO.ERROR SKIP HLT
2196	012144	104000			HLT				;CALL ERROR ROUTINE
2197	012146					79\$:			
2198	012146	022777	000000	166062	CMP	#0,AMQ			;COMPARE MQ WITH 0

H04

```

2199 012154 001401          BEQ      80$      ; IF NO.ERROR SKIP HLT
2200 012156 104000          HLT             ; CALL ERROR ROUTINE
2201 012160          80$:
2202 012160 122777 000211 166056 CMPB     #211,SR   ; COMPARE SR WITH 211
2203 012166 001401          BEQ      81$      ; IF NO.ERROR SKIP HLT
2204 012170 104000          HLT             ; CALL ERROR ROUTINE
2205 012172          81$:
2206 012172 104400          SCOPE
2207 012174 012777 052525 166034 MOV      #52525,MQ ; TEST OF ARITHMETIC SHIFT
2208 012202 012777 000000 166024 MOV      #0,AC     ; LOAD MQ WITH 52525
2209 012210 012777 000020 166034 MOV      #16,ASH   ; LOAD AC WITH 0
2210 012216 022777 052525 166010 CMP      #52525,AC ; LOAD SHIFT COUNT (ASH) WITH 16.
2211 012224 001401          BEQ      82$      ; COMPARE AC WITH 52525
2212 012226 104000          HLT             ; IF NO.ERROR SKIP HLT
2213 012230          82$:
2214 012230 022777 000000 166000 CMP      #0,MQ     ; CALL ERROR ROUTINE
2215 012236 001401          BEQ      83$      ; COMPARE MQ WITH 0
2216 012240 104000          HLT             ; IF NO.ERROR SKIP HLT
2217 012242          83$:
2218 012242 122777 000010 165774 CMPB     #10,SR   ; CALL ERROR ROUTINE
2219 012250 001401          BEQ      84$      ; COMPARE SR WITH 10
2220 012252 104000          HLT             ; IF NO.ERROR SKIP HLT
2221 012254          84$:
2222          84$:
2223 012254 104400          SCOPE
2224 012256 012777 000000 165752 MOV      #0,MQ     ; TEST OF ARITHMETIC SHIFT
2225 012264 012777 177777 165742 MOV      #-1,AC    ; LOAD MQ WITH 0
2226 012272 012777 000020 165752 MOV      #16,ASH   ; LOAD AC WITH -1
2227 012300 022777 100000 165726 CMP      #100000,AC ; LOAD SHIFT COUNT (ASH) WITH 16.
2228 012306 001401          BEQ      85$      ; COMPARE AC WITH 100000
2229 012310 104000          HLT             ; IF NO.ERROR SKIP HLT
2230 012312          85$:
2231 012312 022777 000000 165716 CMP      #0,MQ     ; CALL ERROR ROUTINE
2232 012320 001401          BEQ      86$      ; COMPARE MQ WITH 0
2233 012322 104000          HLT             ; IF NO.ERROR SKIP HLT
2234 012324          86$:
2235 012324 122777 000110 165712 CMPB     #110,SR  ; CALL ERROR ROUTINE
2236 012332 001401          BEQ      87$      ; COMPARE SR WITH 110
2237 012334 104000          HLT             ; IF NO.ERROR SKIP HLT
2238 012336          87$:

```

```

2239
2240
2241
2242
2243
2244
2245 012336
2246
2247 012336 104400
2248 012340 012777 000000 165670
2249 012346 012777 000000 165660
2250 012354 005077 165666
2251 012360 022777 000000 165646
2252 012366 001401
2253 012370 104000
2254 012372
2255 012372 022777 000000 165636
2256 012400 001401
2257 012402 104000
2258 012404
2259 012404 022777 017037 165630
2260
2261 012412 001401
2262 012414 104000
2263 012416
2264
2265
2266 012416 104400
2267 012420 012777 177777 165610
2268 012426 012777 177777 165600
2269 012434 005077 165606
2270 012440 022777 140000 165566
2271 012446 001401
2272 012450 104000
2273 012452
2274 012452 022777 000000 165556
2275 012460 001401
2276 012462 104000
2277 012464
2278 012464 022777 144036 165550
2279

```

```

;*****
; AT THIS POINT, THE ARITHMETIC SHIFT WORKS
;*****
;*****
; TEST OF NORMALIZE
;*****
NORMAL:
SCOPE
MOV #0, MQ ; TEST OF NORMALIZE
MOV #0, AC ; LOAD MQ WITH 0
CLR ANOR ; LOAD AC WITH 0
CMP #0, AC ; START NORMALIZE
BEQ 64$ ; COMPARE AC WITH 0
HLT ; IF NO.ERROR SKIP HLT
; CALL ERROR ROUTINE
64$:
CMP #0, MQ ; COMPARE MQ WITH 0
BEQ 65$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE
65$:
CMP #017037, ASC ; COMPARE SC WITH 37
BEQ 66$ ; AND SR= 36
HLT ; IF NO.ERROR SKIP HLT
; CALL ERROR ROUTINE
66$:
SCOPE
MOV #-1, MQ ; TEST OF NORMALIZE
MOV #-1, AC ; LOAD MQ WITH -1
CLR ANOR ; LOAD AC WITH -1
CMP #140000, AC ; START NORMALIZE
BEQ 67$ ; COMPARE AC WITH 140000
HLT ; IF NO.ERROR SKIP HLT
; CALL ERROR ROUTINE
67$:
CMP #0, MQ ; COMPARE MQ WITH 0
BEQ 68$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE
68$:
CMP #144036, ASC ; COMPARE SC WITH 30.
; AND SR= 310

```





2327	012666	001401					BEQ	76\$		: IF NO.ERROR SKIP HLT
2328	012670	104000					HLT			: CALL ERROR ROUTINE
2329	012672				76\$:					
2330	012672	022777	000000	165336			CMP	#0,AMQ		: COMPARE MQ WITH 0
2331	012700	001401					BEQ	77\$		: IF NO.ERROR SKIP HLT
2332	012702	104000					HLT			: CALL ERROR ROUTINE
2333	012704				77\$:					
2334	012704	022777	004034	165330			CMP	#004034,ASC		: COMPARE SC WITH 28.
2335										: AND SR= 10
2336	012712	001401					BEQ	78\$		: IF NO.ERROR SKIP HLT
2337	012714	104000					HLT			: CALL ERROR ROUTINE
2338	012716				78\$:					
2339										
2340	012716	104400					SCOPE			: TEST OF NORMALIZE
2341	012720	012777	000001	165310			MOV	#1,AMQ		: LOAD MQ WITH 1
2342	012726	012777	100000	165300			MOV	#100000,ASC		: LOAD AC WITH 100000
2343	012734	005077	165306				CLR	ANOR		: START NORMALIZE
2344	012740	022777	100000	165266			CMP	#100000,ASC		: COMPARE AC WITH 100000
2345	012746	001401					BEQ	79\$		: IF NO.ERROR SKIP HLT
2346	012750	104000					HLT			: CALL ERROR ROUTINE
2347	012752				79\$:					
2348	012752	022777	000001	165256			CMP	#1,AMQ		: COMPARE MQ WITH 1
2349	012760	001401					BEQ	80\$		: IF NO.ERROR SKIP HLT
2350	012762	104000					HLT			: CALL ERROR ROUTINE
2351	012764				80\$:					
2352	012764	022777	140000	165250			CMP	#140000,ASC		: COMPARE SC WITH 0
2353										: AND SR= 300
2354	012772	001401					BEQ	81\$		: IF NO.ERROR SKIP HLT
2355	012774	104000					HLT			: CALL ERROR ROUTINE
2356	012776				81\$:					
2357										
2358										
2359										
2360	012776	104400					SCOPE			: TEST OF NORMALIZE
2361	013000	012777	125252	165230			MOV	#125252,AMQ		: LOAD MQ WITH 125252
2362	013006	012777	170000	165220			MOV	#170000,ASC		: LOAD AC WITH 170000
2363	013014	005077	165226				CLR	ANOR		: START NORMALIZE
2364	013020	022777	100005	165206			CMP	#100005,ASC		: COMPARE AC WITH 100005
2365	013026	001401					BEQ	82\$		: IF NO.ERROR SKIP HLT
2366	013030	104000					HLT			: CALL ERROR ROUTINE
2367	013032				82\$:					
2368	013032	022777	052520	165176			CMP	#52520,AMQ		: COMPARE MQ WITH 52520
2369	013040	001401					BEQ	83\$		: IF NO.ERROR SKIP HLT
2370	013042	104000					HLT			: CALL ERROR ROUTINE
2371	013044				83\$:					
2372	013044	022777	140003	165170			CMP	#140003,ASC		: COMPARE SC WITH 3
2373										: AND SR= 300
2374	013052	001401					BEQ	84\$		: IF NO.ERROR SKIP HLT
2375	013054	104000					HLT			: CALL ERROR ROUTINE
2376	013056				84\$:					

2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416

```
*****  
: AT THIS POINT NORMALIZE WORKS  
:*****
```

```
*****  
: SKIP MULT AND DIVIDE TEST  
: IF BIT 2 IS SET  
:*****
```

```
013056 004767 004266 JSR %7,CKSWR ;CHECK FOR 1G  
013062 032777 000002 BIT #2,JSWR  
013070 001402 BEQ .DIV  
013072 000167 003042 JMP .DEV
```

```
*****  
: TEST OF MULTIPLY  
:*****  
.DIV:
```

```
013076 104400 SCOPE ;TEST OF MULTIPLY  
013100 012777 000000 165130 MOV #0,AMQ ;LOAD MQ WITH 0  
013106 012777 000000 165124 MOV #0,AMUL ;LOAD MUL WITH 0 AND MULTIPLY  
013114 022777 000000 165112 CMP #0,AC ;COMPARE AC WITH 0  
013122 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT  
013124 104000 HLT ;CALL ERROR ROUTINE  
013126 022777 000000 165102 64$: CMP #0,AMQ ;COMPARE MQ WITH 0  
013134 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT  
013136 104000 HLT ;CALL ERROR ROUTINE  
013140 122777 000036 165076 65$: CMPB #36,JSR ;COMPARE SR WITH 36
```

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 52

2417	013146	001401			BEQ	66\$			
2418	013150	104000			HLT				; IF NO.ERROR SKIP HLT
2419	013152						66\$:		; CALL ERROR ROUTINE
2420									
2421									
2422	013152	104400			SCOPE				; TEST OF MULTIPLY
2423	013154	012777	177777	165054	MOV	#-1, MQ			; LOAD MQ WITH -1
2424	013162	012777	000001	165050	MOV	#1, MQ			; LOAD MUL WITH 1 AND MULTIPLY
2425	013170	022777	177777	165036	CMP	#-1, AC			; COMPARE AC WITH -1
2426	013176	001401			BEQ	67\$			; IF NO.ERROR SKIP HLT
2427	013200	104000			HLT				; CALL ERROR ROUTINE
2428	013202						67\$:		
2429	013202	022777	177777	165026	CMP	#-1, MQ			; COMPARE MQ WITH -1
2430	013210	001401			BEQ	68\$			; IF NO.ERROR SKIP HLT
2431	013212	104000			HLT				; CALL ERROR ROUTINE
2432	013214						68\$:		
2433	013214	122777	000342	165022	CMPB	#342, SR			; COMPARE SR WITH 342
2434	013222	001401			BEQ	69\$			; IF NO.ERROR SKIP HLT
2435	013224	104000			HLT				; CALL ERROR ROUTINE
2436	013226						69\$:		
2437									
2438									
2439									
2440	013226	104400			SCOPE				; TEST OF MULTIPLY
2441	013230	012777	125252	165000	MOV	#125252, MQ			; LOAD MQ WITH 125252
2442	013236	012777	000002	164774	MOV	#2, MUL			; LOAD MUL WITH 2 AND MULTIPLY
2443	013244	022777	177777	164762	CMP	#-1, AC			; COMPARE AC WITH -1
2444	013252	001401			BEQ	70\$			; IF NO.ERROR SKIP HLT
2445	013254	104000			HLT				; CALL ERROR ROUTINE
2446	013256						70\$:		
2447	013256	022777	052524	164752	CMP	#52524, MQ			; COMPARE MQ WITH 52524
2448	013264	001401			BEQ	71\$			; IF NO.ERROR SKIP HLT
2449	013266	104000			HLT				; CALL ERROR ROUTINE
2450	013270						71\$:		
2451	013270	122777	000340	164746	CMPB	#340, SR			; COMPARE SR WITH 340

2452	013276	001401			BEQ	72\$	: IF NO.ERROR SKIP HLT
2453	013300	104000			HLT		: CALL ERROR ROUTINE
2454	013302				72\$:		
2455	013302	104400			SCOPE		: TEST OF MULTIPLY
2456	013304	012777	052525	164724	MOV	#52525,AMQ	: LOAD MQ WITH 52525
2457	013312	012777	000002	164720	MOV	#2,AMUL	: LOAD MUL WITH 2 AND MULTIPLY
2458	013320	022777	000000	164706	CMP	#0,AC	: COMPARE AC WITH 0
2459	013326	001401			BEQ	73\$	: IF NO.ERROR SKIP HLT
2460	013330	104000			HLT		: CALL ERROR ROUTINE
2461	013332				73\$:		
2462	013332	022777	125252	164676	CMP	#125252,AMQ	: COMPARE MQ WITH 125252
2463	013340	001401			BEQ	74\$	: IF NO.ERROR SKIP HLT
2464	013342	104000			HLT		: CALL ERROR ROUTINE
2465	013344				74\$:		
2466	013344	122777	000020	164672	CMPB	#20,ASR	: COMPARE SR WITH 20
2467	013352	001401			BEQ	75\$	: IF NO.ERROR SKIP HLT
2468	013354	104000			HLT		: CALL ERROR ROUTINE
2469	013356				75\$:		
2470							
2471							
2472							
2473	013356	104400			SCOPE		: TEST OF MULTIPLY
2474	013360	012777	125252	164650	MOV	#125252,AMQ	: LOAD MQ WITH 125252
2475	013366	012777	040000	164644	MOV	#40000,AMUL	: LOAD MUL WITH 40000 AND MULTIPLY
2476	013374	022777	165252	164632	CMP	#165252,AC	: COMPARE AC WITH 165252
2477	013402	001401			BEQ	76\$	: IF NO.ERROR SKIP HLT
2478	013404	104000			HLT		: CALL ERROR ROUTINE
2479	013406				76\$:		
2480	013406	022777	100000	164622	CMP	#100000,AMQ	: COMPARE MQ WITH 100000
2481	013414	001401			BEQ	77\$	: IF NO.ERROR SKIP HLT
2482	013416	104000			HLT		: CALL ERROR ROUTINE
2483	013420				77\$:		
2484	013420	122777	000300	164616	CMPB	#300,ASR	: COMPARE SR WITH 300
2485	013426	001401			BEQ	78\$	: IF NO.ERROR SKIP HLT
2486	013430	104000			HLT		: CALL ERROR ROUTINE
2487	013432				78\$:		
2488							
2489							
2490	013432	104400			SCOPE		: TEST OF MULTIPLY
2491	013434	012777	100000	164574	MOV	#100000,AMQ	: LOAD MQ WITH 100000
2492	013442	012777	100000	164570	MOV	#100000,AMUL	: LOAD MUL WITH 100000 AND MULTIPLY
2493	013450	022777	040000	164556	CMP	#40000,AC	: COMPARE AC WITH 40000
2494	013456	001401			BEQ	79\$	: IF NO.ERROR SKIP HLT
2495	013460	104000			HLT		: CALL ERROR ROUTINE
2496	013462				79\$:		
2497	013462	022777	000000	164546	CMP	#0,AMQ	: COMPARE MQ WITH 0
2498	013470	001401			BEQ	80\$	: IF NO.ERROR SKIP HLT
2499	013472	104000			HLT		: CALL ERROR ROUTINE
2500	013474				80\$:		
2501	013474	122777	000010	164542	CMPB	#10,ASR	: COMPARE SR WITH 10
2502	013502	001401			BEQ	81\$	: IF NO.ERROR SKIP HLT
2503	013504	104000			HLT		: CALL ERROR ROUTINE
2504	013506				81\$:		

```

2505
2506
2507
2508 013506
2509 013506 104400
2510 013510 005077 164522
2511 013514 005077 164512
2512 013520 106177 164520
2513 013524 102401
2514 013526 104000
2515 013530
2516 013530 005777 164500
2517 013534 100405
2518 013536 106177 164502
2519 013542 100006
2520 013544 104000
2521 013546 000404
2522 013550 106177 164470
2523 013554 100401
2524 013556 104000
2525
2526
2527 013560
2528 013560 104400
2529 013562 005077 164450
2530 013566 012777 052525 164440
2531 013574 012777 052525 164430
2532 013602 106177 164436
2533 013606 102401
2534 013610 104000
2535 013612
2536 013612 005777 164416
2537 013616 100405
2538 013620 106177 164420
2539 013624 100006
2540 013626 104000
2541 013630 000404
2542 013632 106177 164406
2543 013636 100401
2544 013640 104000
2545 013642
2546 013642 104400
2547 013644 012777 177777 164364
2548 013652 012777 177777 164354
2549 013660 012777 000001 164344
2550 013666 022777 000000 164340
2551 013674 001401
2552 013676 104000
2553 013700
2554 013700 022777 177777 164330
2555 013706 001401
2556 013710 104000
2557 013712 122777 000320 164324
2558 013720 001401
2559 013722 104000
2560 013724

```

```

*****
: TEST OF DIVIDE
*****
DIVIDE:
SCOPE
CLR 2MQ
CLR 2DIV
ROLB 2SR
BVS 1S
HLT

1S:
TST 2AC
BMI .MIN
ROLB 2SR
BPL .CONT
HLT
BR .CONT
.MIN: ROLB 2SR
.BLT: BMI .CONT
.HLT: HLT

.CONT:
SCOPE
CLR 2MQ
MOV 52525, 2AC
MOV 52525, 2DIV
ROLB 2SR
BVS 1S
HLT

1S:
TST 2AC
BMI .MIN1
ROLB 2SR
BPL .CONT1
HLT
BR .CONT1
.MIN1: ROLB 2SR
.BLT1: BMI .CONT1
.CONT1: HLT

SCOPE
MOV -1, 2MQ
MOV -1, 2AC
MOV 1, 2DIV
CMP 0, 2AC
BEQ 64S
HLT

64S:
CMP -1, 2MQ
BEQ 65S
HLT

65S:
CMPB 320, 2SR
BEQ 66S
HLT

66S:
*****
: TEST OF DIVIDE
: LOAD MQ WITH 0
: LOAD DIV WITH 0 AND DIVIDE
: SHIFT OVERFLOW BIT INTO PS
: SKIP HALT IF GOOD
: HALT ON ERROR

: CHECK AC'S SIGN

: SET APROPRIATE N AND V BITS

: WRONG SIGN

: SET APROPRIATE N AND V BITS

: TEST OF DIVIDE
: CLEAR THE MQ
: LOAD AC WITH 52525
: LOAD DIV WITH 52525 AND DIVIDE
: SHIFT OVERFLOW BIT INTO PS
: SKIP HALT IF GOOD
: HALT ON ERROR

: CHECK AC'S SIGN

: SET APROPRIATE N AND V BITS

: WRONG SIGN

: SET APROPRIATE N AND V BITS

: TEST OF DIVIDE
: LOAD MQ WITH -1
: LOAD AC WITH -1
: LOAD DIV WITH 1 AND DIVIDE
: COMPARE AC WITH 0 (REMAINDER)
: IF NO.ERROR SKIP HLT
: CALL ERROR ROUTINE

: COMPARE MQ WITH -1 (QUOTIANT)
: IF NO.ERROR SKIP HLT
: CALL ERROR ROUTINE

: COMPARE SR WITH 320
: IF NO.ERROR SKIP HLT
: CALL ERROR ROUTINE

```



```

2617
2618 014152 104400        SCOPE                ; TEST OF DIVIDE
2619 014154 012777 177777 164054  MOV  #-1, QMQ      ; LOAD MQ WITH -1
2620 014162 012777 177777 164044  MOV  #-1, QAC      ; LOAD AC WITH -1
2621 014170 012777 177777 164034  MOV  #-1, QDIV     ; LOAD DIV WITH -1 AND DIVIDE
2622 014176 022777 000000 164030  CMP  #0, QAC       ; COMPARE AC WITH 0 (REMAINDER)
2623 014204 001401        BEQ  76$           ; IF NO. ERROR SKIP HLT
2624 014206 104000        HLT                 ; CALL ERROR ROUTINE
2625 014210                76$:
2626 014210 022777 000001 164020  CMP  #1, QMQ      ; COMPARE MQ WITH 1 (QUOTIANT)
2627 014216 001401        BEQ  77$           ; IF NO. ERROR SKIP HLT
2628 014220 104000        HLT                 ; CALL ERROR ROUTINE
2629 014222                77$:
2630 014222 122777 000022 164014  CMPB #22, QSR     ; COMPARE SR WITH 22
2631 014230 001401        BEQ  78$           ; IF NO. ERROR SKIP HLT
2632 014232 104000        HLT                 ; CALL ERROR ROUTINE
2633 014234                78$:
2634
2635
2636
2637 014234 104400        SCOPE                ; TEST OF DIVIDE
2638 014236 012777 000000 163772  MOV  #0, QMQ      ; LOAD MQ WITH 0
2639 014244 012777 025253 163762  MOV  #25253, QAC   ; LOAD AC WITH 25253
2640 014252 012777 125252 163752  MOV  #125252, QDIV ; LOAD DIV WITH 125252 AND DIVIDE
2641 014260 022777 000000 163746  CMP  #0, QAC       ; COMPARE AC WITH 0 (REMAINDER)
2642 014266 001401        BEQ  79$           ; IF NO. ERROR SKIP HLT
2643 014270 104000        HLT                 ; CALL ERROR ROUTINE
2644 014272                79$:
2645 014272 022777 100000 163736  CMP  #100000, QMQ  ; COMPARE MQ WITH 100000 (QUOTIANT)
2646 014300 001401        BEQ  80$           ; IF NO. ERROR SKIP HLT
2647 014302 104000        HLT                 ; CALL ERROR ROUTINE
2648 014304                80$:
2649 014304 122777 000320 163732  CMPB #320, QSR    ; COMPARE SR WITH 320
2650 014312 001401        BEQ  81$           ; IF NO. ERROR SKIP HLT
2651 014314 104000        HLT                 ; CALL ERROR ROUTINE
2652 014316                81$:
2653 014316 104400        SCOPE                ; TEST OF DIVIDE
2654 014320 012777 000001 163710  MOV  #1, QMQ      ; LOAD MQ WITH 1
2655 014326 012777 025253 163700  MOV  #25253, QAC   ; LOAD AC WITH 25253
2656 014334 012777 125252 163670  MOV  #125252, QDIV ; LOAD DIV WITH 125252 AND DIVIDE
2657 014342 022777 000001 163664  CMP  #1, QAC       ; COMPARE AC WITH 1 (REMAINDER)
2658 014350 001401        BEQ  82$           ; IF NO. ERROR SKIP HLT
2659 014352 104000        HLT                 ; CALL ERROR ROUTINE
2660 014354                82$:
2661 014354 022777 100000 163654  CMP  #100000, QMQ  ; COMPARE MQ WITH 100000 (QUOTIANT)
2662 014362 001401        BEQ  83$           ; IF NO. ERROR SKIP HLT
2663 014364 104000        HLT                 ; CALL ERROR ROUTINE
2664 014366                83$:
2665 014366 122777 000300 163650  CMPB #300, QSR    ; COMPARE SR WITH 300
2666 014374 001401        BEQ  84$           ; IF NO. ERROR SKIP HLT
2667 014376 104000        HLT                 ; CALL ERROR ROUTINE
2668 014400                84$:
2669
2670
2671
2672 014400 104400        SCOPE                ; TEST OF DIVIDE
    
```



2673	014402	012777	077777	163626	MOV	#77777,DMQ	;LOAD MQ WITH 77777
2674	014410	012777	037777	163616	MOV	#37777,DAC	;LOAD AC WITH 37777
2675	014416	012777	077777	163606	MOV	#77777,ADIV	;LOAD DIV WITH 77777 AND DIVIDE
2676	014424	022777	077776	163602	CMP	#77776,DAC	;COMPARE AC WITH 77776 (REMAINDER)
2677	014432	001401			BEQ	85\$	;IF NO.ERROR SKIP HLT
2678	014434	104000			HLT		;CALL ERROR ROUTINE
2679	014436				85\$:		
2680	014436	022777	077777	163572	CMP	#77777,DMQ	;COMPARE MQ WITH 77777 (QUOTIANT)
2681	014444	001401			BEQ	86\$	;IF NO.ERROR SKIP HLT
2682	014446	104000			HLT		;CALL ERROR ROUTINE
2683	014450				86\$:		
2684	014450	122777	000000	163566	CMPB	#0,DSR	;COMPARE SR WITH 0
2685	014456	001401			BEQ	87\$	;IF NO.ERROR SKIP HLT
2686	014460	104000			HLT		;CALL ERROR ROUTINE
2687	014462				87\$:		
2688							
2689							
2690							
2691	014462	104400			SCOPE		;TEST OF DIVIDE
2692	014464	012777	100000	163544	MOV	#100000,DMQ	;LOAD MQ WITH 100000
2693	014472	012777	000000	163534	MOV	#0,DAC	;LOAD AC WITH 0
2694	014500	012777	000002	163524	MOV	#2,ADIV	;LOAD DIV WITH 2 AND DIVIDE
2695	014506	022777	000000	163520	CMP	#0,DAC	;COMPARE AC WITH 0 (REMAINDER)
2696	014514	001401			BEQ	88\$	;IF NO.ERROR SKIP HLT
2697	014516	104000			HLT		;CALL ERROR ROUTINE
2698	014520				88\$:		
2699	014520	022777	040000	163510	CMP	#40000,DMQ	;COMPARE MQ WITH 40000 (QUOTIANT)
2700	014526	001401			BEQ	89\$	;IF NO.ERROR SKIP HLT
2701	014530	104000			HLT		;CALL ERROR ROUTINE
2702	014532				89\$:		
2703	014532	122777	000022	163504	CMPB	#22,DSR	;COMPARE SR WITH 22
2704	014540	001401			BEQ	90\$	;IF NO.ERROR SKIP HLT
2705	014542	104000			HLT		;CALL ERROR ROUTINE
2706	014544				90\$:		
2707	014544	104400			SCOPE		;TEST OF DIVIDE
2708	014546	012777	100001	163462	MOV	#100001,DMQ	;LOAD MQ WITH 100001
2709	014554	012777	000000	163452	MOV	#0,DAC	;LOAD AC WITH 0
2710	014562	012777	000002	163442	MOV	#2,ADIV	;LOAD DIV WITH 2 AND DIVIDE
2711	014570	022777	000001	163436	CMP	#1,DAC	;COMPARE AC WITH 1 (REMAINDER)
2712	014576	001401			BEQ	91\$	;IF NO.ERROR SKIP HLT
2713	014600	104000			HLT		;CALL ERROR ROUTINE
2714	014602				91\$:		
2715	014602	022777	040000	163426	CMP	#40000,DMQ	;COMPARE MQ WITH 40000 (QUOTIANT)
2716	014610	001401			BEQ	92\$	;IF NO.ERROR SKIP HLT
2717	014612	104000			HLT		;CALL ERROR ROUTINE
2718	014614				92\$:		
2719	014614	122777	000000	163422	CMPB	#0,DSR	;COMPARE SR WITH 0
2720	014622	001401			BEQ	93\$	;IF NO.ERROR SKIP HLT
2721	014624	104000			HLT		;CALL ERROR ROUTINE
2722	014626				93\$:		
2723							
2724							
2725							
2726	014626	104400			SCOPE		;TEST OF DIVIDE
2727	014630	012777	037776	163400	MOV	#37776,DMQ	;LOAD MQ WITH 37776
2728	014636	012777	020000	163370	MOV	#20000,DAC	;LOAD AC WITH 20000

2729	014644	012777	077777	163360	. MOV	#77777, @DIV	; LOAD DIV WITH 77777 AND DIVIDE
2730	014652	022777	077776	163354	. CMP	#77776, @AC	; COMPARE AC WITH 77776 (REMAINDER)
2731	014660	001401			. BEQ	94\$	; IF NO.ERROR SKIP HLT
2732	014662	104000			. HLT		; CALL ERROR ROUTINE
2733	014664						
2734	014664	022777	040000	163344	94\$: CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2735	014672	001401			. BEQ	95\$	; IF NO.ERROR SKIP HLT
2736	014674	104000			. HLT		; CALL ERROR ROUTINE
2737	014676						
2738	014676	122777	000000	163340	95\$: CMPB	#0, @SR	; COMPARE SR WITH 0
2739	014704	001401			. BEQ	96\$	; IF NO.ERROR SKIP HLT
2740	014706	104000			. HLT		; CALL ERROR ROUTINE
2741	014710						
2742							
2743							
2744	014710	104400			SCOPE		; TEST OF DIVIDE
2745	014712	012777	077777	163316	. MOV	#77777, @MQ	; LOAD MQ WITH 77777
2746	014720	012777	177777	163306	. MOV	#177777, @AC	; LOAD AC WITH 177777
2747	014726	012777	177776	163276	. MOV	#177776, @DIV	; LOAD DIV WITH 177776 AND DIVIDE
2748	014734	022777	177777	163272	. CMP	#177777, @AC	; COMPARE AC WITH 177777 (REMAINDER)
2749	014742	001401			. BEQ	97\$	; IF NO.ERROR SKIP HLT
2750	014744	104000			. HLT		; CALL ERROR ROUTINE
2751	014746						
2752	014746	022777	040000	163262	97\$: CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2753	014754	001401			. BEQ	98\$	; IF NO.ERROR SKIP HLT
2754	014756	104000			. HLT		; CALL ERROR ROUTINE
2755	014760						
2756	014760	122777	000040	163256	98\$: CMPB	#40, @SR	; COMPARE SR WITH 40
2757	014766	001401			. BEQ	99\$	; IF NO.ERROR SKIP HLT
2758	014770	104000			. HLT		; CALL ERROR ROUTINE
2759	014772						
2760	014772	104400			SCOPE		; TEST OF DIVIDE
2761	014774	012777	100001	163234	. MOV	#100001, @MQ	; LOAD MQ WITH 100001
2762	015002	012777	157777	163224	. MOV	#157777, @AC	; LOAD AC WITH 157777
2763	015010	012777	100000	163214	. MOV	#100000, @DIV	; LOAD DIV WITH 100000 AND DIVIDE
2764	015016	022777	100001	163210	. CMP	#100001, @AC	; COMPARE AC WITH 100001 (REMAINDER)
2765	015024	001401			. BEQ	100\$	; IF NO.ERROR SKIP HLT
2766	015026	104000			. HLT		; CALL ERROR ROUTINE
2767	015030						
2768	015030	022777	040000	163200	100\$: CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2769	015036	001401			. BEQ	101\$	; IF NO.ERROR SKIP HLT
2770	015040	104000			. HLT		; CALL ERROR ROUTINE
2771	015042						
2772	015042	122777	000000	163174	101\$: CMPB	#0, @SR	; COMPARE SR WITH 0
2773	015050	001401			. BEQ	102\$	; IF NO.ERROR SKIP HLT
2774	015052	104000			. HLT		; CALL ERROR ROUTINE
2775	015054						
2776							
2777							
2778							
2779	015054	104400			SCOPE		; TEST OF DIVIDE
2780	015056	012777	052525	163152	. MOV	#52525, @MQ	; LOAD MQ WITH 52525
2781	015064	012777	000000	163142	. MOV	#0, @AC	; LOAD AC WITH 0
2782	015072	012777	052525	163132	. MOV	#52525, @DIV	; LOAD DIV WITH 52525 AND DIVIDE
2783	015100	022777	000000	163126	. CMP	#0, @AC	; COMPARE AC WITH 0 (REMAINDER)
2784	015106	001401			. BEQ	103\$	; IF NO.ERROR SKIP HLT

2785	015110	104000				HLT		;CALL ERROR ROUTINE
2786	015112			103\$:				
2787	015112	022777	000001	163116		CMP	#1, QMQ	;COMPARE MQ WITH 1 (QUOTIANT)
2788	015120	001401				BEQ	104\$	;IF NO.ERROR SKIP HLT
2789	015122	104000				HLT		;CALL ERROR ROUTINE
2790	015124			104\$:				
2791	015124	122777	000022	163112		CMPB	#22, QSR	;COMPARE SR WITH 22
2792	015132	001401				BEQ	105\$	;IF NO.ERROR SKIP HLT
2793	015134	104000				HLT		;CALL ERROR ROUTINE
2794	015136			105\$:				
2795	015136	104400				SCOPE		;TEST OF DIVIDE
2796	015140	012777	052524	163070		MOV	#52524, QMQ	;LOAD MQ WITH 52524
2797	015146	012777	000000	163060		MOV	#0, QAC	;LOAD AC WITH 0
2798	015154	012777	052525	163050		MOV	#52525, QDIV	;LOAD DIV WITH 52525 AND DIVIDE
2799	015162	022777	052524	163044		CMP	#52524, QAC	;COMPARE AC WITH 52524 (REMAINDER)
2800	015170	001401				BEQ	106\$	;IF NO.ERROR SKIP HLT
2801	015172	104000				HLT		;CALL ERROR ROUTINE
2802	015174			106\$:				
2803	015174	022777	000000	163034		CMP	#0, QMQ	;COMPARE MQ WITH 0 (QUOTIANT)
2804	015202	001401				BEQ	107\$	;IF NO.ERROR SKIP HLT
2805	015204	104000				HLT		;CALL ERROR ROUTINE
2806	015206			107\$:				
2807	015206	122777	000010	163030		CMPB	#10, QSR	;COMPARE SR WITH 10
2808	015214	001401				BEQ	108\$	;IF NO.ERROR SKIP HLT
2809	015216	104000				HLT		;CALL ERROR ROUTINE
2810	015220			108\$:				
2811	015220	104400				SCOPE		;TEST OF DIVIDE
2812	015222	012777	000000	163006		MOV	#0, QMQ	;LOAD MQ WITH 0
2813	015230	012777	000000	162776		MOV	#0, QAC	;LOAD AC WITH 0
2814	015236	012777	125252	162766		MOV	#125252, QDIV	;LOAD DIV WITH 125252 AND DIVIDE
2815	015244	022777	000000	162762		CMP	#0, QAC	;COMPARE AC WITH 0 (REMAINDER)
2816	015252	001401				BEQ	109\$	;IF NO.ERROR SKIP HLT
2817	015254	104000				HLT		;CALL ERROR ROUTINE
2818	015256			109\$:				
2819	015256	022777	000000	162752		CMP	#0, QMQ	;COMPARE MQ WITH 0 (QUOTIANT)
2820	015264	001401				BEQ	110\$	;IF NO.ERROR SKIP HLT
2821	015266	104000				HLT		;CALL ERROR ROUTINE
2822	015270			110\$:				
2823	015270	122777	000036	162746		CMPB	#36, QSR	;COMPARE SR WITH 36
2824	015276	001401				BEQ	111\$	;IF NO.ERROR SKIP HLT
2825	015300	104000				HLT		;CALL ERROR ROUTINE
2826	015302			111\$:				
2827	015302	104400				SCOPE		;TEST OF SUCCESIVE MULTIPLIES
2828	015304	012777	000001	162724		MOV	#1, QMQ	
2829								
2830	015312	012777	000002	162720		MOV	#2, QMUL	
2831	015320	012777	000002	162712		MOV	#2, QMUL	
2832	015326	012777	000002	162704		MOV	#2, QMUL	
2833	015334	012777	000002	162676		MOV	#2, QMUL	
2834	015342	012777	000002	162670		MOV	#2, QMUL	
2835	015350	012777	000002	162662		MOV	#2, QMUL	
2836	015356	012777	000002	162654		MOV	#2, QMUL	
2837	015364	012777	000002	162646		MOV	#2, QMUL	
2838	015372	012777	000002	162640		MOV	#2, QMUL	
2839	015400	012777	000002	162632		MOV	#2, QMUL	
2840	015406	012777	000002	162624		MOV	#2, QMUL	

2841	015414	012777	000002	162616	MOV	#2, 2MUL		
2842	015422	012777	000002	162610	MOV	#2, 2MUL		
2843	015430	012777	000002	162602	MOV	#2, 2MUL		
2844								
2845	015436	022777	040000	162572	CMP	#40000, 2MQ		
2846	015444	001401			BEQ	112\$		; IF NO.ERROR SKIP HLT
2847	015446	104000			HLT			; CALL ERROR ROUTINE
2848	015450						112\$:	
2849	015450	005777	162560		TST	2AC		
2850	015454	001401			BEQ	113\$		; IF NO.ERROR SKIP HLT
2851	015456	104000			HLT			; CALL ERROR ROUTINE
2852	015460						113\$:	
2853	015460	122777	000022	162556	CMPB	#22, 2SR		; CHECK STATUS 22
2854	015466	001401			BEQ	114\$		; IF NO.ERROR SKIP HLT
2855	015470	104000			HLT			; CALL ERROR ROUTINE
2856	015472						114\$:	
2857	015472	104400			SCOPE			; TEST OF SUCCESIVE DIVIDES
2858	015474	012777	040000	162534	MOV	#40000, 2MQ		
2859								
2860	015502	012777	000002	162522	MOV	#2, 2DIV		
2861	015510	012777	000002	162514	MOV	#2, 2DIV		
2862	015516	012777	000002	162506	MOV	#2, 2DIV		
2863	015524	012777	000002	162500	MOV	#2, 2DIV		
2864	015532	012777	000002	162472	MOV	#2, 2DIV		
2865	015540	012777	000002	162464	MOV	#2, 2DIV		
2866	015546	012777	000002	162456	MOV	#2, 2DIV		
2867	015554	012777	000002	162450	MOV	#2, 2DIV		
2868	015562	012777	000002	162442	MOV	#2, 2DIV		
2869	015570	012777	000002	162434	MOV	#2, 2DIV		
2870	015576	012777	000002	162426	MOV	#2, 2DIV		
2871	015604	012777	000002	162420	MOV	#2, 2DIV		
2872	015612	012777	000002	162412	MOV	#2, 2DIV		
2873	015620	012777	000002	162404	MOV	#2, 2DIV		
2874								
2875	015626	005777	162402		TST	2AC		
2876	015632	001401			BEQ	115\$		; IF NO.ERROR SKIP HLT
2877	015634	104000			HLT			; CALL ERROR ROUTINE
2878	015636						115\$:	
2879	015636	022777	000001	162372	CMP	#1, 2MQ		
2880	015644	001401			BEQ	116\$		; IF NO.ERROR SKIP HLT
2881	015646	104000			HLT			; CALL ERROR ROUTINE
2882	015650						116\$:	
2883	015650	122777	000022	162366	CMPB	#22, 2SR		; CHECK STATUS 22
2884	015656	001401			BEQ	117\$		; IF NO.ERROR SKIP HLT
2885	015660	104000			HLT			; CALL ERROR ROUTINE
2886	015662						117\$:	
2887	015662	104400			SCOPE			; TEST OR ALTERNATE MUL AND DIV
2888	015664	012777	052525	162344	MOV	#52525, 2MQ		
2889								
2890	015672	012777	040000	162340	MOV	#40000, 2MUL		
2891	015700	012777	040000	162324	MOV	#40000, 2DIV		
2892	015706	012777	040000	162324	MOV	#40000, 2MUL		
2893	015714	012777	040000	162310	MOV	#40000, 2DIV		
2894	015722	012777	040000	162310	MOV	#40000, 2MUL		
2895	015730	012777	040000	162274	MOV	#40000, 2DIV		
2896	015736	012777	040000	162274	MOV	#40000, 2MUL		

MAINDEC-11-DZKEB-A  
DZKEBA.P11

MACY11 27(732) 03-NOV-76 15:27 PAGE 61

2897	015744	012777	040000	162260	MOV	#40000, @DIV	
2898	015752	012777	040000	162260	MOV	#40000, @MUL	
2899	015760	012777	040000	162244	MOV	#40000, @DIV	
2900							
2901	015766	022777	052525	162242	CMP	#52525, @MQ	
2902	015774	001401			BEQ	118\$	; IF NO.ERROR SKIP HLT
2903	015776	104000			HLT		; CALL ERROR ROUTINE
2904	016000						
2905	016000	005777	162230		TST	@AC	
2906	016004	001401			BEQ	119\$	; IF NO.ERROR SKIP HLT
2907	016006	104000			HLT		; CALL ERROR ROUTINE
2908	016010						
2909	016010	122777	000022	162226	CMPB	#22, @SR	; CHECK STATUS 22
2910	016016	001401			BEQ	120\$	; IF NO.ERROR SKIP HLT
2911	016020	104000			HLT		; CALL ERROR ROUTINE
2912	016022						
2913	016022	104400			SCOPE		; TEST OF FAST PROCESSING OF DATA
2914	016024	016700	162206		MOV	MQ, %0	; SET UP POINTER
2915	016030	012720	125252		MOV	#125252, (0)+	; LOAD MQ
2916	016034	012710	040000		MOV	#40000, (0)	; LOAD MUL
2917	016040	014001			MOV	-(0), %1	; SAVE MQ
2918	016042	014002			MOV	-(0), %2	; SAVE AC
2919	016044	005720			TST	(0)+	
2920	016046	020127	100000		CMP	%1, #100000	; CHECK MQ
2921	016052	001401			BEQ	121\$	; IF NO.ERROR SKIP HLT
2922	016054	104000			HLT		; CALL ERROR ROUTINE
2923	016056						
2924	016056	020227	165252		CMP	%2, #165252	; CHECK AC
2925	016062	001401			BEQ	122\$	; IF NO.ERROR SKIP HLT
2926	016064	104000			HLT		; CALL ERROR ROUTINE
2927	016066						
2928							
2929	016066	104400			SCOPE		; SAVE WITH DIVIDE
2930	016070	016700	162142		MOV	MQ, %0	
2931	016074	012710	000001		MOV	#1, (0)	; LOAD MQ WITH 1
2932	016100	012740	025253		MOV	#25253, -(0)	; LOAD AC WITH 25253
2933	016104	012740	125252		MOV	#125252, -(0)	; DIVIDE
2934	016110	005720			TST	(0)+	
2935	016112	012001			MOV	(0)+, %1	; SAVE THE AC IN R1
2936	016114	011002			MOV	(0), %2	; SAVE THE MQ IN R2
2937	016116	020127	000001		CMP	%1, #1	; TEST THE AC
2938	016122	001401			BEQ	123\$	; IF NO.ERROR SKIP HLT
2939	016124	104000			HLT		; CALL ERROR ROUTINE
2940	016126						
2941	016126	020227	100000		CMP	%2, #100000	; TEST THE MQ
2942	016132	001401			BEQ	124\$	; IF NO.ERROR SKIP HLT
2943	016134	104000			HLT		; CALL ERROR ROUTINE
2944	016136						
2945	016136	104400			SCOPE		
2946							

```

2947 016140 .DEV:
2948 ;*****
2949 ; AT THIS POINT MULT AND DIV ARE OK
2950 ;*****
2951 ;*****
2952 ; BELL ON PASS COMPLETE
2953 ;*****
2954 ;*****
2955 016140 004767 001204 JSR %7,CKSWR ;CHECK FOR CONT-G
2956 016144 032777 002000 161666 BIT #2000,DSWR
2957 016152 001012 BNE TRTRAP
2958 016154 012737 000207 177566 MOV #207,D#177566
2959 016162 105777 000444 64$: TSTB DTPS ;IS TTY READY FOR NEXT CHARACTER?
2960 016166 100375 BPL 64$ ;IF READY BIT (BIT 7)=0--NO, LOOP
2961 016170 012705 017757 MOV #SEAE,%5 ;PRINT EAE OK
2962 016174 004767 001652 JSR %7,TTOUT
2963 ;*****
2964 ;*****
2965 ; ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
2966 ;*****
2967 016200 004767 001144 TRTRAP: JSR %7,CKSWR ;CHECK FOR CONT-G
2968 016204 032777 010000 161626 BIT #10000,DSWR ;SHOULD WE RUN WITH TRACE TRAP
2969 016212 001417 BEQ YESTR ;YES
2970 016214 005767 000104 TST YESTR1 ;NO HAVE WE RAN WITH TRACE TRAP ON
2971 016220 001411 BEQ TRPA ;IF SO RESTORE PREVIOUS CONTENTS
2972 016222 016767 000076 161564 MOV YESTR1,14
2973 016230 016767 000072 161560 MOV YESTR2,16
2974 016236 042767 000020 161532 BIC #20,PSW
2975 016244 000167 162070 TRPA: JMP BEGIN ;CLEAR TRACE TRAP
2976 016250 000000 TRPB: 0 ;START OF TEST WITH TRACE OFF
2977 ;*****
2978 ;*****
2979 016252 016767 161536 000044 YESTR: MOV OLD CONTENTS, SET UP FOR TRACE TRAP
2980 016260 016767 161532 000040 MOV 14,YESTR1 ;SAVE ODT PC
2981 016266 012767 016330 161520 MOV 16,YESTR2 ;SAVE ODT STATUS
2982 016274 005067 161516 MOV #YESRT,14 ;NEW TRAP VECTOR
2983 016300 005067 161472 CLR 16 ;NEW CONDITION CODES
2984 016304 005167 177740 CLR PSW
2985 016310 100403 COM TRPB
2986 016312 052767 000020 161456 BMI 1$
2987 016320 1$: BIS #20,PSW ;SET TRACE TRAP
2988 016320 000167 162014 JMP BEGIN ;START OF TEST WITH TRACE ON
2989 ;*****
2990 016324 000000 YESTR1: 0 ;STORAGE FOR ODT PC
2991 016326 000000 YESTR2: 0 ;STORAGE FOR ODT STATUS
2992 016330 000002 YESRT: RTI ;RETURN TO PROGRAM FROM TRAP
2993 016332 000000 HALT ;RTI FAILED
2994 ;*****
2995 ;*****
2996 016334 004767 001010 PRINT: JSR %7,CKSWR ;CHECK FOR CONT-G
2997 016340 032777 002000 161472 BIT #2000,DSWR
2998 016346 001406 BEQ PRNT
2999 016350 012737 000007 177566 MOV #7,D#177566
3000 016356 105777 000250 64$: TSTB DTPS ;IS TTY READY FOR NEXT CHARACTER?
3001 016362 100375 BPL 64$ ;IF READY BIT (BIT 7)=0--NO, LOOP
3002 016364 004767 000760 PRNT: JSR %7,CKSWR ;CHECK FOR CONT-G

```

3003	016370	037727	161444	020000		BIT	QSWR, #20000		;TEST FOR INHIBIT PRINT OUT
3004	016376	001401				BEQ	1\$		
3005	016400	000002				RTI			
3006	016402	012667	000226		1\$:	MOV	(6)+, SAVPC		;PC OF FAILING ROUTINE
3007	016406	012667	000224			MOV	(6)+, SAVCC		;CC OF ERROR CONDITION
3008	016412	024646				CMP	-(6), -(6)		;REPOSITION THE STACK
3009	016414	012777	000215	000206		MOV	#215, ATPB		;CR
3010	016422	105777	000204		64\$:	TSTB	ATPS		; IS TTY READY FOR NEXT CHARACTER?
3011	016426	100375				BPL	64\$		; IF READY BIT (BIT 7)=0--NO, LOOP
3012	016430	012777	000212	000172		MOV	#212, ATPB		;LINE FEED
3013	016436	105777	000170		65\$:	TSTB	ATPS		; IS TTY READY FOR NEXT CHARACTER?
3014	016442	100375				BPL	65\$		; IF READY BIT (BIT 7)=0--NO, LOOP
3015	016444	010267	000152			MOV	%2, SAVR2		;SAVE R2
3016	016450	010367	000150			MOV	%3, SAVR3		;SAVE R3
3017	016454	010467	000146			MOV	%4, SAVR4		;SAVE R4
3018	016460	016702	000150			MOV	SAVPC, %2		
3019	016464	004767	000154			JSR	%7, PRTAB		;PRINT OCTAL NUMBER
3020	016470	012777	000240	000132		MOV	#240, ATPB		
3021	016476	105777	000130		66\$:	TSTB	ATPS		; IS TTY READY FOR NEXT CHARACTER?
3022	016502	100375				BPL	66\$		; IF READY BIT (BIT 7)=0--NO, LOOP
3023	016504	017702	161524			MOV	QAC, %2		
3024	016510	004767	000130			JSR	%7, PRTAB		;PRINT OCTAL NUMBER
3025	016514	012777	000240	000106		MOV	#240, ATPB		
3026	016522	105777	000104		67\$:	TSTB	ATPS		; IS TTY READY FOR NEXT CHARACTER?
3027	016526	100375				BPL	67\$		; IF READY BIT (BIT 7)=0--NO, LOOP
3028	016530	017702	161502			MOV	QMQ, %2		
3029	016534	004767	000104			JSR	%7, PRTAB		;PRINT OCTAL NUMBER
3030	016540	012777	000240	000062		MOV	#240, ATPB		
3031	016546	105777	000060		68\$:	TSTB	ATPS		; IS TTY READY FOR NEXT CHARACTER?
3032	016552	100375				BPL	68\$		; IF READY BIT (BIT 7)=0--NO, LOOP
3033	016554	017702	161462			MOV	QSC, %2		
3034	016560	004767	000060			JSR	%7, PRTAB		;PRINT OCTAL NUMBER
3035	016564	016702	000032			MOV	SAVR2, %2		;RESTORE REGISTERS
3036	016570	016703	000030			MOV	SAVR3, %3		
3037	016574	016704	000026			MOV	SAVR4, %4		
3038	016600	004767	000544			JSR	%7, CKSWR		;CHECK FOR CONT-G
3039	016604	005777	161230			TST	QSWR		;CHECK SR FOR HALT SWITCH
3040	016610	100003				BPL	2\$		
3041	016612	000000				HALT			; HALT ON ERROR
3042									; GET SWITCH AFTER HALT
3043	016614	004767	000530		2\$:	JSR	%7, CKSWR		;CHECK FOR CONT-G
3044	016620								
3045	016620	000002				RTI			;RETURN TO MAINLINE
3046	016622	000000			SAVR2:	0			
3047	016624	000000			SAVR3:	0			
3048	016626	000000			SAVR4:	0			
3049	016630	177566			TPB:	177566			;DATA
3050	016632	177564			TPS:	177564			;STATUS
3051	016634	000000			SAVPC:	0			
3052	016636	000000			SAVCC:	0			
3053	016640	177560			TKS:	177560			
3054	016642	177562			TKB:	177562			
3055	016644	005067	000260		PRTAB:	CLR	BINCT		
3056	016650	005067	000252			CLR	WGTCT		
3057	016654	012704	017134			MOV	#LIST, %4		;GET LIST ADDRESS
3058	016660	142777	000177	177744		BICB	#177, ATPS		;CLR INT FLAG

3059	016666	012767	000005	000236		MOV	#5, ASCNT	
3060	016674	012767	000007	000220		MOV	#7, SEVEN	
3061	016702	012767	000001	000214		MOV	#1, DECML	
3062	016710				WAIT1:			
3063	016710	105777	177716		64\$:	TSTB	@TPS	; IS TTY READY FOR NEXT CHARACTER?
3064	016714	100375				BPL	64\$	; IF READY BIT (BIT 7)=0--NO, LOOP
3065	016716	005702				TST	%2	
3066	016720	100404				BMI	MINUS	; NEG SIGN PRINT 1
3067	016722	012777	000260	177700		MOV	#260, @TPB	; POS SIGN PRINT 0
3068	016730	000403				BR	START	
3069	016732	012777	000261	177670	MINUS:	MOV	#261, @TPB	
3070	016740	016703	000156		START:	MOV	SEVEN, %3	; PUT MASK IN R3
3071	016744	010267	000150			MOV	%2, TODDLE	; GET READY TO DOODLE NUMBER IN TODDLE
3072	016750	005167	000144			COM	TODDLE	; COMPENSATES FOR COMPLEMENT DURING BIC
3073	016754	046703	000140			BIC	TODDLE, %3	; AND IN OCTAL CHARACTER
3074	016760	001410				BEQ	WRTOC	; ZERO, WRITE 0 IN LIST
3075	016762	066767	000136	000136	MKNUM:	ADD	DECML, WGTCT	; COUNT UP TO
3076	016770	005267	000134			INC	BINCT	; AND RECORD
3077	016774	026703	000126			CMP	WGTCT, %3	; SAME BINARY WEIGHT
3078	017000	001370				BNE	MKNUM	; KEEP COUNTN
3079	017002	062767	000260	000120	WRTOC:	ADD	#260, BINCT	; ADD ASCII PREFIX
3080	017010	016724	000114			MOV	BINCT, (4)+	; WRITE ASCII CHAR IN LIST
3081	017014	066767	000102	000102		ADD	SEVEN, DECML	; EXPAND BINARY WEIGHT
3082	017022	005067	000100			CLR	WGTCT	
3083	017026	005067	000076			CLR	BINCT	
3084	017032	005367	000074			DEC	ASCNT	
3085	017036	001410				BEQ	XLIST	; 5 CHAR IN LIST
3086	017040	012703	000003			MOV	#3, %3	; SET X3 FOR ADD LOOP
3087	017044	066767	000052	000050	MOADD:	ADD	SEVEN, SEVEN	; MAKING SEVENTY BY SEVEN
3088	017052	005303				DEC	%3	
3089	017054	001373				BNE	MOADD	
3090	017056	000730				BR	START	; NX SEVEN SET GET NX OCTAL
3091	017060	012767	000005	000044	XLIST:	MOV	#5, ASCNT	; SEND 5 CHAR TO TTY
3092	017066				WAIT2:			
3093	017066	105777	177540		64\$:	TSTB	@TPS	; IS TTY READY FOR NEXT CHARACTER?
3094	017072	100375				BPL	64\$	; IF READY BIT (BIT 7)=0--NO, LOOP
3095	017074	014477	177530			MOV	-(4), @TPB	
3096	017100	005367	000026			DEC	ASCNT	
3097	017104	001401				BEQ	HDFHM	
3098	017106	000767				BR	WAIT2	
3099	017110				HDFHM:			
3100	017110	105777	177516		64\$:	TSTB	@TPS	; IS TTY READY FOR NEXT CHARACTER?
3101	017114	100375				BPL	64\$	; IF READY BIT (BIT 7)=0--NO, LOOP
3102	017116	000207				RTS	%7	; HEAD FOR HOME
3103	017120	000000			TODDLE:	0		
3104	017122	000000			SEVEN:	0		
3105	017124	000000			DECML:	0		
3106	017126	000000			WGTCT:	0		
3107	017130	000000			BINCT:	0		
3108	017132	000000			ASCNT:	0		
3109	017134	000000			LIST:	0		
3110	017136	000000				0		
3111	017140	000000				0		
3112	017142	000000				0		
3113	017144	000000				0		



```

3114
3115
3116
3117
3118
3119 017146
3120 017146 004767 000176
3121 017152 032777 040000 160660
3122 017160 001003
3123 017162 011667 000076
3124 017166 000002
3125 017170 022606
3126 017172 012667 160600
3127 017176 000177 000062
3128
3129 017202 004767 000142
3130 017206 032777 040000 160624
3131
3132 017214 001365
3133 017216 004767 000126
3134 017222 032777 004000 160610
3135 017230 001007
3136 017232 026727 000024 004000
3137 017240 001403
3138 017242 005267 000014
3139 017246 000750
3140 017250 005067 000006
3141 017254 011667 000004
3142 017260 000002
3143 017262 000000
3144 017264 000340
3145 017266 000
3146 017270 000167 160704
3147 017274 000040
3148 017334 000000
3149 017336 000000
3150 017336 000000
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161 017340 000000
3162 017342 000000
3163 017344 000000
3164 017346 000000
3165

```

```

;*****
; SCOPE LOOP
; ENTERED BY USER TRAP
;*****
SCOPEA:
JSR %7,CKSWR ;CHECK FOR CONT-G
BIT #40000,@SWR
BNE SCOPEB ;SCOPE BIT IS A ONE
MOV @%6,RETURN ;NO - SAVE %7 FOR NEXT TIME
RTI ;RETURN IN SEQUENCE
SCOPEB: CMP (6)+,%6 ;REPOSITION THE STACK
MOV (6)+,PSW
JMP @RETURN ;SCOPE RETURN
SCOPEC: JSR %7,CKSWR ;CHECK FOR CONT-G
BIT #40000,@SWR ;TEST SR FOR SCOPE
BNE SCOPEB ;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 400
JSR %7,CKSWR ;YES SCOPE
BIT #4000,@SWR ;CHECK FOR CONT-G
BNE SCOPEG ;NO - TEST FOR ITERATION
CMP SCOPEF,#4000 ;INHIBIT ITERATION
BEQ SCOPEG ;EXIT - DONE
INC SCOPEF ;INCREMENT COUNT
BR SCOPEB ;LOOP SOME MORE
SCOPEG: CLR SCOPEF ;CLEAR COUNT
MOV @%6,RETURN ;SAVE SCOPE RETURN POINTER
RTI ;RETURN INLINE-NEXT TEST
SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
RETURN: BEGIN ;ADDRESS OF LAST TEST
EOMK: .BYTE 0
.EVEN
JMP 200
.BLK 40
BUFF: 0 ;FOR STACK POINTER 40 LOCATIONS
CP: 0
;*****
; CHECK SWITCH REGISTER ROUTINE. CHECKS FOR IG TO ALLOW CHANGING
; OF LOC.176.
; LOCATIONS USED:
;*****
TEMPST: .WORD 0
COUNT: .WORD 0
RDSW: .WORD 0
TIB: .WORD 0

```

3166	017350				CKSWR:					
3167	017350	022767	000176	160462		CMP	#SWREG, SWR			; SOFTWARE SWITCH REGISTER PRESENT
3168	017356	001131				BNE	OUT			; NO, GET OUT
3169	017360	105777	177254			TSTB	@TKS			; YES, WAIT FOR
3170	017364	100126				BPL	OUT			; READY, GET CHARACTER
3171	017366	017767	177250	177752		MOV	@TKB, TIB			; AND STRIP OFF
3172	017374	042767	177600	177744		BIC	#177600, TIB			; THE GARBAGE
3173	017402	022767	000007	177736		CMP	#7, TIB			; IS IT A <IG>
3174	017410	001114				BNE	OUT			
3175	017412	012705	017716			MOV	#SCNTG, %5			
3176	017416	004767	000430			JSR	%7, TTOUT			
3177	017422	012705	017730		CNTLU:	MOV	#MSWR, %5			
3178	017426	004767	000420			JSR	%7, TTOUT			
3179	017432	017702	160402			MOV	@SWR, R2			
3180	017436	004767	177202			JSR	%7, PRTAB			
3181	017442	012705	017740			MOV	#MNEW, %5			
3182	017446	004767	000400			JSR	%7, TTOUT			
3183	017452	005067	177662		\$READ:	CLR	TEMPST			
3184	017456	012767	000007	177656		MOV	#7, COUNT			
3185	017464	004767	000154		1\$:	JSR	%7, TTIN			; GO READ A CHARACTER
3186	017470	042767	177600	177650		BIC	#177600, TIB			; STRIP OFF GARBAGE
3187	017476	122767	000025	177642		CMPB	#25, TIB			; IS IT A ↑U?
3188	017504	001001				BNE	2\$			; BRANCH IF NOT
3189	017506	000745			3\$:	BR	CNTLU			; START OVER
3190	017510	122767	000015	177630	2\$:	CMPB	#15, TIB			; IS IT A <CR>?
3191	017516	001011				BNE	4\$			; BRANCH IF NOT
3192	017520	012705	017724			MOV	#SCRLF, %5			
3193	017524	004767	000322			JSR	%7, TTOUT			
3194	017530	022767	000007	177604		CMP	#7, COUNT			; WAS IT FIRST CHARACTER
3195	017536	001036				BNE	7\$			; CHANGE SWR IF NOT FIRST ONE
3196	017540	000440			8\$:	BR	OUT			; GET OUT
3197	017542	122767	000060	177576	4\$:	CMPB	#60, TIB			
3198	017550	003004				BGT	5\$			
3199	017552	122767	000067	177566		CMPB	#67, TIB			
3200	017560	002005				BGE	6\$			
3201	017562	012705	017751		5\$:	MOV	#SQUEST, %5			
3202	017566	004767	000260			JSR	%7, TTOUT			
3203	017572	000745				BR	3\$			; START OVER IF NOT LEGAL CHARACTER
3204	017574	006367	177540		6\$:	ASL	TEMPST			
3205	017600	006367	177534			ASL	TEMPST			
3206	017604	006367	177530			ASL	TEMPST			
3207	017610	142767	000060	177530		BICB	#60, TIB			; GET NITTY-GRITTY
3208	017616	156767	177524	177514		BISB	TIB, TEMPST			
3209	017624	005367	177512			DEC	COUNT			; ONLY WANT 6 DIGITS
3210	017630	001754				BEQ	5\$			
3211	017632	000714				BR	1\$			
3212	017634	016777	177500	160176	7\$:	MOV	TEMPST, @SWR			; CHANGE SWITCH REGISTER CONTENTS
3213	017642	000207			OUT:	RTS	%7			; RETURN TO PROGRAM

3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227  
3228  
3229  
3230  
3231  
3232  
3233  
3234  
3235  
3236  
3237  
3238  
3239  
3240  
3241  
3242  
3243  
3244  
3245  
3246  
3247  
3248  
3249  
3250  
3251  
3252  
3253  
3254  
3255  
3256  
3257  
3258  
3259

017644  
017644 005077 176770  
017650 005077 176766  
017654 005067 177466  
017660 005277 176754  
017664 105777 176750  
017670 100375  
017672 017767 176744 177446  
017700  
017700 105777 176726  
017704 100375  
017706 116777 177434 176714  
017714 000207  
017716 057137 020107 000046  
017724 020137 000046  
017730 051537 051127 020075  
017736 000046  
017740 020040 042516 036527  
017746 023040 000  
017751 137 020077 023137  
017756 000  
017757 137 040505 020105  
017764 045517 020040 020137  
017772 000046  
017774 020137 040515 047111  
020002 042504 026503 030461  
020010 042055 045532 041105  
020016 040455 023040 000  
020023 137 045440 030505  
020030 020061 047514 044507  
020036 020103 042524 052123  
020044 023040 000  
020050 020050  
020050 000000

\*\*\*\*\*  
TTY READ SUBROUTINE\*\*\*\*\*  
\*\*\*\*\*

TTIN:  
CLR @TKS  
CLR @TKB  
CLR TIB  
INC @TKS  
TTIN1: TSTB @TKS  
BPL TTIN1  
MOV @TKB, TIB  
TTIN2:  
645: TSTB @TPS  
BPL 645  
MOVB TIB, @TPB  
RTS %7  
\$CNTG: .ASCIZ '+IG &'  
\$CRLF: .ASCIZ '+ &'  
\$MSWR: .ASCIZ '+SWR= &'  
\$MNEW: .ASCIZ ' NEW= &'  
\$QUEST: .ASCIZ '+? +&'  
\$EAE: .ASCIZ '+EAE OK + &'  
\$MAIN: .ASCIZ '+ MAINDEC-11-DZKEB-A &'  
\$HEAD: .ASCIZ '+ KE11 LOGIC TEST &'  
.EVEN  
OFL: 0

;IS TTY READY FOR NEXT CHARACTER?  
;IF READY BIT (BIT 7)=0--NO, LOOP

;FIRST CHAR FLAG

3260  
3261  
3262  
3263  
3264  
3265  
3266  
3267  
3268  
3269  
3270  
3271  
3272  
3273  
3274  
3275  
3276  
3277  
3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286  
3287  
3288  
3289  
3290  
3291  
3292

020052  
020052 105715  
020054 001403  
020056 122715 000046  
020062 001005  
020064 042777 000100 176540  
020072 005005  
020074 000207  
020076 122715 000137  
020102 001411  
020104 122715 000041  
020110 001414  
020112  
020112 105777 176514  
020116 100375  
020120 112577 176504  
020124 000752  
020126 005205  
020130 010567 000020  
020134 012705 020150  
020140 000767  
020142 016705 000006  
020146 000741  
020150 015 012 041  
020154 000000  
000001

\*\*\*\*\*  
TTY ASCII OUTPUT ROUTINE  
\*\*\*\*\*

TTOUT:  
TSTB (5) ;CHECK FOR NULL CHARACTER  
BEQ 1\$ ;IF NOT, TYPE THE CHARACTER  
CMPB #'8,(5) ;CHECK FOR TERMINATOR  
BNE .EMPTY  
BIC @100,@TPS  
CLR %5 ;CLEAR POINTER TO CHARACTER  
RTS %7 ;RETURN  
;CRLF CHAR?  
.EMPTY: CMPB #'+(5)  
BEQ .RET  
CMPB #'!(5) ;CHECK FOR RETURN TERMINATOR  
BEQ .REST  
1\$:  
64\$: TSTB @TPS ;IS TTY READY FOR NEXT CHARACTER?  
BPL 64\$ ;IF READY BIT (BIT 7)=0--NO, LOOP  
.1: MOVB (5)+,@TPB ;TYPE CHARACTER  
BR TTOUT  
.RET: INC %5  
MOV %5, SAV ;SET UP NEW POINTER  
BR .1  
.REST: MOV .SAV,%5  
BR TTOUT  
.RETR: .BYTE 15,12,'!  
;EVEN  
.SAV: 0  
.END











B.4	290#	385	394	402	407	414	419	425	430	437	441	451	462	468	474
	478	484	488	494	499	503	509	515	541	545	561	565	580	584	592
	596	604	608	617	621	629	633	643	647	656	660	668	672	680	684
	693	697	706	710	719	723	731	735	744	748	757	761	770	774	786
	790	797	801	810	814	823	827	834	838	847	851	859	863	872	876
	883	887	896	900	909	913	921	925	932	936	945	949	958	962	971
	975	988	992	1001	1005	1013	1017	1025	1029	1036	1040	1047	1051	1059	1063
	1071	1075	1082	1086	1094	1098	1106	1110	1118	1122	1129	1133	1141	1145	1153
	1157	1165	1169	1181	1185	1193	1197	1206	1210	1219	1223	1230	1234	1243	1247
	1256	1260	1269	1273	1281	1285	1293	1297	1306	1310	1319	1323	1331	1335	1344
	1348	1356	1360	1373	1377	1381	1392	1396	1400	1409	1413	1417	1425	1429	1433
	1444	1448	1452	1462	1466	1470	1479	1483	1487	1497	1501	1505	1515	1519	1523
	1532	1536	1541	1552	1556	1560	1570	1574	1578	1586	1590	1594	1602	1606	1610
	1618	1622	1626	1634	1638	1642	1652	1656	1660	1670	1674	1678	1686	1690	1694
	1708	1712	1716	1726	1730	1734	1742	1746	1750	1761	1765	1769	1780	1784	1788
	1797	1801	1805	1814	1818	1822	1832	1836	1840	1848	1852	1856	1866	1871	1875
	1885	1889	1893	1901	1905	1909	1919	1923	1927	1937	1941	1945	1953	1957	1961
	1971	1975	1974	1989	1993	1997	2018	2022	2026	2037	2041	2045	2055	2059	2063
	2071	2075	2079	2090	2094	2098	2112	2116	2120	2128	2132	2136	2145	2149	2153
	2162	2166	2170	2178	2182	2186	2195	2199	2203	2211	2215	2219	2228	2232	2236
	2252	2256	2261	2271	2275	2280	2288	2292	2297	2308	2312	2317	2327	2331	2336
	2345	2349	2354	2365	2369	2374	2409	2413	2417	2426	2430	2434	2444	2448	2452
	2459	2463	2467	2477	2481	2485	2494	2498	2502	2550	2554	2558	2569	2573	2577
	2588	2592	2596	2604	2608	2612	2623	2627	2631	2642	2646	2650	2658	2662	2666
	2677	2681	2685	2696	2700	2704	2712	2716	2720	2731	2735	2739	2749	2753	2757
	2765	2769	2773	2784	2788	2792	2800	2804	2808	2816	2820	2824	2846	2850	2854
	2876	2880	2884	2902	2906	2910	2921	2925	2938	2942					

COMMEN 1#  
 ENDCOM 1#  
 ESCAPE 1#  
 GETPRI 1#  
 GETSMR 1#  
 MULT 1#  
 NEWTST 1#  
 POP 1#  
 PUSH 1#  
 REPORT 1#  
 SETPRI 1#  
 SETUP 1#  
 SKIP 1#  
 SLASH 1#  
 STARS 1#

	290#	291	305	320	357	359	378	380	519	522	528	531	552	554	
	571	573	777	779	978	980	1172	1174	1363	1365	1697	1699	2001	2003	2005
	2009	2103	2105	2239	2241	2242	2244	2380	2382	2389	2392	2399	2401	2505	2507
	2948	2950	2951	2953	2964	2966	3114	3117	3154	3158	3217	3219	3260	3262	

SMRSU 1#  
 TYPBIN 1#  
 TYPDEC 1#  
 TYPNAM 1#  
 TYPNUM 1#  
 TYPOCS 1#  
 TYPOCT 1#  
 TYPTXT 1#  
 WATTPS 290#  
 \$\$ESCA 1#  
 \$\$NEWT 1#

	2959	3000	3010	3013	3021	3026	3031	3063	3093	3100	3232	3278			
--	------	------	------	------	------	------	------	------	------	------	------	------	--	--	--

SSSKIP 1#  
.EQUAT 1#  
.HEADE 1# 290# 293  
.KT11 1#  
.SETUP 1#  
.SWRHI 1#  
.SACT1 1#  
.SAPT8 1#  
.SAPTH 1#  
.SAPTY 1#  
.SASTA 1#  
.SCATC 1#  
.SCMTA 1#  
.SDB2D 1#  
.SDB20 1#  
.SDIV 1#  
.SEOP 1#  
.SERRO 1#  
.SERRT 1#  
.SMULT 1#  
.SPOWE 1#  
.SRAND 1#  
.SRDEE 1#  
.SRDOC 1#  
.SREAD 1#  
.SR2AZ 1#  
.SSAVE 1#  
.SSB2D 1#  
.SSB20 1#  
.SSCOP 1#  
.SSIZE 1#  
.SSLPR 1#  
.STRAP 1#  
.STYPB 1#  
.STYPD 1#  
.STYPE 1#  
.STYPO 1#  
.S4OCA 1#  
.1170 1#



	1589	1601	1605	1617	1621	1633	1637	1651	1655	1669	1673	1685	1689	1707	1711
	1725	1729	1741	1745	1760	1764	1779	1783	1796	1800	1813	1817	1831	1835	1847
	1851	1865	1870	1884	1888	1900	1904	1918	1922	1936	1940	1952	1956	1970	1974
	1988	1992	2017	2021	2036	2040	2054	2058	2070	2074	2089	2093	2111	2115	2127
	2131	2144	2148	2161	2165	2177	2181	2194	2198	2210	2214	2227	2231	2251	2255
	2259	2270	2274	2278	2287	2291	2295	2307	2311	2315	2326	2330	2334	2344	2348
	2352	2364	2368	2372	2408	2412	2425	2429	2443	2447	2458	2462	2476	2480	2493
	2497	2549	2553	2568	2572	2587	2591	2603	2607	2622	2626	2641	2645	2657	2661
	2676	2680	2695	2699	2711	2715	2730	2734	2748	2752	2764	2768	2783	2787	2799
	2803	2815	2819	2845	2879	2901	2920	2924	2937	2941	3008	3077	3125	3136	3167
	3173	3194													
CMPB	406	418	429	440	544	564	583	595	607	620	632	646	659	671	683
	696	709	722	734	747	760	773	789	800	813	826	837	850	862	875
	886	899	912	924	935	948	961	974	991	1012	1035	1058	1081	1105	1128
	1152	1184	1192	1205	1218	1229	1242	1255	1268	1280	1292	1305	1318	1330	1343
	1355	1380	1399	1416	1432	1451	1469	1486	1504	1522	1540	1559	1577	1593	1609
	1625	1641	1659	1677	1693	1715	1733	1749	1768	1787	1804	1821	1839	1855	1874
	1892	1908	1926	1944	1960	1978	1996	2025	2044	2062	2078	2097	2119	2135	2152
	2169	2185	2202	2218	2235	2416	2433	2451	2466	2484	2501	2557	2576	2595	2611
	2630	2649	2665	2684	2703	2719	2738	2756	2772	2791	2807	2823	2853	2883	2909
	3187	3190	3197	3199	3268	3273	3275								
	2984	3072													
COM	557	3084	3088	3096	3209										
DEC	330														
ENT	324	2993	3041												
HALT	447	537	3076	3138	3227	3282									
INC	344	526	2398	2975	2988	3127	3147								
JMP	341	343	373	523	2395	2955	2962	2967	2996	3002	3019	3024	3029	3034	3038
JSR	3043	3120	3129	3133	3176	3178	3180	3182	3185	3193	3202				
	338	340	342	361	362	363	368	369	370	374	383	390	391	400	412
MOV	423	435	449	454	458	460	492	507	536	539	556	559	576	588	590
	600	602	613	615	625	627	638	640	652	654	664	666	676	678	689
	691	702	704	715	717	727	729	740	742	753	755	766	768	783	794
	795	807	808	820	821	831	832	844	845	856	857	869	870	880	881
	893	894	906	907	918	919	929	930	942	943	955	956	968	969	984
	998	999	1010	1011	1022	1023	1033	1034	1044	1045	1056	1057	1068	1069	1079
	1080	1091	1092	1103	1104	1115	1116	1126	1127	1138	1139	1150	1151	1162	1163
	1178	1190	1191	1203	1204	1216	1217	1227	1228	1240	1241	1253	1254	1266	1267
	1278	1279	1290	1291	1303	1304	1316	1317	1328	1329	1341	1342	1353	1354	1369
	1370	1371	1388	1389	1390	1405	1406	1407	1421	1422	1423	1440	1441	1442	1458
	1459	1460	1475	1476	1477	1493	1494	1495	1511	1512	1513	1528	1529	1530	1548
	1549	1550	1566	1567	1568	1582	1583	1584	1598	1599	1600	1614	1615	1616	1630
	1631	1632	1648	1649	1650	1666	1667	1668	1682	1683	1684	1704	1705	1706	1722
	1723	1724	1738	1739	1740	1757	1758	1759	1776	1777	1778	1793	1794	1795	1810
	1811	1812	1828	1829	1830	1844	1845	1846	1862	1863	1864	1881	1882	1883	1897
	1898	1899	1915	1916	1917	1933	1934	1935	1949	1950	1951	1967	1968	1969	1985
	1986	1987	2014	2015	2016	2033	2034	2035	2051	2052	2053	2067	2068	2069	2086
	2087	2088	2108	2109	2110	2124	2125	2126	2141	2142	2143	2158	2159	2160	2174
	2175	2176	2191	2192	2193	2207	2208	2209	2224	2225	2226	2248	2249	2267	2268
	2284	2285	2304	2305	2323	2324	2341	2342	2361	2362	2406	2407	2423	2424	2441
	2442	2456	2457	2474	2475	2491	2492	2529	2530	2546	2547	2548	2565	2566	2567
	2584	2585	2586	2600	2601	2602	2619	2620	2621	2638	2639	2640	2654	2655	2656
	2673	2674	2675	2692	2693	2694	2708	2709	2710	2727	2728	2729	2745	2746	2747
	2761	2762	2763	2780	2781	2782	2796	2797	2798	2812	2813	2814	2828	2830	2831
	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2858	2860	2861
	2862	2863	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2888	2890	2891