

# EAE

BASIC LOGIC TEST  
MD-11-DZKEB-B

EP-DZKEB-B-DL  
COPYRIGHT © 70-78  
FICHE 1 OF 1

MAR 1978  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The frames on the left side contain data, while the right side is mostly blank. The data in the frames is organized into columns and rows, with some frames containing headers and footers. The text is small and difficult to read, but it appears to be a list or table of information. The frames are arranged in a regular grid pattern, with approximately 10 columns and 15 rows of frames visible on the left side.

IDENTIFICATION  
\*\*\*\*\*

PRODUCT CODE:           MAINDEC-11-DZKEB-B-D  
 PRODUCT NAME:           EAE BASIC LOGIC TEST  
 PRODUCT DATE:           FEB 1978  
 MAINTAINER:             DIAGNOSTIC ENGINEERING

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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1970, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

- 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 ALWAYS BE RUN.

## 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.

## 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.

5.2.5 TRAPCATCHER  
\*\*\*\*\*

SEQ 0005

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 (00000). (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

## 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 ARITH-  
METIC SHIFT, NORMALIZE, MULTIPLY, AND DIVIDE.

## 10. LISTING

FOLLOWING

## 11. FLOW CHART(S)

NA

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

```

*****
*****
*TITLE MAINDEC-11-DZKEB-B
*COPYRIGHT (C) 1970,1977
*DIGITAL EQUIPMENT CORP.
*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
*****
*****

```



```

35 000000 . = 0
36 000030 . = 30
37 000030 016366 PRINT
38 000032 000340 340
39 000034 . = 34
40 000034 017254 SCOPEC
41 000036 000000 0
42 000040 . = 40
43 000040 000000 0
44 000042 000000 0
45 000044 000000 0
46 000046 . = 46
47 000046 016222 SENDAD
48 000050 . = 50
49 000050 000000 0
50 000052 . = 52
51 000052 000000 0
52 104000 HLT = EMT
53 177776 PSW= 177776
54 000240 NOP= 240
55 104400 SCOPE= TRAP
56 000174 . = 174
57 000174 177570 SWR: 177570
58 000176 . = 176
59 000176 000000 SWREG: 0
60 000200 . = 200
61 000200 012706 017406 MOV #BUFF,%6 ;SET UP STACK FOR SCOPE LOOP
62 000204 022737 016222 000042 CMP #SENDAD,%4 ;CHECK FOR AUTO MODE
63 000212 001411 BEQ 1$ ;YES, DO NOT TYPE TITLE
64 000214 005005 CLR %5
65 000216 012705 020075 MOV #SHEAD,%5 ;TYPE PROGRAM NAME
66 000222 004767 017676 JSR %7,TTOUT
67 000226 012705 0200 5 MOV #SMIN,%5 ;TYPE MAINDEC NUMBER
68 000232 004767 017676 JSR %7,TTOUT
69 000236 000167 000022 1$: JMP SUSWR
70 000242 177300 DIV: 177300
71 000244 177302 AC: 177302
72 000246 177304 MQ: 177304
73 000250 177306 MUL: 177306
74 000252 177310 SC: 177310
75 000254 177311 SR: 177311
76 000256 177312 NOR: 177312
77 000260 177314 LSH: 177314
78 000262 177316 ASH: 177316
79
80
81
82
83 ;*****
84 ; TEST FOR HARDWARE SWITCH REGISTER
85 ;*****
86 000264 013746 000006 SUSWR: MOV @#6,-(6) ;SAVE VECTORS
87 000270 013746 000004 MOV @#4,-(6)
88 000274 012737 000314 000004 MOV #64,@#4 ;SET UP FOR TIMEOUT
89 000302 022777 177777 177664 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
90 000310 001402 BEQ 65$

```

91	CJ0312	000404				BR	66\$	
92	000314	022621				CMP	(6)+ (6)+	;ADJUST STACK
93	000316	012767	J00176	177650	64\$:	MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
94	000324	012637	000004		65\$:	MOV	(6)+,a#4	;RESTORE VECTORS
95	J00330	012637	000006		66\$:	MOV	(6)+,a#6	
96	000334	022767	000176	177632		CMP	#SWREG,SWR	; IS SWREG USED
97	000342	001006				BNE	BEGIN	
98	000344	022737	016222	000042		CMP	#SENDAD,a#42	;CHECK FOR AUTO MODE
99	000352	001402				BEQ	BEGIN	;YES, ASSUME WORST CASE
100	000354	004767	017114			JSR	%7,CNTLU	;ALLOW SWREG TO BE LOADED
101	000360	012767	000360	016750	BEGIN:	MOV	#BEGIN,RETURN	;SET UP RESTART OF PROGRAM
102	J00366	005077	177652			CLR	a#C	
103	000372	005077	177650			CLR	a#M	
104	000376	005003				CLR	%3	

```

105 ;*****
106 ; REGISTER TEST FOR WRITABILITY
107 ;*****
108 R^WRI:
109 000400 104400 SCOPE ;CHECK SR AND SC FLOPS FOR 0 STATE
110 000402 012777 000000 177642 MOV #0,ASC
111 000410 105777 177636 TSTB ASC
112 000414 001401 BEQ 64$ ; IF NO.ERROR SKIP HLT
113 000416 104000 HLT ;CALL ERROR ROUTINE
114 000420 64$:
115
116 000420 104400 SCOPE ;CHECK SR AND SC FLOPS FOR 1 STATE
117 000422 012777 140477 177622 MOV #140477,ASC
118 000430 017746 177616 MOV ASC, -(6)
119 000434 042716 037300 BIC #37300, (6)
120 000440 022726 140477 CMP #140477, (6)+
121 000444 001401 BEQ 65$ ; IF NO.ERROR SKIP HLT
122 000446 104000 HLT ;CALL ERROR ROUTINE
123 000450 65$:
124 000450 005077 177576 CLR ASC
125
126 000454 104400 SCOPE ;CHECK MQ FLOPS FOR 0 STATE
127 000456 012777 000000 177562 MOV #0,AMQ
128 000464 005777 177556 TST AMQ
129 000470 001401 BEQ 66$ ; IF NO.ERROR SKIP HLT
130 000472 104000 HLT ;CALL ERROR ROUTINE
131 000474 66$:
132
133 000474 122777 000036 177552 CMPB #36,ASR ;CHECK STATUS 36
134 000502 001401 BEQ 67$ ; IF NO.ERROR SKIP HLT
135 000504 104000 HLT ;CALL ERROR ROUTINE
136 000506 67$:
137
138 000506 104400 SCOPE ;CHECK MQ FLOPS FOR 1 STATE
139 000510 012777 177777 177530 MOV #-1,AMQ
140 000516 022777 177777 177522 CMP #-1,AMQ
141 000524 001401 BEQ 68$ ; IF NO.ERROR SKIP HLT
142 000526 104000 HLT ;CALL ERROR ROUTINE
143 000530 68$:
144
145 000530 122777 000042 177516 CMPB #42,ASR ;CHECK STATUS 42
146 000536 001401 BEQ 69$ ; IF NO.ERROR SKIP HLT
147 000540 104000 HLT ;CALL ERROR ROUTINE
148 000542 69$:
149 000542 104400 SCOPE ;CHECK AC FLOPS FOR 0 STATE
150 000544 012777 000000 177472 MOV #0,ASC
151 000552 005777 177466 TST ASC
152 000556 001401 BEQ 70$ ; IF NO.ERROR SKIP HLT
153 000560 104000 HLT ;CALL ERROR ROUTINE
154 000562 70$:
155
156 000562 122777 000020 177464 CMPB #20,ASR ;CHECK STATUS 20
157 000570 001401 BEQ 71$ ; IF NO.ERROR SKIP HLT
158 000572 104000 HLT ;CALL ERROR ROUTINE
159 000574 71$:
160

```

```

161 000574 104400          SCOPE          ;CHECK AC FLOPS FOR 1 STATE
162 000576 012777 177777 177440  MOV          #-1, @AC
163 000604 022777 177777 177432  CMP          #-1, @AC
164 000612 001401          BEQ          72$
165 000614 104000          HLT
166 000616          ;
167 000616 122777 000042 177430 72$:  CMPB        #42, @SR
168 000624 001401          BEQ          73$
169 000626 104000          HLT
170 000630          ;
171          ;
172 000630 104400          SCOPE          ;CHECK AC AND MQ WITH ALL NUMBERS
173 000632 005067 016552          CLR          CP
174 000636 005267 016546          INC          CP
175 000642 001420          BEQ          OUTPC
176 000644 016777 016540 177374  MOV          CP, @MQ
177 000652 026777 016532 177366  CMP          CP, @MQ
178 000660 001401          BEQ          64$
179 000662 104000          HLT
180 000664          ;
181 000664 016777 016520 177352  MOV          CP, @AC
182 000672 026777 016512 177344  CMP          CP, @AC
183 000700 001756          BEQ          CP1
184 000702 104000          HLT
185 000704 012767 004000 016422  OUTPC:  MOV          #4000, SCOPEF
186 000712 104400          SCOPE
187 000714 012777 177777 177324  MOV          #-1, @MQ
188 000722 022777 177777 177314  CMP          #-1, @AC
189 000730 001401          BEQ          64$
190 000732 104000          HLT
191 000734          ;
192          ;
193 000734 005077 177306          CLR          @MQ
194 000740 005777 177300          TST          @AC
195 000744 001401          BEQ          65$
196 000746 104000          HLT
197 000750          ;
198          ;
199 000750 112777 177777 177270  MOVB        #-1, @MQ
200 000756 022777 177777 177262  CMP          #-1, @MQ
201 000764 001401          BEQ          66$
202 000766 104000          HLT
203 000770          ;
204 000770 022777 177777 177246 66$:  CMP          #-1, @AC

```

```

; IF NO.ERROR SKIP HLT
; CALL ERROR ROUTINE

; CHECK STATUS 42
; IF NO.ERROR SKIP HLT
; CALL ERROR ROUTINE

; 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

```

```

205 000776 001401 BEQ 67$ ; IF NO.ERROR SKIP HLT
206 001000 104000 HLT ; CALL ERROR ROUTINE
207 001002 67$:
208
209 001002 105077 177240 CLRB 2MQ ; CHECK FOR BYTE ZERO SIGN EXTENTION
210 001006 005777 177234 TST 2MQ ; CHECK FOR ZERO MQ
211 001012 001401 BEQ 68$ ; IF NO.ERROR SKIP HLT
212 001014 104000 HLT ; CALL ERROR ROUTINE
213 001016 68$:
214 001016 005777 177222 TST 2AC ; CHECK FOR ZERO AC
215 001022 001401 BEQ 69$ ; IF NO.ERROR SKIP HLT
216 001024 104000 HLT ; CALL ERROR ROUTINE
217 001026 69$:
218
219 001026 012777 100000 177212 MOV #100000,2MQ ; LOAD MQ WITH LARGEST NUMBER
220 001034 022777 177777 177202 CMP #-1,2AC ; DID IT SIGN EXTEND
221 001042 001401 BEQ 70$ ; IF NO.ERROR SKIP HLT
222 001044 104000 HLT ; CALL ERROR ROUTINE
223 001046 70$:
224 001046 112777 000200 177172 MOVB #200,2MQ ; LOAD MQ WITH LARGEST BYTE
225 001054 022777 177777 177162 CMP #-1,2AC ; DID IT SIGN EXTEND
226 001062 001401 BEQ 71$ ; IF NO.ERROR SKIP HLT
227 001064 104000 HLT ; CALL ERROR ROUTINE
228 001066 71$:
229 001066 022777 177600 177152 CMP #177600,2MQ ; DID IT SIGN EXTEND
230 001074 001401 BEQ 72$ ; IF NO.ERROR SKIP HLT
231 001076 104000 HLT ; CALL ERROR ROUTINE
232 001100 72$:
233
234 001100 012777 000077 177144 MOV #77,2SC ; LOAD SC WITH -1
235 001106 022777 000077 177142 CMP #77,2NOR ; CHECK FOR SIGN EXTENTION
236 001114 001401 BEQ 73$ ; IF NO.ERROR SKIP HLT
237 001116 104000 HLT ; CALL ERROR ROUTINE
238 001120 73$:
239
240 001120 005777 177126 CLR 2SC ; CLEAR SC
241 001124 005777 177126 TST 2NOR ; CHECK NOR
242 001130 001401 BEQ 74$ ; IF NO.ERROR SKIP HLT
243 001132 104000 HLT ; CALL ERROR ROUTINE
244 001134 74$:
245
246 ;*****
247 ; AT THIS POINT, ALL THE REGISTERS CAN HANDLE DATA OF
248 ; ANY FORM
249 ;*****
250 001134 004767 016262 JSR %7,CKSWR ; CHECK FOR CONT-G
251 001140 022777 000001 177026 BIT #1,2SWR
252 001146 001402 BEQ MQCL
253 001150 000167 011736 JMP .DIV

```

255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296

001154  
001154 104400  
001156 005077 177062  
001162 005077 177060  
001166 012700 177777  
001172 005200  
001174 005077 177052  
001200 010077 177054  
001204 005777 177036  
001210 001401  
001212 104000  
001214  
001214 122777 000036 177032  
001222 001401  
001224 104000  
001226  
001232 022700 000020  
001357

```
*****
: LOGICAL SHIFT TEST SECTION
: TEST MQ SHIFT OF 0'S LEFT
*****
MOVL:
SCOPE
CLR @AC
CLR @MQ
MOV #-1,%0
LOOP: INC %0
CLR @SC ;CLEAR SR AND SC
MOV %0,@LSH ;SHIFT R0 TIMES LEFT
TST @MQ ;TEST MQ FOR '0'
BEQ 64$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
64$: CMPB #36,@SR ;CHECK STATUS REGISTER
BEQ 65$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
65$: CMP #16,%0 ;LAST ONE
BNE LOOP

*****
: TEST AC SHIFT OF 0'S RIGHT
*****
MOVL:
SCOPE
MOV #1,%0
DEC %0
CLR @SC ;CLEAR SR AND SC
MOV %0,@LSH ;SHIFT R0 TIMES RIGHT
TST @AC ;TEST AC FOR '0'
BEQ 64$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
64$: CMPB #36,@SR ;CHECK STATUS REGISTER
BEQ 65$ ;IF NO.ERROR SKIP HLT
HLT ;CALL ERROR ROUTINE
65$: CMP #-16,%0 ;LAST ONE
BNE LOOP1
```

297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352

001304  
001304 104400  
001306 012777 177777 176732  
001314 005077 176724  
001320 005077 176734  
001324 027727 176716 177777  
001332 001401  
001334 104000  
001336  
001336 122777 000020 176710  
001344 001401  
001346 104000  
001350  
001350 104400  
001352 012777 177777 176666  
001360 005077 176660  
001364 012777 177777 176666  
001372 022777 077777 175646  
001400 001401  
001402 104000  
001404  
001404 122777 000023 176642  
001412 001401  
001414 104000  
001416  
001416 104400  
001420 012777 177777 176620  
001426 005077 176612  
001432 012777 177776 176620  
001440 022777 037777 176600  
001446 001401  
001450 104000  
001452  
001452 122777 000023 176574  
001460 001401  
001462 104000  
001464  
001464 104400  
001466 012777 177777 176552  
001474 005077 176544  
001500 012777 177775 176552  
001506 022777 017777 176532  
001514 001401  
001516 104000  
001520  
001520 122777 000023 176526  
001526 001401  
001530 104000  
001532  
001532 104400  
001534 012777 177777 176504

```
*****  
: TEST MQ SHIFT OF 1'S RIGHT  
*****  
MQ1R:  
SCOPE  
MOV #-1, @MQ ; SET MQ=-1  
CLR @AC ; CLEAR AC  
CLR @LSH ; INITIALIZE SHIFT BY 0  
CMP @MQ, #-1 ; COMPARE UNSHIFTED MQ TO -1  
BEQ 64$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
64$:  
CMPB #20, @SR ; CHECK STATUS  
BEQ 65$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
65$:  
SCOPE  
MOV #-1, @MQ ; RESET MQ  
CLR @AC  
MOV #177777, @LSH ; SHIFT 177777 TIMES RIGHT  
CMP #77777, @MQ ; CHECK MQ  
BEQ 66$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
66$:  
CMPB #23, @SR ; CHECK STATUS  
BEQ 67$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
67$:  
SCOPE  
MOV #-1, @MQ ; RESET MQ  
CLR @AC  
MOV #177776, @LSH ; SHIFT 177776 TIMES RIGHT  
CMP #37777, @MQ ; CHECK MQ  
BEQ 68$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
68$:  
CMPB #23, @SR ; CHECK STATUS  
BEQ 69$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
69$:  
SCOPE  
MOV #-1, @MQ ; RESET MQ  
CLR @AC  
MOV #177775, @LSH ; SHIFT 177775 TIMES RIGHT  
CMP #17777, @MQ ; CHECK MQ  
BEQ 70$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
70$:  
CMPB #23, @SR ; CHECK STATUS  
BEQ 71$ ; IF NO ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
71$:  
SCOPE  
MOV #-1, @MQ ; RESET MQ
```

353	001542	005077	176476		CLR	0AC		
354	001546	012777	177774	176504	MOV	#177774, 0LSH		;SHIFT 177774 TIMES RIGHT
355	001554	022777	007777	176464	CMP	#7777, 0MQ		;CHECK MQ
356	001562	001401			BEQ	72\$		;IF NO.ERROR SKIP HLT
357	001564	104000			HLT			;CALL ERROR ROUTINE
358	001566						72\$:	
359	001566	122777	000023	176460	CMPB	#23, 0SR		;CHECK STATUS
360	001574	001401			BEQ	73\$		;IF NO.ERROR SKIP HLT
361	001576	104000			HLT			;CALL ERROR ROUTINE
362	001600						73\$:	
363								
364	001600	104400			SCOPE			
365	001602	012777	177777	176436	MOV	#-1, 0MQ		;RESET MQ
366	001610	005077	176430		CLR	0AC		
367	001614	012777	177773	176436	MOV	#177773, 0LSH		;SHIFT 177773 TIMES RIGHT
368	001622	000241			CLC			
369	001624	022777	003777	176414	CMP	#3777, 0MQ		;CHECK MQ
370	001632	001401			BEQ	74\$		;IF NO.ERROR SKIP HLT
371	001634	104000			HLT			;CALL ERROR ROUTINE
372	001636						74\$:	
373	001636	122777	000023	176410	CMPB	#23, 0SR		;CHECK STATUS
374	001644	001401			BEQ	75\$		;IF NO.ERROR SKIP HLT
375	001646	104000			HLT			;CALL ERROR ROUTINE
376	001650						75\$:	
377								
378	001650	104400			SCOPE			
379	001652	012777	177777	176366	MOV	#-1, 0MQ		;RESET MQ
380	001660	005077	176360		CLR	0AC		
381	001664	012777	177772	176366	MOV	#177772, 0LSH		;SHIFT 177772 TIMES RIGHT
382	001672	022777	001777	176346	CMP	#1777, 0MQ		;CHECK MQ
383	001700	001401			BEQ	76\$		;IF NO.ERROR SKIP HLT
384	001702	104000			HLT			;CALL ERROR ROUTINE
385	001704						76\$:	
386	001704	122777	000023	176342	CMPB	#23, 0SR		;CHECK STATUS
387	001712	001401			BEQ	77\$		;IF NO.ERROR SKIP HLT
388	001714	104000			HLT			;CALL ERROR ROUTINE
389	001716						77\$:	
390	001716	104400			SCOPE			
391	001720	012777	177777	176320	MOV	#-1, 0MQ		;RESET MQ
392	001726	005077	176312		CLR	0AC		
393	001732	012777	177771	176320	MOV	#177771, 0LSH		;SHIFT 177771 TIMES RIGHT
394	001740	022777	000777	176300	CMP	#777, 0MQ		;CHECK MQ
395	001746	001401			BEQ	78\$		;IF NO.ERROR SKIP HLT
396	001750	104000			HLT			;CALL ERROR ROUTINE
397	001752						78\$:	
398	001752	122777	000023	176274	CMPB	#23, 0SR		;CHECK STATUS



```

399 001760 001401          BEQ      79$          ; IF NO.ERROR SKIP HLT
400 001762 104000          HLT                    ; CALL ERROR ROUTINE
401 001764          79$:
402 001764 104400          SCOPE
403 001766 012777 177777 176252  MOV     #-1, @MQ      ; RESET MQ
404 001774 005077 176244  CLR     @AC
405 002000 012777 177770 176252  MOV     #177770, @LSH ; SHIFT 177770 TIMES RIGHT
406 002006 022777 000377 176232  CMP     #377, @MQ     ; CHECK MQ
407 002014 001401          BEQ      80$          ; IF NO.ERROR SKIP HLT
408 002016 104000          HLT                    ; CALL ERROR ROUTINE
409 002020          80$:
410 002020 122777 000023 176226  CMPB   #23, @SR      ; CHECK STATUS
411 002026 001401          BEQ      81$          ; IF NO.ERROR SKIP HLT
412 002030 104000          HLT                    ; CALL ERROR ROUTINE
413 002032          81$:
414
415 002032 104400          SCOPE
416 002034 012777 177777 176204  MOV     #-1, @MQ      ; RESET MQ
417 002042 005077 176176  CLR     @AC
418 002046 012777 177767 176204  MOV     #177767, @LSH ; SHIFT 177767 TIMES RIGHT
419 002054 022777 000177 176164  CMP     #177, @MQ     ; CHECK MQ
420 002062 001401          BEQ      82$          ; IF NO.ERROR SKIP HLT
421 002064 104000          HLT                    ; CALL ERROR ROUTINE
422 002066          82$:
423 002066 122777 000023 176160  CMPB   #23, @SR      ; CHECK STATUS
424 002074 001401          BEQ      83$          ; IF NO.ERROR SKIP HLT
425 002076 104000          HLT                    ; CALL ERROR ROUTINE
426 002100          83$:
427
428 002100 104400          SCOPE
429 002102 012777 177777 176136  MOV     #-1, @MQ      ; RESET MQ
430 002110 005077 176130  CLR     @AC
431 002114 012777 177766 176136  MOV     #177766, @LSH ; SHIFT 177766 TIMES RIGHT
432 002122 022777 000077 176116  CMP     #77, @MQ     ; CHECK MQ
433 002130 001401          BEQ      84$          ; IF NO.ERROR SKIP HLT
434 002132 104000          HLT                    ; CALL ERROR ROUTINE
435 002134          84$:
436 002134 122777 000023 176112  CMPB   #23, @SR      ; CHECK STATUS
437 002142 001401          BEQ      85$          ; IF NO.ERROR SKIP HLT
438 002144 104000          HLT                    ; CALL ERROR ROUTINE
439 002146          85$:
440
441 002146 104400          SCOPE
442 002150 012777 177777 176070  MOV     #-1, @MQ      ; RESET MQ
443 002156 005077 176062  CLR     @AC
444 002162 012777 177765 176070  MOV     #177765, @LSH ; SHIFT 177765 TIMES RIGHT
445 002170 022777 000037 176050  CMP     #37, @MQ     ; CHECK MQ
446 002176 001401          BEQ      86$          ; IF NO.ERROR SKIP HLT
447 002200 104000          HLT                    ; CALL ERROR ROUTINE
448 002202          86$:
449 002202 122777 000023 176044  CMPB   #23, @SR      ; CHECK STATUS

```

## E02

MAINDEC-11-DZKEB-B MACY11 30(1046) 01-NOV-77 08:13 PAGE 11  
 OZKEBB.P11 01-NOV-77 08:12

SEQ 0017

450	002210	001401			BEQ	87\$			; IF NO.ERROR SKIP HLT
451	002212	104000			HLT				; CALL ERROR ROUTINE
452	002214						87\$:		
453	002214	104400			SCOPE				
454	002216	012777	177777	176022	MOV	#-1, @MQ			; RESET MQ
455	002224	005077	176014		CLR	@AC			
456	002230	012777	177764	176022	MOV	#177764, @LSH			; SHIFT 177764 TIMES RIGHT
457	002236	022777	000017	176002	CMP	#17, @MQ			; CHECK MQ
458	002244	001401			BEQ	88\$			; IF NO.ERROR SKIP HLT
459	002246	104000			HLT				; CALL ERROR ROUTINE
460	002250						88\$:		
461	002250	122777	000023	175776	CMPB	#23, @SR			; CHECK STATUS
462	002256	001401			BEQ	89\$			; IF NO.ERROR SKIP HLT
463	002260	104000			HLT				; CALL ERROR ROUTINE
464	002262						89\$:		
465									
466	002262	104400			SCOPE				
467	002264	012777	177777	175754	MOV	#-1, @MQ			; RESET MQ
468	002272	005077	175746		CLR	@AC			
469	002276	012777	177763	175754	MOV	#177763, @LSH			; SHIFT 177763 TIMES RIGHT
470	002304	022777	000007	175734	CMP	#7, @MQ			; CHECK MQ
471	002312	001401			BEQ	90\$			; IF NO.ERROR SKIP HLT
472	002314	104000			HLT				; CALL ERROR ROUTINE
473	002316						90\$:		
474	002316	122777	000023	175730	CMPB	#23, @SR			; CHECK STATUS
475	002324	001401			BEQ	91\$			; IF NO.ERROR SKIP HLT
476	002326	104000			HLT				; CALL ERROR ROUTINE
477	002330						91\$:		
478									
479	002330	104400			SCOPE				
480	002332	012777	177777	175706	MOV	#-1, @MQ			; RESET MQ
481	002340	005077	175700		CLR	@AC			
482	002344	012777	177762	175706	MOV	#177762, @LSH			; SHIFT 177762 TIMES RIGHT
483	002352	022777	000003	175666	CMP	#3, @MQ			; CHECK MQ
484	002360	001401			BEQ	92\$			; IF NO.ERROR SKIP HLT
485	002362	104000			HLT				; CALL ERROR ROUTINE
486	002364						92\$:		
487	002364	122777	000023	175662	CMPB	#23, @SR			; CHECK STATUS
488	002372	001401			BEQ	93\$			; IF NO.ERROR SKIP HLT
489	002374	104000			HLT				; CALL ERROR ROUTINE
490	002376						93\$:		
491									
492	002376	104400			SCOPE				
493	002400	012777	177777	175640	MOV	#-1, @MQ			; RESET MQ
494	002406	005077	175632		CLR	@AC			
495	002412	012777	177761	175640	MOV	#177761, @LSH			; SHIFT 177761 TIMES RIGHT
496	002420	022777	000001	175620	CMP	#1, @MQ			; CHECK MQ
497	002426	001401			BEQ	94\$			; IF NO.ERROR SKIP HLT
498	002430	104000			HLT				; CALL ERROR ROUTINE
499	002432						94\$:		
500	002432	122777	000023	175614	CMPB	#23, @SR			; CHECK STATUS
501	002440	001401			BEQ	95\$			; IF NO.ERROR SKIP HLT
502	002442	104000			HLT				; CALL ERROR ROUTINE
503	002444						95\$:		

```

504 ;*****
505 ; TEST AC SHIFT OF 1'S LEFT
506 ;*****
507 AC1L:
508 002444 104400 SCOPE
509 002446 005077 175574 CLR @MG
510 002452 012777 177777 175564 MOV #-1,@AC ;SET AC=-1
511 002460 005077 175574 CLR @LSH ;INITALIZE SHIFT BY 0
512 002464 027727 175554 177777 CMP @AC,#-1 ;COMPARE UNSHIFTED AC TO -1
513 002472 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT
514 002474 104000 HLT ;CALL ERROR ROUTINE
515 002476 122777 000350 175550 64$: CMPB #350,@SR ;CHECK STATUS
516 002504 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT
517 002506 104000 HLT ;CALL ERROR ROUTINE
518 002510 104400 SCOPE
519 002512 012777 177777 175524 MOV #-1,@AC ;SET AC=-1
520 002520 012777 000001 175532 MOV #1,@LSH ;SHIFT 1 TIMES LEFT
521 002526 022777 177776 175510 CMP #177776,@AC ;CHECK AC
522 002534 001401 BEQ 66$ ;IF NO.ERROR SKIP HLT
523 002536 104000 HLT ;CALL ERROR ROUTINE
524 002540 122777 000311 175506 66$: CMPB #311,@SR ;CHECK STATUS
525 002546 001401 BEQ 67$ ;IF NO.ERROR SKIP HLT
526 002550 104000 HLT ;CALL ERROR ROUTINE
527 002552 104400 SCOPE
528 002554 012777 177777 175462 MOV #-1,@AC ;SET AC=-1
529 002562 012777 000002 175470 MOV #2,@LSH ;SHIFT 2 TIMES LEFT
530 002570 022777 177774 175446 CMP #177774,@AC ;CHECK AC
531 002576 001401 BEQ 68$ ;IF NO.ERROR SKIP HLT
532 002600 104000 HLT ;CALL ERROR ROUTINE
533 002602 122777 000311 175444 68$: CMPB #311,@SR ;CHECK STATUS
534 002610 001401 BEQ 69$ ;IF NO.ERROR SKIP HLT
535 002612 104000 HLT ;CALL ERROR ROUTINE
536 002614 104400 SCOPE
537 002616 012777 177777 175420 MOV #-1,@AC ;SET AC=-1
538 002624 012777 000003 175426 MOV #3,@LSH ;SHIFT 3 TIMES LEFT
539 002632 022777 177770 175404 CMP #177770,@AC ;CHECK AC
540 002640 001401 BEQ 70$ ;IF NO.ERROR SKIP HLT
541 002642 104000 HLT ;CALL ERROR ROUTINE
542 002644 122777 000311 175402 70$: CMPB #311,@SR ;CHECK STATUS
543 002652 001401 BEQ 71$ ;IF NO.ERROR SKIP HLT
544 002654 104000 HLT ;CALL ERROR ROUTINE
545 002656 104400 SCOPE
546 002660 012777 177777 175356 MOV #-1,@AC ;SET AC=-1
547 002666 012777 000004 175364 MOV #4,@LSH ;SHIFT 4 TIMES LEFT

```

560	002674	022777	177760	175342		CMP	#177760, @AC		:CHECK AC
561	002702	001401				BEQ	72\$		:IF NO.ERROR SKIP HLT
562	002704	104000				HLT			:CALL ERROR ROUTINE
563	002706				72\$:				
564	002706	122777	000311	175340		CMPB	#311, @SR		:CHECK STATUS
565	002714	001401				BEQ	73\$		:IF NO.ERROR SKIP HLT
566	002716	104000				HLT			:CALL ERROR ROUTINE
567	002720				73\$:				
568									
569									
570	002720	104400				SCOPE			
571	002722	012777	177777	175314		MOV	#-1, @AC		:SET AC=-1
572	002730	012777	000005	175322		MOV	#5, @LSH		:SHIFT 5 TIMES LEFT
573	002736	022777	177740	175300		CMP	#177740, @AC		:CHECK AC
574	002744	001401				BEQ	74\$		:IF NO.ERROR SKIP HLT
575	002746	104000				HLT			:CALL ERROR ROUTINE
576	002750				74\$:				
577	002750	122777	000311	175276		CMPB	#311, @SR		:CHECK STATUS
578	002756	001401				BEQ	75\$		:IF NO.ERROR SKIP HLT
579	002760	104000				HLT			:CALL ERROR ROUTINE
580	002762				75\$:				
581									
582	002762	104400				SCOPE			
583	002764	012777	177777	175252		MOV	#-1, @AC		:SET AC=-1
584	002772	012777	000006	175260		MOV	#6, @LSH		:SHIFT 6 TIMES LEFT
585	003000	022777	177700	175236		CMP	#177700, @AC		:CHECK AC
586	003006	001401				BEQ	76\$		:IF NO.ERROR SKIP HLT
587	003010	104000				HLT			:CALL ERROR ROUTINE
588	003012				76\$:				
589	003012	122777	000311	175234		CMPB	#311, @SR		:CHECK STATUS
590	003020	001401				BEQ	77\$		:IF NO.ERROR SKIP HLT
591	003022	104000				HLT			:CALL ERROR ROUTINE
592	003024				77\$:				
593									
594									
595	003024	104400				SCOPE			
596	003026	012777	177777	175210		MOV	#-1, @AC		:SET AC=-1
597	003034	012777	000007	175216		MOV	#7, @LSH		:SHIFT 7 TIMES LEFT
598	003042	022777	177600	175174		CMP	#177600, @AC		:CHECK AC
599	003050	001401				BEQ	78\$		:IF NO.ERROR SKIP HLT
600	003052	104000				HLT			:CALL ERROR ROUTINE
601	003054				78\$:				
602	003054	122777	000311	175172		CMPB	#311, @SR		:CHECK STATUS

603	003062	001401			BEQ	79\$				: IF NO.ERROR SKIP HLT
604	003064	104000			HLT					: CALL ERROR ROUTINE
605	003066				79\$:					
606	003066	104400			SCOPE					
607	003070	012777	177777	175155	MOV	#-1, @AC				: SET AC=-1
608	003076	012777	000010	175154	MOV	#10, @LSH				: SHIFT 10 TIMES LEFT
609	003104	022777	177400	175132	CMP	#17400, @AC				: CHECK AC
610	003112	001401			BEQ	80\$				: IF NO.ERROR SKIP HLT
611	003114	104000			HLT					: CALL ERROR ROUTINE
612	003116				80\$:					
613	003116	122777	000311	175130	CMPB	#311, @SR				: CHECK STATUS
614	003124	001401			BEQ	81\$				: IF NO.ERROR SKIP HLT
615	003126	104000			HLT					: CALL ERROR ROUTINE
616	003130				81\$:					
617										
618										
619	003130	104400			SCOPE					
620	003132	012777	177777	175104	MOV	#-1, @AC				: SET AC=-1
621	003140	012777	000011	175112	MOV	#11, @LSH				: SHIFT 11 TIMES LEFT
622	003146	022777	177000	175070	CMP	#177000, @AC				: CHECK AC
623	003154	001401			BEQ	82\$				: IF NO.ERROR SKIP HLT
624	003156	104000			HLT					: CALL ERROR ROUTINE
625	003160				82\$:					
626	003160	122777	000311	175066	CMPB	#311, @SR				: CHECK STATUS
627	003166	001401			BEQ	83\$				: IF NO.ERROR SKIP HLT
628	003170	104000			HLT					: CALL ERROR ROUTINE
629	003172				83\$:					
630										
631										
632	003172	104400			SCOPE					
633	003174	012777	177777	175042	MOV	#-1, @AC				: SET AC=-1
634	003202	012777	000012	175050	MOV	#12, @LSH				: SHIFT 12 TIMES LEFT
635	003210	022777	176000	175026	CMP	#176000, @AC				: CHECK AC
636	003216	001401			BEQ	84\$				: IF NO.ERROR SKIP HLT
637	003220	104000			HLT					: CALL ERROR ROUTINE
638	003222				84\$:					
639	003222	122777	000311	175024	CMPB	#311, @SR				: CHECK STATUS
640	003230	001401			BEQ	85\$				: IF NO.ERROR SKIP HLT
641	003232	104000			HLT					: CALL ERROR ROUTINE
642	003234				85\$:					
643										
644	003234	104400			SCOPE					
645	003236	012777	177777	175000	MOV	#-1, @AC				: SET AC=-1
646	003244	012777	000013	175006	MOV	#13, @LSH				: SHIFT 13 TIMES LEFT
647	003252	022777	174000	174764	CMP	#174000, @AC				: CHECK AC
648	003260	001401			BEQ	86\$				: IF NO.ERROR SKIP HLT
649	003262	104000			HLT					: CALL ERROR ROUTINE
650	003264				86\$:					
651	003264	122777	000311	174762	CMPB	#311, @SR				: CHECK STATUS

652	003272	001401			BEQ	87\$			; IF NO.ERROR SKIP HLT
653	003274	104000			HLT				; CALL ERROR ROUTINE
654	003276					87\$:			
655	003276	104400			SCOPE				
656	003300	012777	177777	174736	MOV	#-1, @AC			; SET AC=-1
657	003306	012777	000014	174744	MOV	#14, @LSH			; SHIFT 14 TIMES LEFT
658	003314	022777	170000	174722	CMP	#170000, @AC			; CHECK AC
659	003322	001401			BEQ	88\$			; IF NO.ERROR SKIP HLT
660	003324	104000			HLT				; CALL ERROR ROUTINE
661	003326					88\$:			
662	003326	122777	000311	174720	CMPB	#311, @SR			; CHECK STATUS
663	003334	001401			BEQ	89\$			; IF NO.ERROR SKIP HLT
664	003336	104000			HLT				; CALL ERROR ROUTINE
665	003340					89\$:			
666									
667									
668	003340	104400			SCOPE				
669	003342	012777	177777	174674	MOV	#-1, @AC			; SET AC=-1
670	003350	012777	000015	174702	MOV	#15, @LSH			; SHIFT 15 TIMES LEFT
671	003356	022777	160000	174660	CMP	#160000, @AC			; CHECK AC
672	003364	001401			BEQ	90\$			; IF NO.ERROR SKIP HLT
673	003366	104000			HLT				; CALL ERROR ROUTINE
674	003370					90\$:			
675	003370	122777	000311	174656	CMPB	#311, @SR			; CHECK STATUS
676	003376	001401			BEQ	91\$			; IF NO.ERROR SKIP HLT
677	003400	104000			HLT				; CALL ERROR ROUTINE
678	003402					91\$:			
679									
680									
681	003402	104400			SCOPE				
682	003404	012777	177777	174632	MOV	#-1, @AC			; SET AC=-1
683	003412	012777	000016	174640	MOV	#16, @LSH			; SHIFT 16 TIMES LEFT
684	003420	022777	140000	174616	CMP	#140000, @AC			; CHECK AC
685	003426	001401			BEQ	92\$			; IF NO.ERROR SKIP HLT
686	003430	104000			HLT				; CALL ERROR ROUTINE
687	003432					92\$:			
688	003432	122777	000311	174614	CMPB	#311, @SR			; CHECK STATUS
689	003440	001401			BEQ	93\$			; IF NO.ERROR SKIP HLT
690	003442	104000			HLT				; CALL ERROR ROUTINE
691	003444					93\$:			
692									
693									
694	003444	104400			SCOPE				
695	003446	012777	177777	174570	MOV	#-1, @AC			; SET AC=-1
696	003454	012777	000017	174576	MOV	#17, @LSH			; SHIFT 17 TIMES LEFT
697	003462	022777	100000	174554	CMP	#100000, @AC			; CHECK AC
698	003470	001401			BEQ	94\$			; IF NO.ERROR SKIP HLT
699	003472	104000			HLT				; CALL ERROR ROUTINE
700	003474					94\$:			
701	003474	122777	000311	174552	CMPB	#311, @SR			; CHECK STATUS
702	003502	001401			BEQ	95\$			; IF NO.ERROR SKIP HLT
703	003504	104000			HLT				; CALL ERROR ROUTINE
704	003506					95\$:			

```

705 ;*****
706 ; TEST MQ SHIFT RIGHT OF ALTERNATE 1'S AND 0'S
707 ;*****
708
709 MQ10R:
710 003506 104400 SCOPE
711 003506 012777 .25252 174530 MOV #125252, @MQ ;SET MQ=125252
712 003516 005C77 174522 CLR @AC ;CLEAR AC
713 003522 005077 174532 CLR @LSH ;INITIALIZE SHIFT BY 0
714 003536 027727 174514 125252 CMP @MQ, #125252 ;COMPARE MQ
715 003534 001401 BEQ 64$ ;IF NO.ERROR SKIP HLT
716 003536 104000 HLT ;CALL ERROR ROUTINE
717 003540
718 003540 122777 000020 174506 64$: CMPB #20, @SR ;CHECK STATUS
719 003546 001401 BEQ 65$ ;IF NO.ERROR SKIP HLT
720 003550 104000 HLT ;CALL ERROR ROUTINE
721 003552
722
723
724 003552 104400 SCOPE
725 003554 012777 125252 174464 MOV #125252, @MQ ;SET MQ=125252
726 003562 012777 177777 174470 MOV #177777, @LSH ;SHIFT 177777 TIMES
727 003570 105777 174460 TSTB @SR ;CHECK STATUS - NO CARRY
728 003574 001401 LEQ 66$ ;IF NO.ERROR SKIP HLT
729 003576 104000 HLT ;CALL ERROR ROUTINE
730 003600
731 003600 022777 152525 174440 66$: CMP #152525, @MQ ;COMPARE MQ
732 003606 001401 BEQ 67$ ;IF NO.ERROR SKIP HLT
733 003610 104000 HLT ;CALL ERROR ROUTINE
734 003612
735 67$:
736 003612 104400 SCOPE
737 003614 012777 125252 174424 MOV #125252, @MQ ;SET MQ=125252
738 003622 012777 177776 174430 MOV #177776, @LSH ;SHIFT 177776 TIMES
739 003630 122777 000001 174416 CMPB #1, @SR ;CHECK STATUS - WITH CARRY
740 003636 001401 BEQ 68$ ;IF NO.ERROR SKIP HLT
741 003640 104000 HLT ;CALL ERROR ROUTINE
742 003642
743 003642 022777 165252 174376 68$: CMP #165252, @MQ ;COMPARE MQ
744 003650 001401 BEQ 69$ ;IF NO.ERROR SKIP HLT
745 003652 104000 HLT ;CALL ERROR ROUTINE
746 003654
747 69$:
748 003654 104400 SCOPE
749 003656 012777 125252 174362 MOV #125252, @MQ ;SET MQ=125252
750 003664 012777 177775 174366 MOV #177775, @LSH ;SHIFT 177775 TIMES
751 003672 105777 174356 TSTB @SR ;CHECK STATUS - NO CARRY
752 003676 001401 BEQ 70$ ;IF NO.ERROR SKIP HLT
753 003700 104000 HLT ;CALL ERROR ROUTINE
754 003702
755 003702 022777 172525 174336 70$: CMP #172525, @MQ ;COMPARE MQ
756 003710 001401 BEQ 71$ ;IF NO.ERROR SKIP HLT
757 003712 104000 HLT ;CALL ERROR ROUTINE
758 003714
759 003714 104400 SCOPE
760 003716 012777 125252 174322 MOV #125252, @MQ ;SET MQ=125252

```

761	003724	012777	177774	174326	MOV	#177774, @LSH	;SHIFT 177774 TIMES
762	003732	122777	000001	174314	CMPB	#1, @SR	;CHECK STATUS - WITH CARRY
763	003740	001401			BEQ	72\$	;IF NO.ERROR SKIP HLT
764	003742	104000			HLT		;CALL ERROR ROUTINE
765	003744				72\$:		
766	003744	022777	175252	174274	CMP	#175252, @MQ	;COMPARE MQ
767	003752	001401			BEQ	73\$	;IF NO.ERROR SKIP HLT
768	003754	104000			HLT		;CALL ERROR ROUTINE
769	003756				73\$:		
770	003756	104400			SCOPE		
771	003760	012777	125252	174260	MOV	#125252, @MQ	;SET MQ=125252
772	003766	012777	177773	174264	MOV	#177773, @LSH	;SHIFT 177773 TIMES
773	003774	105777	17254		TSTB	@SR	;CHECK STATUS - NO CARRY
774	004000	001401			BEQ	74\$	;IF NO.ERROR SKIP HLT
775	004002	104000			HLT		;CALL ERROR ROUTINE
776	004004				74\$:		
777	004004	022777	176525	174234	CMP	#176525, @MQ	;COMPARE MQ
778	004012	001401			BEQ	75\$	;IF NO.ERROR SKIP HLT
779	004014	104000			HLT		;CALL ERROR ROUTINE
780	004016				75\$:		
781							
782	004016	104400			SCOPE		
783	004020	012777	125252	174220	MOV	#125252, @MQ	;SET MQ=125252
784	004026	012777	177772	174224	MOV	#177772, @LSH	;SHIFT 177772 TIMES
785	004034	122777	000001	174212	CMPB	#1, @SR	;CHECK STATUS - WITH CARRY
786	004042	001401			BEQ	76\$	;IF NO.ERROR SKIP HLT
787	004044	104000			HLT		;CALL ERROR ROUTINE
788	004046				76\$:		
789	004046	022777	177252	174172	CMP	#177252, @MQ	;COMPARE MQ
790	004054	001401			BEQ	77\$	;IF NO.ERROR SKIP HLT
791	004056	104000			HLT		;CALL ERROR ROUTINE
792	004060				77\$:		
793							
794	004060	104400			SCOPE		
795	004062	012777	125252	174156	MOV	#125252, @MQ	;SET MQ=125252
796	004070	012777	177771	174162	MOV	#177771, @LSH	;SHIFT 177771 TIMES
797	004076	105777	174152		TSTB	@SR	;CHECK STATUS - NO CARRY
798	004102	001401			BEQ	79\$	;IF NO.ERROR SKIP HLT
799	004104	104000			HLT		;CALL ERROR ROUTINE
800	004106				78\$:		
801	004106	022777	177525	174132	CMP	#177525, @MQ	;COMPARE MQ



802	004114	001401			BEQ	79\$			; IF NO.ERROR SKIP HLT
803	004116	104000			HLT				; CALL ERROR ROUTINE
804	004120						79\$:		
805	004120	104400			SCOPE				
806	004122	012777	125252	174116	MOV	#125252, @MQ			; SET MQ=125252
807	004130	012777	177770	174122	MOV	#177770, @LSH			; SHIFT 177770 TIMES
808	004136	122777	000001	174110	CMPB	#1, @SR			; CHECK STATUS - WITH CARRY
809	004144	001401			BEQ	80\$			; IF NO.ERROR SKIP HLT
810	004146	104000			HLT				; CALL ERROR ROUTINE
811	004150						80\$:		
812	004150	022777	177652	174070	CMP	#177652, @MQ			; COMPARE MQ
813	004156	001401			BEQ	81\$			; IF NO.ERROR SKIP HLT
814	004160	104000			HLT				; CALL ERROR ROUTINE
815	004162						81\$:		
816									
817	004162	104400			SCOPE				
818	004164	012777	125252	174054	MOV	#125252, @MQ			; SET MQ=125252
819	004172	012777	177767	174060	MOV	#177767, @LSH			; SHIFT 177767 TIMES
820	004200	105777	174050		TSTB	@SR			; CHECK STATUS - NO CARRY
821	004204	001401			BEQ	82\$			; IF NO.ERROR SKIP HLT
822	004206	104000			HLT				; CALL ERROR ROUTINE
823	004210						82\$:		
824	004210	022777	177725	174030	CMP	#177725, @MQ			; COMPARE MQ
825	004216	001401			BEQ	83\$			; IF NO.ERROR SKIP HLT
826	004220	104000			HLT				; CALL ERROR ROUTINE
827	004222						83\$:		
828									
829	004222	104400			SCOPE				
830	004224	012777	125252	174014	MOV	#125252, @MQ			; SET MQ=125252
831	004232	012777	177766	174020	MOV	#177766, @LSH			; SHIFT 177766 TIMES
832	004240	122777	000001	174006	CMPB	#1, @SR			; CHECK STATUS - WITH CARRY
833	004246	001401			BEQ	84\$			; IF NO.ERROR SKIP HLT
834	004250	104000			HLT				; CALL ERROR ROUTINE
835	004252						84\$:		
836	004252	022777	177752	173766	CMP	#177752, @MQ			; COMPARE MQ
837	004260	001401			BEQ	85\$			; IF NO.ERROR SKIP HLT
838	004262	104000			HLT				; CALL ERROR ROUTINE
839	004264						85\$:		
840									
841	004264	104400			SCOPE				
842	004266	012777	125252	173752	MOV	#125252, @MQ			; SET MQ=125252
843	004274	012777	177765	173756	MOV	#177765, @LSH			; SHIFT 177765 TIMES
844	004302	122777	173746		TSTB	@SR			; CHECK STATUS - NO CARRY
845	004306	001401			BEQ	86\$			; IF NO.ERROR SKIP HLT
846	004310	104000			HLT				; CALL ERROR ROUTINE
847	004312						86\$:		
848	004312	022777	177765	173726	CMP	#177765, @MQ			; COMPARE MQ

849	004320	001401			BEQ	87\$			; IF NO.ERROR SKIP HLT
850	004322	104000			HLT				; CALL ERROR ROUTINE
851	004324					87\$:			
852	004324	104400			SCOPE				
853	004326	012777	125252	173712	MOV	#125252, @MQ		; SET MQ=125252	
854	004334	012777	177764	173716	MOV	#177764, @LSH		; SHIFT 177764 TIMES	
855	004342	122777	000001	173704	CMPB	#1, @SR		; CHECK STATUS - WITH CARRY	
856	004350	001401			BEQ	88\$			; IF NO.ERROR SKIP HLT
857	004352	104000			HLT				; CALL ERROR ROUTINE
858	004354					88\$:			
859	004354	022777	177772	173664	CMP	#177772, @MQ		; COMPARE MQ	
860	004362	001401			BEQ	89\$			; IF NO.ERROR SKIP HLT
861	004364	104000			HLT				; CALL ERROR ROUTINE
862	004366					89\$:			
863									
864	004366	104400			SCOPE				
865	004370	012777	125252	173650	MOV	#125252, @MQ		; SET MQ=125252	
866	004376	012777	177763	173654	MOV	#177763, @LSH		; SHIFT 177763 TIMES	
867	004404	105777	173644		TSTB	@SR		; CHECK STATUS - NO CARRY	
868	004410	001401			BEQ	90\$			; IF NO.ERROR SKIP HLT
869	004412	104000			HLT				; CALL ERROR ROUTINE
870	004414					90\$:			
871	004414	022777	177775	173624	CMP	#177775, @MQ		; COMPARE MQ	
872	004422	001401			BEQ	91\$			; IF NO.ERROR SKIP HLT
873	004424	104000			HLT				; CALL ERROR ROUTINE
874	004426					91\$:			
875									
876	004426	104400			SCOPE				
877	004430	012777	125252	173610	MOV	#125252, @MQ		; SET MQ=125252	
878	004436	012777	177762	173614	MOV	#177762, @LSH		; SHIFT 177762 TIMES	
879	004444	122777	000001	173602	CMPB	#1, @SR		; CHECK STATUS - WITH CARRY	
880	004452	001401			BEQ	92\$			; IF NO.ERROR SKIP HLT
881	004454	104000			HLT				; CALL ERROR ROUTINE
882	004456					92\$:			
883	004456	022777	177776	173562	CMP	#177776, @MQ		; COMPARE MQ	
884	004464	001401			BEQ	93\$			; IF NO.ERROR SKIP HLT
885	004466	104000			HLT				; CALL ERROR ROUTINE
886	004470					93\$:			
887									
888	004470	104400			SCOPE				
889	004472	012777	125252	173546	MOV	#125252, @MQ		; SET MQ=125252	
890	004500	012777	177761	173552	MOV	#177761, @LSH		; SHIFT 177761 TIMES	
891	004506	105777	173542		TSTB	@SR		; CHECK STATUS - NO CARRY	
892	004512	001401			BEQ	94\$			; IF NO.ERROR SKIP HLT
893	004514	104000			HLT				; CALL ERROR ROUTINE
894	004516					94\$:			
895	004516	022777	177777	173522	CMP	#177777, @MQ		; COMPARE MQ	
896	004524	001401			BEQ	95\$			; IF NO.ERROR SKIP HLT
897	004526	104000			HLT				; CALL ERROR ROUTINE
898	004530					95\$:			

```

899 ;*****
900 ; TEST AC SHIFT LEFT OF ALTERNATE 1'S AND 0'S
901 ;*****
902 AC10L:
903 SCOPE
904 CLR @M0
905 MOV #125252,@AC ;SET AC=125252
906 CLP @LSH ;INITIALIZE SHIFT BY 0
907 CMP @AC,#125252 ;COMPARE AC
908 BEQ 64$ ;IF NO.ERROR SKIP HLT
909 HLT ;CALL ERROR ROUTINE
910 64$:
911 CMPB #310,@SR ;CHECK STATUS
912 BEQ 65$ ;IF NO.ERROR SKIP HLT
913 HLT ;CALL ERROR ROUTINE
914 65$:
915 SCOPE
916 MOV #125252,@AC ;SET AC=125252
917 MOV #1,@LSH ;SHIFT 1 TIMES
918 CMPB #211,@SR ;CHECK STATUS - WITH CARRY
919 BEQ 66$ ;IF NO.ERROR SKIP HLT
920 HLT ;CALL ERROR ROUTINE
921 66$:
922 CMP #52524,@AC ;COMPARE AC
923 BEQ 67$ ;IF NO.ERROR SKIP HLT
924 HLT ;CALL ERROR ROUTINE
925 67$:
926 SCOPE
927 MOV #125252,@AC ;SET AC=125252
928 MOV #2,@LSH ;SHIFT 2 TIMES
929 CMPB #110,@SR ;CHECK STATUS - NO CARRY
930 BEQ 68$ ;IF NO.ERROR SKIP HLT
931 HLT ;CALL ERROR ROUTINE
932 68$:
933 CMP #125250,@AC ;COMPARE AC
934 BEQ 69$ ;IF NO.ERROR SKIP HLT
935 HLT ;CALL ERROR ROUTINE
936 69$:
937 SCOPE
938 MOV #125252,@AC ;SET AC=125252
939 MOV #3,@LSH ;SHIFT 3 TIMES
940 CMPB #211,@SR ;CHECK STATUS - WITH CARRY
941 BEQ 70$ ;IF NO.ERROR SKIP HLT
942 HLT ;CALL ERROR ROUTINE
943 70$:
944 CMP #52520,@AC ;COMPARE AC
945

```

950	004736	001401			BEQ	715			; IF NO.ERROR SKIP HLT
951	004740	104000			HLT				; CALL ERROR ROUTINE
952	004742					715:			
953	004742	104400			SCOPE				
954	004744	012777	125252	173272	MOV	#125252, @AC			; SET AC=125252
955	004752	012777	000004	173300	MOV	#4, @LSH			; SHIFT 4 TIMES
956	004760	122777	000110	173266	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
957	004766	001401			BEQ	725			; IF NO.ERROR SKIP HLT
958	004770	104000			HLT				; CALL ERROR ROUTINE
959	004772					725:			
960	004772	022777	125240	173244	CMP	#125240, @AC			; COMPARE AC
961	005000	001401			BEQ	735			; IF NO.ERROR SKIP HLT
962	005002	104000			HLT				; CALL ERROR ROUTINE
963	005004					735:			
964									
965									
966	005004	104400			SCOPE				
967	005006	012777	125252	173230	MOV	#125252, @AC			; SET AC=125252
968	005014	012777	000005	173236	MOV	#5, @LSH			; SHIFT 5 TIMES
969	005022	122777	000211	173224	CMPB	#211, @SR			; CHECK STATUS - WITH CARRY
970	005030	001401			BEQ	745			; IF NO.ERROR SKIP HLT
971	005032	104000			HLT				; CALL ERROR ROUTINE
972	005034					745:			
973	005034	022777	052500	173202	CMP	#52500, @AC			; COMPARE AC
974	005042	001401			BEQ	755			; IF NO.ERROR SKIP HLT
975	005044	104000			HLT				; CALL ERROR ROUTINE
976	005046					755:			
977									
978									
979	005046	104400			SCOPE				
980	005050	012777	125252	173166	MOV	#125252, @AC			; SET AC=125252
981	005056	012777	000006	173174	MOV	#6, @LSH			; SHIFT 6 TIMES
982	005064	122777	000110	173162	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
983	005072	001401			BEQ	765			; IF NO.ERROR SKIP HLT
984	005074	104000			HLT				; CALL ERROR ROUTINE
985	005076					765:			
986	005076	022777	125200	173140	CMP	#125200, @AC			; COMPARE AC

987	005104	001401			BEQ	77\$			; IF NO.ERROR SKIP HLT
988	005106	104000			HLT				; CALL ERROR ROUTINE
989	005110				77\$:				
990									
991									
992	005110	104400			SCOPE				
993	005112	012777	125252	173124	MOV	#125252, @AC			; SET AC=125252
994	005120	012777	000007	173132	MOV	#7, @LSH			; SHIFT 7 TIMES
995	005126	122777	000211	173120	CMPB	#211, @SR			; CHECK STATUS - WITH CARRY
996	005134	001401			BEQ	78\$			; IF NO.ERROR SKIP HLT
997	005136	104000			HLT				; CALL ERROR ROUTINE
998	005140				78\$:				
999	005140	022777	052400	173076	CMP	#52400, @AC			; COMPARE AC
1000	005146	001401			BEQ	79\$			; IF NO.ERROR SKIP HLT
1001	005150	104000			HLT				; CALL ERROR ROUTINE
1002	005152				79\$:				
1003									
1004	005152	104400			SCOPE				
1005	005154	012777	125252	173062	MOV	#125252, @AC			; SET AC=125252
1006	005162	012777	000010	173070	MOV	#10, @LSH			; SHIFT 10 TIMES
1007	005170	122777	000110	173056	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
1008	005176	001401			BEQ	80\$			; IF NO.ERROR SKIP HLT
1009	005200	104000			HLT				; CALL ERROR ROUTINE
1010	005202				80\$:				
1011	005202	022777	125000	173034	CMP	#125000, @AC			; COMPARE AC
1012	005210	001401			BEQ	81\$			; IF NO.ERROR SKIP HLT
1013	005212	104000			HLT				; CALL ERROR ROUTINE
1014	005214				81\$:				
1015									
1016	005214	104400			SCOPE				
1017	005216	012777	125252	173020	MOV	#125252, @AC			; SET AC=125252
1018	005224	012777	000011	173026	MOV	#11, @LSH			; SHIFT 11 TIMES
1019	005232	122777	000211	173014	CMPB	#211, @SR			; CHECK STATUS - WITH CARRY
1020	005240	001401			BEQ	82\$			; IF NO.ERROR SKIP HLT
1021	005242	104000			HLT				; CALL ERROR ROUTINE
1022	005244				82\$:				
1023	005244	022777	052000	172772	CMP	#52000, @AC			; COMPARE AC
1024	005252	001401			BEQ	83\$			; IF NO.ERROR SKIP HLT
1025	005254	104000			HLT				; CALL ERROR ROUTINE
1026	005256				83\$:				
1027									
1028									
1029	005256	104400			SCOPE				
1030	005260	012777	125252	172756	MOV	#125252, @AC			; SET AC=125252
1031	005266	012777	000012	172764	MOV	#12, @LSH			; SHIFT 12 TIMES
1032	005274	122777	000110	172752	CMPB	#110, @SR			; CHECK STATUS - NO CARRY
1033	005302	001401			BEQ	84\$			; IF NO.ERROR SKIP HLT
1034	005304	104000			HLT				; CALL ERROR ROUTINE
1035	005306				84\$:				
1036	005306	022777	124000	172730	CMP	#124000, @AC			; COMPARE AC

1037	005314	001401				BEQ	85\$		; IF NO.ERROR SKIP HLT
1038	005316	104000				HLT			; CALL ERROR ROUTINE
1039	005320				85\$:				
1040									
1041									
1042	005320	104400				SCOPE			
1043	005322	012777	125252	172714		MOV	#125252, @AC		; SET AC=125252
1044	005330	012777	000013	172722		MOV	#13, @LSH		; SHIFT 13 TIMES
1045	005336	122777	000211	172710		CMPB	#211, @SR		; CHECK STATUS - WITH CARRY
1046	005344	001401				BEQ	86\$		; IF NO.ERROR SKIP HLT
1047	005346	104000				HLT			; CALL ERROR ROUTINE
1048	005350				86\$:				
1049	005350	022777	050000	172666		CMP	#50000, @AC		; COMPARE AC
1050	005356	001401				BEQ	87\$		; IF NO.ERROR SKIP HLT
1051	005360	104000				HLT			; CALL ERROR ROUTINE
1052	005362				87\$:				
1053									
1054	005362	104400				SCOPE			
1055	005364	012777	125252	172652		MOV	#125252, @AC		; SET AC=125252
1056	005372	012777	000014	172660		MOV	#14, @LSH		; SHIFT 14 TIMES
1057	005400	122777	000110	172646		CMPB	#110, @SR		; CHECK STATUS - NO CARRY
1058	005406	001401				BEQ	88\$		; IF NO.ERROR SKIP HLT
1059	005410	104000				HLT			; CALL ERROR ROUTINE
1060	005412				88\$:				
1061	005412	022777	120000	172624		CMP	#120000, @AC		; COMPARE AC
1062	005420	001401				BEQ	89\$		; IF NO.ERROR SKIP HLT
1063	005422	104000				HLT			; CALL ERROR ROUTINE
1064	005424				89\$:				
1065									
1066									
1067	005424	104400				SCOPE			
1068	005426	012777	125252	172610		MOV	#125252, @AC		; SET AC=125252
1069	005434	012777	000015	172616		MOV	#15, @LSH		; SHIFT 15 TIMES
1070	005442	122777	000211	172604		CMPB	#211, @SR		; CHECK STATUS - WITH CARRY
1071	005450	001401				BEQ	90\$		; IF NO.ERROR SKIP HLT
1072	005452	104000				HLT			; CALL ERROR ROUTINE
1073	005454				90\$:				
1074	005454	022777	040000	172562		CMP	#40000, @AC		; COMPARE AC
1075	005462	001401				BEQ	91\$		; IF NO.ERROR SKIP HLT
1076	005464	104000				HLT			; CALL ERROR ROUTINE
1077	005466				91\$:				
1078									
1079	005466	104400				SCOPE			
1080	005470	012777	125252	172546		MOV	#125252, @AC		; SET AC=125252
1081	005476	012777	000016	172554		MOV	#16, @LSH		; SHIFT 16 TIMES
1082	005504	122777	000110	172542		CMPB	#110, @SR		; CHECK STATUS - NO CARRY
1083	005512	001401				BEQ	92\$		; IF NO.ERROR SKIP HLT
1084	005514	104000				HLT			; CALL ERROR ROUTINE
1085	005516				92\$:				
1086	005516	022777	100000	172520		CMP	#100000, @AC		; COMPARE AC
1087	005524	001401				BEQ	93\$		; IF NO.ERROR SKIP HLT
1088	005526	104000				HLT			; CALL ERROR ROUTINE
1089	005530				93\$:				

```

1090      ;*****
1091      ;      TEST OF MQ SHIFT INTO AC
1092      ;*****
1093      MQAC:
1094
1095      005530 104400      SCOPE      ;TEST OF LOGICAL SHIFT
1096      005532 012777 000000 172506  MOV      #0, MQ      ;LOAD MQ WITH 0
1097      005540 012777 000000 172476  MOV      #0, AC      ;LOAD AC WITH 0
1098      005546 012777 000020 172504  MOV      #16, LSH   ;LOAD SHIFT COUNT (LSH) WITH 16.
1099      005554 022777 000000 172462  CMP      #0, AC      ;COMPARE AC WITH 0
1100      005562 001401      BEQ      64$        ;IF NO ERROR SKIP HLT
1101      005564 104000      HLT                    ;CALL ERROR ROUTINE
1102
1103      005566 022777 000000 172452  64$:  CMP      #0, MQ      ;COMPARE MQ WITH 0
1104      005574 001401      BEQ      65$        ;IF NO ERROR SKIP HLT
1105      005576 104000      HLT                    ;CALL ERROR ROUTINE
1106
1107      005600 122777 000036 172446  65$:  CMPB     #36, SR    ;COMPARE SR WITH 36
1108      005606 001401      BEQ      66$        ;IF NO ERROR SKIP HLT
1109      005610 104000      HLT                    ;CALL ERROR ROUTINE
1110      005612
1111
1112
1113
1114      005612 104400      SCOPE      ;TEST OF LOGICAL SHIFT
1115      005614 012777 177777 172424  MOV      #-1, MQ     ;LOAD MQ WITH -1
1116      005622 012777 000000 172414  MOV      #0, AC      ;LOAD AC WITH 0
1117      005630 012777 000020 172422  MOV      #16, LSH   ;LOAD SHIFT COUNT (LSH) WITH 16.
1118      005636 022777 177777 172400  CMP      #-1, AC    ;COMPARE AC WITH -1
1119      005644 001401      BEQ      67$        ;IF NO ERROR SKIP HLT
1120      005646 104000      HLT                    ;CALL ERROR ROUTINE
1121
1122      005650 022777 000000 172370  67$:  CMP      #0, MQ      ;COMPARE MQ WITH 0
1123      005656 001401      BEQ      68$        ;IF NO ERROR SKIP HLT
1124      005660 104000      HLT                    ;CALL ERROR ROUTINE
1125
1126      005662 122777 000150 172364  68$:  CMPB     #150, SR   ;COMPARE SR WITH 150
1127      005670 001401      BEQ      69$        ;IF NO ERROR SKIP HLT
1128      005672 104000      HLT                    ;CALL ERROR ROUTINE
1129      005674
1130
1131      005674 104400      SCOPE      ;TEST OF LOGICAL SHIFT
1132      005676 012777 177777 172342  MOV      #-1, MQ     ;LOAD MQ WITH -1
1133      005704 012777 000000 172332  MOV      #0, AC      ;LOAD AC WITH 0
1134      005712 012777 000037 172340  MOV      #31, LSH   ;LOAD SHIFT COUNT (LSH) WITH 31.
1135      005720 022777 100000 172316  CMP      #100000, AC ;COMPARE AC WITH 100000
1136      005726 001401      BEQ      70$        ;IF NO ERROR SKIP HLT
1137      005730 104000      HLT                    ;CALL ERROR ROUTINE
1138      005732
1139      005732 022777 000000 172306  70$:  CMP      #0, MQ      ;COMPARE MQ WITH 0
1140      005740 001401      BEQ      71$        ;IF NO ERROR SKIP HLT
1141      005742 104000      HLT                    ;CALL ERROR ROUTINE
1142
1143      005744 122777 000111 172302  71$:  CMPB     #111, SR   ;COMPARE SR WITH 111
1144      005752 001401      BEQ      72$        ;IF NO ERROR SKIP HLT
1145      005754 104000      HLT                    ;CALL ERROR ROUTINE

```

1146	005756				72\$:					
1147	005756	104400				SCOPE				; TEST OF LOGICAL SHIFT
1148	005760	012777	125252	172260		MOV	#125252, MQ			; LOAD MQ WITH 125252
1149	005766	012777	000000	172250		MOV	#0, AC			; LOAD AC WITH 0
1150	005774	012777	000020	172256		MOV	#16, LSH			; LOAD SHIFT COUNT (LSH) WITH 16.
1151	006002	022777	125252	172234		CMP	#125252, AC			; COMPARE AC WITH 125252
1152	006010	001401				BEQ	73\$			; IF NO. ERROR SKIP HLT
1153	006012	104000				HLT				; CALL ERROR ROUTINE
1154	006014				73\$:					
1155	006014	022777	000000	172224		CMP	#0, MQ			; COMPARE MQ WITH 0
1156	006022	001401				BEQ	74\$			; IF NO. ERROR SKIP HLT
1157	006024	104000				HLT				; CALL ERROR ROUTINE
1158	006026				74\$:					
1159	006026	122777	000110	172220		CMPB	#110, SR			; COMPARE SR WITH 110
1160	006034	001401				BEQ	75\$			; IF NO. ERROR SKIP HLT
1161	006036	104000				HLT				; CALL ERROR ROUTINE
1162	006040				75\$:					
1163										
1164										
1165										
1166	006040	104400				SCOPE				; TEST OF LOGICAL SHIFT
1167	006042	012777	125252	172176		MOV	#125252, MQ			; LOAD MQ WITH 125252
1168	006050	012777	000000	172166		MOV	#0, AC			; LOAD AC WITH 0
1169	006056	012777	000001	172174		MOV	#1, LSH			; LOAD SHIFT COUNT (LSH) WITH 1
1170	006064	022777	000001	172152		CMP	#1, AC			; COMPARE AC WITH 1
1171	006072	001401				BEQ	76\$			; IF NO. ERROR SKIP HLT
1172	006074	104000				HLT				; CALL ERROR ROUTINE
1173	006076				76\$:					
1174	006076	022777	052524	172142		CMP	#52524, MQ			; COMPARE MQ WITH 52524
1175	006104	001401				BEQ	77\$			; IF NO. ERROR SKIP HLT
1176	006106	104000				HLT				; CALL ERROR ROUTINE
1177	006110				77\$:					
1178	006110	122777	000000	172136		CMPB	#0, SR			; COMPARE SR WITH 0
1179	006116	001401				BEQ	78\$			; IF NO. ERROR SKIP HLT
1180	006120	104000				HLT				; CALL ERROR ROUTINE
1181	006122				78\$:					
1182										
1183										
1184	006122	104400				SCOPE				; TEST OF LOGICAL SHIFT
1185	006124	012777	000001	172114		MOV	#1, MQ			; LOAD MQ WITH 1
1186	006132	012777	000000	172104		MOV	#0, AC			; LOAD AC WITH 0
1187	006140	012777	000021	172112		MOV	#21, LSH			; LOAD SHIFT COUNT (LSH) WITH 21
1188	006146	022777	000002	172070		CMP	#2, AC			; COMPARE AC WITH 2
1189	006154	001401				BEQ	79\$			; IF NO. ERROR SKIP HLT
1190	006156	104000				HLT				; CALL ERROR ROUTINE
1191	006160				79\$:					
1192	006160	022777	000000	172060		CMP	#0, MQ			; COMPARE MQ WITH 0
1193	006166	001401				BEQ	80\$			; IF NO. ERROR SKIP HLT
1194	006170	104000				HLT				; CALL ERROR ROUTINE
1195	006172				80\$:					
1196	006172	122777	000010	172054		CMPB	#10, SR			; COMPARE SR WITH 10
1197	006200	001401				BEQ	81\$			; IF NO. ERROR SKIP HLT
1198	006202	104000				HLT				; CALL ERROR ROUTINE
1199	006204				81\$:					
1200										
1201	006204	104400				SCOPE				; TEST OF LOGICAL SHIFT



1202	006206	012777	000001	172032	MOV	#1,AMQ	;LOAD MQ WITH 1
1203	006214	012777	000000	172022	MOV	#0,AC	;LOAD AC WITH 0
1204	006222	012777	000022	172030	MOV	#22,ALSH	;LOAD SHIFT COUNT (LSH) WITH 22
1205	006230	022777	000004	172006	CMP	#4,AC	;COMPARE AC WITH 4
1206	006236	001401			BEQ	82\$	;IF NO.ERROR SKIP HLT
1207	006240	104000			HLT		;CALL ERROR ROUTINE
1208	006242					82\$:	
1209	006242	022777	000000	171776	CMP	#0,AMQ	;COMPARE MQ WITH 0
1210	006250	001401			BEQ	83\$	;IF NO.ERROR SKIP HLT
1211	006252	104000			HLT		;CALL ERROR ROUTINE
1212	006254					83\$:	
1213	006254	122777	000010	171772	CMPB	#10,ASR	;COMPARE SR WITH 10
1214	006262	001401			BEQ	84\$	;IF NO.ERROR SKIP HLT
1215	006264	104000			HLT		;CALL ERROR ROUTINE
1216	006266					84\$:	
1217							
1218							
1219	006266	104400			SCOPE		;TEST OF LOGICAL SHIFT
1220	006270	012777	000001	171750	MOV	#1,AMQ	;LOAD MQ WITH 1
1221	006276	012777	000000	171740	MOV	#0,AC	;LOAD AC WITH 0
1222	006304	012777	000023	171746	MOV	#23,ALSH	;LOAD SHIFT COUNT (LSH) WITH 23
1223	006312	022777	000010	171724	CMP	#10,AC	;COMPARE AC WITH 10
1224	006320	001401			BEQ	85\$	;IF NO.ERROR SKIP HLT
1225	006322	104000			HLT		;CALL ERROR ROUTINE
1226	006324					85\$:	
1227	006324	022777	000000	171714	CMP	#0,AMQ	;COMPARE MQ WITH 0
1228	006332	001401			BEQ	86\$	;IF NO.ERROR SKIP HLT
1229	006334	104000			HLT		;CALL ERROR ROUTINE
1230	006336					86\$:	
1231	006336	122777	000010	171710	CMPB	#10,ASR	;COMPARE SR WITH 10
1232	006344	001401			BEQ	87\$	;IF NO.ERROR SKIP HLT
1233	006346	104000			HLT		;CALL ERROR ROUTINE
1234	006350					87\$:	
1235							
1236							
1237	006350	104400			SCOPE		;TEST OF LOGICAL SHIFT
1238	006352	012777	000001	171666	MOV	#1,AMQ	;LOAD MQ WITH 1
1239	006360	012777	000000	171656	MOV	#0,AC	;LOAD AC WITH 0
1240	006366	012777	000024	171664	MOV	#24,ALSH	;LOAD SHIFT COUNT (LSH) WITH 24
1241	006374	022777	000020	171642	CMP	#20,AC	;COMPARE AC WITH 20
1242	006402	001401			BEQ	88\$	;IF NO.ERROR SKIP HLT
1243	006404	104000			HLT		;CALL ERROR ROUTINE
1244	006406					88\$:	
1245	006406	022777	000000	171632	CMP	#0,AMQ	;COMPARE MQ WITH 0
1246	006414	001401			BEQ	89\$	;IF NO.ERROR SKIP HLT
1247	006416	104000			HLT		;CALL ERROR ROUTINE
1248	006420					89\$:	
1249	006420	122777	000010	171626	CMPB	#10,ASR	;COMPARE SR WITH 10
1250	006426	001401			BEQ	90\$	;IF NO.ERROR SKIP HLT
1251	006430	104000			HLT		;CALL ERROR ROUTINE
1252	006432					90\$:	
1253							
1254	006432	104400			SCOPE		;TEST OF LOGICAL SHIFT
1255	006434	012777	000001	171604	MOV	#1,AMQ	;LOAD MQ WITH 1
1256	006442	012777	000000	171574	MOV	#0,AC	;LOAD AC WITH 0
1257	006450	012777	000025	171602	MOV	#25,ALSH	;LOAD SHIFT COUNT (LSH) WITH 25

1258	006456	022777	000040	171560		CMP	#40, @AC		: COMPARE AC WITH 40
1259	006464	001401				BEQ	91\$		: IF NO. ERROR SKIP HLT
1260	006466	104000				HLT			: CALL ERROR ROUTINE
1261	006470				91\$:				
1262	006470	022777	000000	171550		CMP	#0, @MQ		: COMPARE MQ WITH 0
1263	006476	001401				BEQ	92\$		: IF NO. ERROR SKIP HLT
1264	006500	104000				HLT			: CALL ERROR ROUTINE
1265	006502				92\$:				
1266	006502	122777	000010	171544		CMPB	#10, @SR		: COMPARE SR WITH 10
1267	006510	001401				BEQ	93\$		: IF NO. ERROR SKIP HLT
1268	006512	104000				HLT			: CALL ERROR ROUTINE
1269	006514				93\$:				
1270									
1271									
1272									
1273	006514	104400				SCOPE			: TEST OF LOGICAL SHIFT
1274	006516	012777	000001	171522		MOV	#1, @MQ		: LOAD MQ WITH 1
1275	006524	012777	000000	171512		MOV	#0, @AC		: LOAD AC WITH 0
1276	006532	012777	000026	171520		MOV	#26, @LSH		: LOAD SHIFT COUNT (LSH) WITH 26
1277	006540	022777	000100	171476		CMP	#100, @AC		: COMPARE AC WITH 100
1278	006546	001401				BEQ	94\$		: IF NO. ERROR SKIP HLT
1279	006550	104000				HLT			: CALL ERROR ROUTINE
1280	006552				94\$:				
1281	006552	022777	000000	171466		CMP	#0, @MQ		: COMPARE MQ WITH 0
1282	006560	001401				BEQ	95\$		: IF NO. ERROR SKIP HLT
1283	006562	104000				HLT			: CALL ERROR ROUTINE
1284	006564				95\$:				
1285	006564	122777	000010	171462		CMPB	#10, @SR		: COMPARE SR WITH 10
1286	006572	001401				BEQ	96\$		: IF NO. ERROR SKIP HLT
1287	006574	104000				HLT			: CALL ERROR ROUTINE
1288	006576				96\$:				
1289									
1290									
1291	006576	104400				SCOPE			: TEST OF LOGICAL SHIFT
1292	006600	012777	000001	171440		MOV	#1, @MQ		: LOAD MQ WITH 1
1293	006606	012777	000000	171430		MOV	#0, @AC		: LOAD AC WITH 0
1294	006614	012777	000027	171436		MOV	#27, @LSH		: LOAD SHIFT COUNT (LSH) WITH 27
1295	006622	022777	000200	171414		CMP	#200, @AC		: COMPARE AC WITH 200
1296	006630	001401				BEQ	97\$		: IF NO. ERROR SKIP HLT
1297	006632	104000				HLT			: CALL ERROR ROUTINE
1298	006634				97\$:				
1299	006634	022777	000000	171404		CMP	#0, @MQ		: COMPARE MQ WITH 0
1300	006642	001401				BEQ	98\$		: IF NO. ERROR SKIP HLT
1301	006644	104000				HLT			: CALL ERROR ROUTINE
1302	006646				98\$:				
1303	006646	122777	000010	171400		CMPB	#10, @SR		: COMPARE SR WITH 10
1304	006654	001401				BEQ	99\$		: IF NO. ERROR SKIP HLT
1305	006656	104000				HLT			: CALL ERROR ROUTINE
1306	006660				99\$:				
1307	006660	104400				SCOPE			: TEST OF LOGICAL SHIFT
1308	006662	012777	000001	171356		MOV	#1, @MQ		: LOAD MQ WITH 1
1309	006670	012777	000000	171346		MOV	#0, @AC		: LOAD AC WITH 0
1310	006676	012777	000030	171354		MOV	#30, @LSH		: LOAD SHIFT COUNT (LSH) WITH 30
1311	006704	022777	000400	171332		CMP	#400, @AC		: COMPARE AC WITH 400
1312	006712	01401				BEQ	100\$		: IF NO. ERROR SKIP HLT
1313	006714	104000				HLT			: CALL ERROR ROUTINE

1314	006716				100\$:						
1315	006716	022777	000000	171322		CMP	#0,2MQ			:COMPARE MQ WITH 0	
1316	006724	001401				BEQ	101\$			:IF NO.ERROR SKIP HLT	
1317	006726	104000				HLT				:CALL ERROR ROUTINE	
1318	006730				101\$:						
1319	006730	122777	000010	171316		CMPB	#10,2SR			:COMPARE SR WITH 10	
1320	006736	001401				BEQ	102\$			:IF NO.ERROR SKIP HLT	
1321	006740	104000				HLT				:CALL ERROR ROUTINE	
1322	006742				102\$:						
1323	006742	104400				SCOPE				:TEST OF LOGICAL SHIFT	
1324	006744	012777	000001	171274		MOV	#1,2MQ			:LOAD MQ WITH 1	
1325	006752	012777	000000	171264		MOV	#0,2AC			:LOAD AC WITH 0	
1326	006760	012777	000031	171272		MOV	#31,2LSH			:LOAD SHIFT COUNT (LSH) WITH 31	
1327	006766	022777	001000	171250		CMP	#1000,2AC			:COMPARE AC WITH 1000	
1328	006774	001401				BEQ	103\$			:IF NO.ERROR SKIP HLT	
1329	006776	104000				HLT				:CALL ERROR ROUTINE	
1330	007000				103\$:						
1331	007000	022777	000000	171240		CMP	#0,2MQ			:COMPARE MQ WITH 0	
1332	007006	001401				BEQ	104\$			:IF NO.ERROR SKIP HLT	
1333	007010	104000				HLT				:CALL ERROR ROUTINE	
1334	007012				104\$:						
1335	007012	122777	000010	171234		CMPB	#10,2SR			:COMPARE SR WITH 10	
1336	007020	001401				BEQ	105\$			:IF NO.ERROR SKIP HLT	
1337	007022	104000				HLT				:CALL ERROR ROUTINE	
1338	007024				105\$:						
1339	007024	104400				SCOPE				:TEST OF LOGICAL SHIFT	
1340	007026	012777	000001	171212		MOV	#1,2MQ			:LOAD MQ WITH 1	
1341	007034	012777	000000	171202		MOV	#0,2AC			:LOAD AC WITH 0	
1342	007042	012777	000032	171210		MOV	#32,2LSH			:LOAD SHIFT COUNT (LSH) WITH 32	
1343	007050	022777	002000	171166		CMP	#2000,2AC			:COMPARE AC WITH 2000	
1344	007056	001401				BEQ	106\$			:IF NO.ERROR SKIP HLT	
1345	007060	104000				HLT				:CALL ERROR ROUTINE	
1346	007062				106\$:						
1347	007062	022777	000000	171156		CMP	#0,2MQ			:COMPARE MQ WITH 0	
1348	007070	001401				BEQ	107\$			:IF NO.ERROR SKIP HLT	
1349	007072	104000				HLT				:CALL ERROR ROUTINE	
1350	007074				107\$:						
1351	007074	122777	000010	171152		CMPB	#10,2SR			:COMPARE SR WITH 10	
1352	007102	001401				BEQ	108\$			:IF NO.ERROR SKIP HLT	
1353	007104	104000				HLT				:CALL ERROR ROUTINE	
1354	007106				108\$:						
1355	007106	104400				SCOPE				:TEST OF LOGICAL SHIFT	
1356	007110	012777	000001	171130		MOV	#1,2MQ			:LOAD MQ WITH 1	
1357	007116	012777	000000	171120		MOV	#0,2AC			:LOAD AC WITH 0	
1358	007124	012777	000033	171126		MOV	#33,2LSH			:LOAD SHIFT COUNT (LSH) WITH 33	
1359	007132	022777	004000	171104		CMP	#4000,2AC			:COMPARE AC WITH 4000	
1360	007140	001401				BEQ	109\$			:IF NO.ERROR SKIP HLT	
1361	007142	104000				HLT				:CALL ERROR ROUTINE	
1362	007144				109\$:						
1363	007144	022777	000000	171074		CMP	#0,2MQ			:COMPARE MQ WITH 0	
1364	007152	001401				BEQ	110\$			:IF NO.ERROR SKIP HLT	
1365	007154	104000				HLT				:CALL ERROR ROUTINE	
1366	007156				110\$:						
1367	007156	122777	000010	171070		CMPB	#10,2SR			:COMPARE SR WITH 10	
1368	007164	001401				BEQ	111\$			:IF NO.ERROR SKIP HLT	
1369	007166	104000				HLT				:CALL ERROR ROUTINE	

1370	007170				111\$:			
1371								
1372								
1373	007170	104400				SCOPE		; TEST OF LOGICAL SHIFT
1374	007172	012777	000001	171046		MOV	#1, @MQ	; LOAD MQ WITH 1
1375	007200	012777	000000	171036		MOV	#0, @AC	; LOAD AC WITH 0
1376	007206	012777	000034	171044		MOV	#34, @LSH	; LOAD SHIFT COUNT (LSH) WITH 34
1377	007214	022777	010000	171022		CMP	#10000, @AC	; COMPARE AC WITH 10000
1378	007222	001401				BEQ	112\$	; IF NO. ERROR SKIP HLT
1379	007224	104000				HLT		; CALL ERROR ROUTINE
1380	007226				112\$:			
1381	007226	022777	000000	171012		CMP	#0, @MQ	; COMPARE MQ WITH 0
1382	007234	001401				BEQ	113\$	; IF NO. ERROR SKIP HLT
1383	007236	104000				HLT		; CALL ERROR ROUTINE
1384	007240				113\$:			
1385	007240	122777	000010	171006		CMPB	#10, @SR	; COMPARE SR WITH 10
1386	007246	001401				BEQ	114\$	; IF NO. ERROR SKIP HLT
1387	007250	104000				HLT		; CALL ERROR ROUTINE
1388	007252				114\$:			
1389								
1390								
1391	007252	104400				SCOPE		; TEST OF LOGICAL SHIFT
1392	007254	012777	000001	170764		MOV	#1, @MQ	; LOAD MQ WITH 1
1393	007262	012777	000000	170754		MOV	#0, @AC	; LOAD AC WITH 0
1394	007270	012777	000035	170762		MOV	#35, @LSH	; LOAD SHIFT COUNT (LSH) WITH 35
1395	007276	022777	020000	170740		CMP	#20000, @AC	; COMPARE AC WITH 20000
1396	007304	001401				BEQ	115\$	; IF NO. ERROR SKIP HLT
1397	007306	104000				HLT		; CALL ERROR ROUTINE
1398	007310				115\$:			
1399	007310	022777	000000	170730		CMP	#0, @MQ	; COMPARE MQ WITH 0
1400	007316	001401				BEQ	116\$	; IF NO. ERROR SKIP HLT
1401	007320	104000				HLT		; CALL ERROR ROUTINE
1402	007322				116\$:			
1403	007322	122777	000010	170724		CMPB	#10, @SR	; COMPARE SR WITH 10
1404	007330	001401				BEQ	117\$	; IF NO. ERROR SKIP HLT
1405	007332	104000				HLT		; CALL ERROR ROUTINE
1406	007334				117\$:			
1407	007334	104400				SCOPE		; TEST OF LOGICAL SHIFT
1408	007336	012777	000001	170702		MOV	#1, @MQ	; LOAD MQ WITH 1
1409	007344	012777	000000	170672		MOV	#0, @AC	; LOAD AC WITH 0
1410	007352	012777	000036	170700		MOV	#36, @LSH	; LOAD SHIFT COUNT (LSH) WITH 36
1411	007360	022777	040000	170656		CMP	#40000, @AC	; COMPARE AC WITH 40000
1412	007366	001401				BEQ	118\$	; IF NO. ERROR SKIP HLT
1413	007370	104000				HLT		; CALL ERROR ROUTINE
1414	007372				118\$:			
1415	007372	022777	000000	170646		CMP	#0, @MQ	; COMPARE MQ WITH 0
1416	007400	001401				BEQ	119\$	; IF NO. ERROR SKIP HLT
1417	007402	104000				HLT		; CALL ERROR ROUTINE
1418	007404				119\$:			
1419	007404	122777	000010	170642		CMPB	#10, @SR	; COMPARE SR WITH 10
1420	007412	001401				BEQ	120\$	; IF NO. ERROR SKIP HLT
1421	007414	104000				HLT		; CALL ERROR ROUTINE
1422	007416				120\$:			

```

1423
1424
1425
1426 007416
1427
1428
1429 007416 104400
1430 007420 012777 000000 170620
1431 007426 012777 000000 170610
1432 007434 012777 177760 170616
1433 007442 022777 000000 170574
1434 007450 001401
1435 007452 104000
1436 007454
1437 007454 022777 000000 170564
1438 007462 001401
1439 007464 104000
1440 007466
1441 007466 122777 000036 170560
1442 007474 001401
1443 007476 104000
1444 007500
1445
1446
1447 007500 104400
1448 007502 012777 000000 170536
1449 007510 012777 177777 170526
1450 007516 012777 177760 170534
1451 007524 022777 000000 170512
1452 007532 001401
1453 007534 104000
1454 007536
1455 007536 022777 177777 170502
1456 007544 001401
1457 007546 104000
1458 007550
1459 007550 122777 000020 170476
1460 007556 001401
1461 007560 104000
1462 007562
1463 007562 104400
1464 007564 012777 000000 170454
1465 007572 012777 177777 170444
1466 007600 012777 177741 170452
1467 007606 022777 000000 170430
1468 007614 001401
1469 007616 104000
1470 007620
1471 007620 022777 000001 170420

```

;\*\*\*\*\*  
; TEST OF AC SHIFT INTO MQ  
;\*\*\*\*\*  
ACMQ:

```

SCOPE ; TEST OF LOGICAL SHIFT
MOV #0,AMQ ; LOAD MQ WITH 0
MOV #0,AC ; LOAD AC WITH 0
MOV #-16,ALSH ; LOAD SHIFT COUNT (LSH) 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$:
SCOPE ; TEST OF LOGICAL SHIFT
MOV #0,AMQ ; LOAD MQ WITH 0
MOV #-1,AC ; LOAD AC WITH -1
MOV #-16,ALSH ; LOAD SHIFT COUNT (LSH) WITH -16.
CMP #0,AC ; COMPARE AC WITH 0
BEQ 67$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE

67$:
CMP #-1,AMQ ; COMPARE MQ WITH -1
BEQ 68$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE

68$:
CMPB #20,SR ; COMPARE SR WITH 20
BEQ 69$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE

69$:
SCOPE ; TEST OF LOGICAL SHIFT
MOV #0,AMQ ; LOAD MQ WITH 0
MOV #-1,AC ; LOAD AC WITH -1
MOV #-31,ALSH ; LOAD SHIFT COUNT (LSH) WITH -31.
CMP #0,AC ; COMPARE AC WITH 0
BEQ 70$ ; IF NO.ERROR SKIP HLT
HLT ; CALL ERROR ROUTINE

70$:
CMP #1,AMQ ; COMPARE MQ WITH 1

```

1472	007626	001401			BEQ	71\$		: IF NO.ERROR SKIP HLT
1473	007630	104000			HLT			: CALL ERROR ROUTINE
1474	007632						71\$:	
1475	007632	122777	000023	170414	CMPB	#23, @SR		: COMPARE SR WITH 23
1476	007640	001401			BEQ	72\$		: IF NO.ERROR SKIP HLT
1477	007642	104000			HLT			: CALL ERROR ROUTINE
1478	007644						72\$:	
1479								
1480								
1481								
1482	007644	104400			SCOPE			: TEST OF LOGICAL SHIFT
1483	007646	012777	000000	170372	MOV	#0, @MQ		: LOAD MQ WITH 0
1484	007654	012777	125252	170362	MOV	#125252, @AC		: LOAD AC WITH 125252
1485	007662	012777	177760	170370	MOV	#-16, @LSH		: LOAD SHIFT COUNT (LSH) WITH -16.
1486	007670	022777	000000	170346	CMP	#0, @AC		: COMPARE AC WITH 0
1487	007676	001401			BEQ	73\$		: IF NO.ERROR SKIP HLT
1488	007700	104000			HLT			: CALL ERROR ROUTINE
1489	007702						73\$:	
1490	007702	012777	125252	170336	CMP	#125252, @MQ		: COMPARE MQ WITH 125252
1491	007710	001401			BEQ	74\$		: IF NO.ERROR SKIP HLT
1492	007712	104000			HLT			: CALL ERROR ROUTINE
1493	007714						74\$:	
1494	007714	122777	000020	170332	CMPB	#20, @SR		: COMPARE SR WITH 20
1495	007722	001401			BEQ	75\$		: IF NO.ERROR SKIP HLT
1496	007724	104000			HLT			: CALL ERROR ROUTINE
1497	007726						75\$:	
1498								
1499								
1500								
1501	007726	104400			SCOPE			: TEST OF LOGICAL SHIFT
1502	007730	012777	000000	170310	MOV	#0, @MQ		: LOAD MQ WITH 0
1503	007736	012777	052525	170300	MOV	#52525, @AC		: LOAD AC WITH 52525
1504	007744	012777	177777	170306	MOV	#-1, @LSH		: LOAD SHIFT COUNT (LSH) WITH -1
1505	007752	022777	025252	170264	CMP	#25252, @AC		: COMPARE AC WITH 25252
1506	007760	001401			BEQ	76\$		: IF NO.ERROR SKIP HLT
1507	007762	104000			HLT			: CALL ERROR ROUTINE
1508	007764						76\$:	
1509	007764	022777	100000	170254	CMP	#100000, @MQ		: COMPARE MQ WITH 100000
1510	007772	001401			BEQ	77\$		: IF NO.ERROR SKIP HLT
1511	007774	104000			HLT			: CALL ERROR ROUTINE
1512	007776						77\$:	
1513	007776	122777	000000	170250	CMPB	#0, @SR		: COMPARE SR WITH 0

1514	010004	001401			BEQ	78\$		; IF NO.ERROR SKIP HLT
1515	010006	104000			HLT			;CALL ERROR ROUTINE
1516	010010					78\$:		
1517								
1518	010010	104400			SCOPE			; TEST OF LOGICAL SHIFT
1519	010012	012777	000000	170226	MOV	#0,AMQ		;LOAD MQ WITH 0
1520	010020	012777	100000	170216	MOV	#100000,AC		;LOAD AC WITH 100000
1521	010026	012777	177742	170224	MOV	#177742,LSH		;LOAD SHIFT COUNT (LSH) WITH 177742
1522	010034	022777	000000	170202	CMP	#0,AC		;COMPARE AC WITH 0
1523	010042	001401			BEQ	79\$		; IF NO.ERROR SKIP HLT
1524	010044	104000			HLT			;CALL ERROR ROUTINE
1525	010046					79\$:		
1526	010046	022777	000002	170172	CMP	#2,AMQ		;COMPARE MQ WITH 2
1527	010054	001401			BEQ	80\$		; IF NO.ERROR SKIP HLT
1528	010056	104000			HLT			;CALL ERROR ROUTINE
1529	010060					80\$:		
1530	010060	122777	000022	170166	CMPB	#22,SR		;COMPARE SR WITH 22
1531	010066	001401			BEQ	81\$		; IF NO.ERROR SKIP HLT
1532	010070	104000			HLT			;CALL ERROR ROUTINE
1533	010072					81\$:		
1534								
1535	010072	104400			SCOPE			; TEST OF LOGICAL SHIFT
1536	010074	012777	000000	170144	MOV	#0,AMQ		;LOAD MQ WITH 0
1537	010102	012777	100000	170134	MOV	#100000,AC		;LOAD AC WITH 100000
1538	010110	012777	177743	170142	MOV	#177743,LSH		;LOAD SHIFT COUNT (LSH) WITH 177743
1539	010115	022777	000000	170120	CMP	#0,AC		;COMPARE AC WITH 0
1540	010121	001401			BEQ	82\$		; IF NO.ERROR SKIP HLT
1541	010126	104000			HLT			;CALL ERROR ROUTINE
1542	010130					82\$:		
1543	010130	022777	000004	170110	CMP	#4,AMQ		;COMPARE MQ WITH 4
1544	010136	001401			BEQ	83\$		; IF NO.ERROR SKIP HLT
1545	010140	104000			HLT			;CALL ERROR ROUTINE
1546	010142					83\$:		
1547	010142	122777	000022	170104	CMPB	#22,SR		;COMPARE SR WITH 22
1548	010150	001401			BEQ	84\$		; IF NO.ERROR SKIP HLT
1549	010152	104000			HLT			;CALL ERROR ROUTINE
1550	010154					84\$:		
1551								
1552								
1553	010154	104400			SCOPE			; TEST OF LOGICAL SHIFT
1554	010156	012777	000000	170062	MOV	#0,AMQ		;LOAD MQ WITH 0
1555	010164	012777	100000	170052	MOV	#100000,AC		;LOAD AC WITH 100000
1556	010172	012777	177744	170060	MOV	#177744,LSH		;LOAD SHIFT COUNT (LSH) WITH 177744
1557	010200	022777	000000	170036	CMP	#0,AC		;COMPARE AC WITH 0
1558	010206	001401			BEQ	85\$		; IF NO.ERROR SKIP HLT
1559	010210	104000			HLT			;CALL ERROR ROUTINE
1560	010212					85\$:		
1561	010212	022777	000010	170026	CMP	#10,AMQ		;COMPARE MQ WITH 10
1562	010220	001401			BEQ	86\$		; IF NO.ERROR SKIP HLT
1563	010222	104000			HLT			;CALL ERROR ROUTINE
1564	010224					86\$:		
1565	010224	122777	000022	170022	CMPB	#22,SR		;COMPARE SR WITH 22

1566	010232	001401			BEQ	87\$			; IF NO.ERROR SKIP HLT
1567	010234	104000			HLT				; CALL ERROR ROUTINE
1568	010236				87\$:				
1569	010236	104400			SCOPE				; TEST OF LOGICAL SHIFT
1570	010240	012777	000000	170000	MOV	#0,2MQ			; LOAD MQ WITH 0
1571	010246	012777	100000	167770	MOV	#100000,2AC			; LOAD AC WITH 100000
1572	010254	012777	177745	167776	MOV	#177745,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177745
1573	010262	022777	000000	167754	CMP	#0,2AC			; COMPARE AC WITH 0
1574	010270	001401			BEQ	88\$			; IF NO.ERROR SKIP HLT
1575	010272	104000			HLT				; CALL ERROR ROUTINE
1576	010274				88\$:				
1577	010274	022777	000020	167744	CMP	#20,2MQ			; COMPARE MQ WITH 20
1578	010302	001401			BEQ	89\$			; IF NO.ERROR SKIP HLT
1579	010304	104000			HLT				; CALL ERROR ROUTINE
1580	010307				89\$:				
1581	010306	122777	000022	167740	CMPB	#22,2SR			; COMPARE SR WITH 22
1582	010314	001401			BEQ	90\$			; IF NO.ERROR SKIP HLT
1583	010316	104000			HLT				; CALL ERROR ROUTINE
1584	010320				90\$:				
1585									
1586									
1587	010320	104400			SCOPE				; TEST OF LOGICAL SHIFT
1588	010322	012777	000000	167716	MOV	#0,2MQ			; LOAD MQ WITH 0
1589	010330	012777	100000	167706	MOV	#100000,2AC			; LOAD AC WITH 100000
1590	010333	012777	177746	167714	MOV	#177746,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177746
1591	010344	022777	000000	167672	CMP	#0,2AC			; COMPARE AC WITH 0
1592	010352	001401			BEQ	91\$			; IF NO.ERROR SKIP HLT
1593	010354	104000			HLT				; CALL ERROR ROUTINE
1594	010356				91\$:				
1595	010356	022777	000040	167662	CMP	#40,2MQ			; COMPARE MQ WITH 40
1596	010364	001401			BEQ	92\$			; IF NO.ERROR SKIP HLT
1597	010366	104000			HLT				; CALL ERROR ROUTINE
1598	010370				92\$:				
1599	010370	122777	000022	167656	CMPB	#22,2SR			; COMPARE SR WITH 22
1600	010376	001401			BEQ	93\$			; IF NO.ERROR SKIP HLT
1601	010400	104000			HLT				; CALL ERROR ROUTINE
1602	010402				93\$:				
1603									
1604									
1605	010402	104400			SCOPE				; TEST OF LOGICAL SHIFT
1606	010404	012777	000000	167634	MOV	#0,2MQ			; LOAD MQ WITH 0
1607	010412	012777	100000	167624	MOV	#100000,2AC			; LOAD AC WITH 100000
1608	010420	012777	177747	167632	MOV	#177747,2LSH			; LOAD SHIFT COUNT (LSH) WITH 177747
1609	010426	022777	000000	167610	CMP	#0,2AC			; COMPARE AC WITH 0
1610	010434	001401			BEQ	94\$			; IF NO.ERROR SKIP HLT
1611	010436	104000			HLT				; CALL ERROR ROUTINE
1612	010440				94\$:				
1613	010440	022777	000100	167600	CMP	#100,2MQ			; COMPARE MQ WITH 100
1614	010446	001401			BEQ	95\$			; IF NO.ERROR SKIP HLT
1615	010450	104000			HLT				; CALL ERROR ROUTINE
1616	010452				95\$:				
1617	010452	122777	000022	167574	CMPB	#22,2SR			; COMPARE SR WITH 22



1618	010460	001401				BEQ	96\$		; IF NO.ERROR SKIP HLT
1619	010462	104000				HLT			; CALL ERROR ROUTINE
1620	010464				96\$:				
1621	010464	104400				SCOPE			; TEST OF LOGICAL SHIFT
1622	010466	012777	000000	167552		MOV	#0, @MQ		; LOAD MQ WITH 0
1623	010474	012777	100000	167542		MOV	#100000, @AC		; LOAD AC WITH 100000
1624	010502	012777	177750	167550		MOV	#177750, @LSH		; LOAD SHIFT COUNT (LSH) WITH 177750
1625	010510	022777	000000	167526		CMP	#0, @AC		; COMPARE AC WITH 0
1626	010516	001401				BEQ	97\$		; IF NO.ERROR SKIP HLT
1627	010520	104000				HLT			; CALL ERROR ROUTINE
1628	010522				97\$:				
1629	010522	022777	000200	167516		CMP	#200, @MQ		; COMPARE MQ WITH 200
1630	010530	001401				BEQ	98\$		; IF NO.ERROR SKIP HLT
1631	010532	104000				HLT			; CALL ERROR ROUTINE
1632	010534				98\$:				
1633	010534	122777	000022	167512		CMPB	#22, @SR		; COMPARE SR WITH 22
1634	010542	001401				BEQ	99\$		; IF NO.ERROR SKIP HLT
1635	010544	104000				HLT			; CALL ERROR ROUTINE
1636	010546				99\$:				
1637									
1638									
1639	010546	104400				SCOPE			; TEST OF LOGICAL SHIFT
1640	010550	012777	000000	167470		MOV	#0, @MQ		; LOAD MQ WITH 0
1641	010556	012777	100000	167460		MOV	#100000, @AC		; LOAD AC WITH 100000
1642	010564	012777	177751	167466		MOV	#177751, @LSH		; LOAD SHIFT COUNT (LSH) WITH 177751
1643	010572	022777	000000	167444		CMP	#0, @AC		; COMPARE AC WITH 0
1644	010600	001401				BEQ	100\$		; IF NO.ERROR SKIP HLT
1645	010602	104000				HLT			; CALL ERROR ROUTINE
1646	010604				100\$:				
1647	010604	022777	000400	167434		CMP	#400, @MQ		; COMPARE MQ WITH 400
1648	010612	001401				BEQ	101\$		; IF NO.ERROR SKIP HLT
1649	010614	104000				HLT			; CALL ERROR ROUTINE
1650	010616				101\$:				
1651	010616	122777	000022	167430		CMPB	#22, @SR		; COMPARE SR WITH 22
1652	010624	001401				BEQ	102\$		; IF NO.ERROR SKIP HLT
1653	010626	104000				HLT			; CALL ERROR ROUTINE
1654	010630				102\$:				
1655									
1656									
1657	010630	104400				SCOPE			; TEST OF LOGICAL SHIFT
1658	010632	012777	000000	167406		MOV	#0, @MQ		; LOAD MQ WITH 0
1659	010640	012777	100000	167376		MOV	#100000, @AC		; LOAD AC WITH 100000
1660	010646	012777	177752	167404		MOV	#177752, @LSH		; LOAD SHIFT COUNT (LSH) WITH 177752
1661	010654	022777	000000	167362		CMP	#0, @AC		; COMPARE AC WITH 0
1662	010662	001401				BEQ	103\$		; IF NO.ERROR SKIP HLT
1663	010664	104000				HLT			; CALL ERROR ROUTINE
1664	010666				103\$:				
1665	010666	022777	001000	167352		CMP	#1000, @MQ		; COMPARE MQ WITH 1000
1666	010674	001401				BEQ	104\$		; IF NO.ERROR SKIP HLT
1667	010676	104000				HLT			; CALL ERROR ROUTINE
1668	010700				104\$:				
1669	010700	122777	000022	167346		CMPB	#22, @SR		; COMPARE SR WITH 22



1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781

011140

011140 104400  
011142 012777 000000 167076  
011150 012777 000000 167066  
011156 012777 177760 167076  
011164 022777 000000 167052  
011172 001401  
011174 104000  
011176  
011176 022777 000000 167042  
011204 001401  
011206 104000  
011210  
011210 122777 000036 167036  
011216 001401  
011220 104000  
011222  
011222 104400  
011224 012777 000000 167014  
011232 012777 177777 167004  
011240 012777 177760 167014  
011246 022777 177777 166770  
011254 001401  
011256 104000  
011260  
011260 022777 177777 166760  
011266 001401  
011270 104000  
011272  
011272 122777 000342 166754  
011300 001401  
011302 104000  
011304  
011304 104400  
011306 012777 000000 166732  
011314 012777 125252 166722  
011322 012777 177760 166732  
011330 022777 177777 166706  
011336 001401  
011340 104000

\*\*\*\*\*  
: AT THIS POINT, THE LOGICAL SHIFT WORKS  
:\*\*\*\*\*  
:\*\*\*\*\*  
: TEST OF ARITHMETIC SHIFT  
:\*\*\*\*\*  
: SHIFT RIGHT  
:\*\*\*\*\*  
ASR:

SCOPE : TEST OF ARITHMETIC SHIFT  
MOV #0, @MQ : 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, @MQ : 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 #0, @MQ : LOAD MQ WITH 0  
MOV #-1, @AC : LOAD AC WITH -1  
MOV #-16, @ASH : LOAD SHIFT COUNT (ASH) WITH -16.  
CMP #-1, @AC : COMPARE AC WITH -1  
BEQ 67\$ : IF NO. ERROR SKIP HLT  
HLT : CALL ERROR ROUTINE  
67\$:  
CMP #-1, @MQ : COMPARE MQ WITH -1  
BEQ 68\$ : IF NO. ERROR SKIP HLT  
HLT : CALL ERROR ROUTINE  
68\$:  
CMPB #342, @SR : COMPARE SR WITH 342  
BEQ 69\$ : IF NO. ERROR SKIP HLT  
HLT : CALL ERROR ROUTINE  
69\$:  
SCOPE : TEST OF ARITHMETIC SHIFT  
MOV #0, @MQ : LOAD MQ WITH 0  
MOV #125252, @AC : LOAD AC WITH 125252  
MOV #-16, @ASH : LOAD SHIFT COUNT (ASH) WITH -16.  
CMP #-1, @AC : COMPARE AC WITH -1  
BEQ 70\$ : IF NO. ERROR SKIP HLT  
HLT : CALL ERROR ROUTINE



```

1828      ;:*****
1829      ;:      SHIFT LEFT
1830      ;:*****
1831      ASL:
1832      SCOPE      ;:TEST OF ARITHMETIC SHIFT
1833      MOV      #0,AMQ      ;:LOAD MQ WITH 0
1834      MOV      #0,AC      ;:LOAD AC WITH 0
1835      MOV      #16,ASH      ;:LOAD SHIFT COUNT (ASH) WITH 16.
1836      CMP      #0,AC      ;:COMPARE AC WITH 0
1837      BEQ      64$      ;:IF NO.ERROR SKIP HLT
1838      HLT
1839      64$:
1840      CMP      #0,AMQ      ;:COMPARE MQ WITH 0
1841      BEQ      65$      ;:IF NO.ERROR SKIP HLT
1842      HLT      ;:CALL ERROR ROUTINE
1843      65$:
1844      CMPB     #36,SR      ;:COMPARE SR WITH 36
1845      BEQ      66$      ;:IF NO.ERROR SKIP HLT
1846      HLT      ;:CALL ERROR ROUTINE
1847      66$:
1848      SCOPE      ;:TEST OF ARITHMETIC SHIFT
1849      MOV      #-1,AMQ     ;:LOAD MQ WITH -1
1850      MOV      #0,AC      ;:LOAD AC WITH 0
1851      MOV      #15,ASH     ;:LOAD SHIFT COUNT (ASH) WITH 15.
1852      CMP      #7777,AC    ;:COMPARE AC WITH 7777
1853      BEQ      67$      ;:IF NO.ERROR SKIP HLT
1854      HLT      ;:CALL ERROR ROUTINE
1855      67$:
1856      CMP      #10000,AMQ   ;:COMPARE MQ WITH 10000
1857      BEQ      68$      ;:IF NO.ERROR SKIP HLT
1858      HLT      ;:CALL ERROR ROUTINE
1859      68$:
1860      CMPB     #0,SR      ;:COMPARE SR WITH 0
1861      BEQ      69$      ;:IF NO.ERROR SKIP HLT
1862      HLT      ;:CALL ERROR ROUTINE
1863      69$:
1864      SCOPE      ;:TEST OF ARITHMETIC SHIFT
1865      MOV      #-1,AMQ     ;:LOAD MQ WITH -1
1866      MOV      #0,AC      ;:LOAD AC WITH 0
1867      MOV      #16,ASH     ;:LOAD SHIFT COUNT (ASH) WITH 16.
1868      CMP      #7777,AC    ;:COMPARE AC WITH 7777
1869      BEQ      70$      ;:IF NO.ERROR SKIP HLT
1870      HLT      ;:CALL ERROR ROUTINE
1871      70$:
1872      CMP      #0,AMQ      ;:COMPARE MQ WITH 0
1873      BEQ      71$      ;:IF NO.ERROR SKIP HLT
1874      HLT      ;:CALL ERROR ROUTINE
1875      71$:
1876      CMPB     #211,SR     ;:COMPARE SR WITH 211
1877

```

1878	011754	001401			BEQ	72\$			: IF NO.ERROR SKIP HLT
1879	011756	104000			HLT				: CALL ERROR ROUTINE
1880	011760						72\$:		
1881									
1882	011760	104400			SCOPE				: TEST OF ARITHMETIC SHIFT
1883	011762	012777	177776	166256	MOV	#-2, MQ			: LOAD MQ WITH -2
1884	011770	012777	100000	166246	MOV	#100000, AC			: LOAD AC WITH 100000
1885	011776	012777	000017	166256	MOV	#15, ASH			: LOAD SHIFT COUNT (ASH) WITH 15.
1896	012004	022777	177777	166232	CMP	#-1, AC			: COMPARE AC WITH -1
1887	012012	001401			BEQ	73\$			: IF NO.ERROR SKIP HLT
1888	012014	104000			HLT				: CALL ERROR ROUTINE
1889	012016						73\$:		
1890	012016	022777	000000	166222	CMP	#0, MQ			: COMPARE MQ WITH 0
1891	012024	001401			BEQ	74\$			: IF NO.ERROR SKIP HLT
1892	012026	104000			HLT				: CALL ERROR ROUTINE
1893	012030						74\$:		
1894	012030	122777	000150	166216	CMPB	#150, SR			: COMPARE SR WITH 150
1895	012036	001401			BEQ	75\$			: IF NO.ERROR SKIP HLT
1896	012040	104000			HLT				: CALL ERROR ROUTINE
1897	012042						75\$:		
1898	012042	104400			SCOPE				: TEST OF ARITHMETIC SHIFT
1899	012044	012777	125252	166174	MOV	#125252, MQ			: LOAD MQ WITH 125252
1900	012052	012777	000000	166164	MOV	#0, AC			: LOAD AC WITH 0
1901	012060	012777	000017	166174	MOV	#15, ASH			: LOAD SHIFT COUNT (ASH) WITH 15.
1902	012066	022777	052525	166150	CMP	#52525, AC			: COMPARE AC WITH 52525
1903	012074	001401			BEQ	76\$			: IF NO.ERROR SKIP HLT
1904	012076	104000			HLT				: CALL ERROR ROUTINE
1905	012100						76\$:		
1906	012100	022777	000000	166140	CMP	#0, MQ			: COMPARE MQ WITH 0
1907	012106	001401			BEQ	77\$			: IF NO.ERROR SKIP HLT
1908	012110	104000			HLT				: CALL ERROR ROUTINE
1909	012112						77\$:		
1910	012112	122777	000010	166134	CMPB	#10, SR			: COMPARE SR WITH 10
1911	012120	001401			BEQ	78\$			: IF NO.ERROR SKIP HLT
1912	012122	104000			HLT				: CALL ERROR ROUTINE
1913	012124						78\$:		
1914									
1915	012124	104400			SCOPE				: TEST OF ARITHMETIC SHIFT
1916	012126	012777	125252	166112	MOV	#125252, MQ			: LOAD MQ WITH 125252
1917	012134	012777	000000	166102	MOV	#0, AC			: LOAD AC WITH 0
1918	012142	012777	000020	166112	MOV	#16, ASH			: LOAD SHIFT COUNT (ASH) WITH 16.
1919	012150	022777	025252	166066	CMP	#25252, AC			: COMPARE AC WITH 25252
1920	012156	001401			BEQ	79\$			: IF NO.ERROR SKIP HLT
1921	012160	104000			HLT				: CALL ERROR ROUTINE
1922	012162						79\$:		
1923	012162	022777	000000	166056	CMP	#0, MQ			: COMPARE MQ WITH 0

```

1924 012170 001401 BEQ 80$ ; IF NO.ERROR SKIP HLT
1925 012172 104000 HLT ; CALL ERROR ROUTINE
1926 012174 80$:
1927 012174 122777 000211 166052 CMPB #211,SR ; COMPARE SR WITH 211
1928 012202 001401 BEQ 81$ ; IF NO.ERROR SKIP HLT
1929 012204 104000 HLT ; CALL ERROR ROUTINE
1930 012206 81$:
1931 012206 104400 SCOPE ; TEST OF ARITHMETIC SHIFT
1932 012210 012777 052525 166030 MOV #52525,AMQ ; LOAD MQ WITH 52525
1933 012216 012777 000000 166020 MOV #0,AC ; LOAD AC WITH 0
1934 012224 012777 000020 166030 MOV #16,ASH ; LOAD SHIFT COUNT (ASH) WITH 16.
1935 012232 022777 052525 166004 CMP #52525,AC ; COMPARE AC WITH 52525
1936 012240 001401 BEQ 82$ ; IF NO.ERROR SKIP HLT
1937 012242 104000 HLT ; CALL ERROR ROUTINE
1938 012244 82$:
1939 012244 022777 000000 165774 CMP #0,AMQ ; COMPARE MQ WITH 0
1940 012252 001401 BEQ 83$ ; IF NO.ERROR SKIP HLT
1941 012254 104000 HLT ; CALL ERROR ROUTINE
1942 012256 83$:
1943 012256 122777 000010 165770 CMPB #10,SR ; COMPARE SR WITH 10
1944 012264 001401 BEQ 84$ ; IF NO.ERROR SKIP HLT
1945 012266 104000 HLT ; CALL ERROR ROUTINE
1946 012270 84$:
1947
1948 012270 104400 SCOPE ; TEST OF ARITHMETIC SHIFT
1949 012272 012777 000000 165746 MOV #0,AMQ ; LOAD MQ WITH 0
1950 012300 012777 177777 165736 MOV #-1,AC ; LOAD AC WITH -1
1951 012306 012777 000020 165746 MOV #16,ASH ; LOAD SHIFT COUNT (ASH) WITH 16.
1952 012314 022777 100000 165722 CMP #100000,AC ; COMPARE AC WITH 100000
1953 012322 001401 BEQ 85$ ; IF NO.ERROR SKIP HLT
1954 012324 104000 HLT ; CALL ERROR ROUTINE
1955 012326 85$:
1956 012326 022777 000000 165712 CMP #0,AMQ ; COMPARE MQ WITH 0
1957 012334 001401 BEQ 86$ ; IF NO.ERROR SKIP HLT
1958 012336 104000 HLT ; CALL ERROR ROUTINE
1959 012340 86$:
1960 012340 122777 000110 165706 CMPB #110,SR ; COMPARE SR WITH 110
1961 012346 001401 BEQ 87$ ; IF NO.ERROR SKIP HLT
1962 012350 104000 HLT ; CALL ERROR ROUTINE
1963 012352 87$:

```

1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004

012352

012352 104400  
012354 012777 000000 165664  
012362 012777 000000 165654  
012370 005077 165662  
012374 022777 000000 165642  
012402 001401  
012404 104000  
012406  
012406 022777 000000 165632  
012414 001401  
012416 104000  
012420  
012420 022777 017037 165624  
012426 001401  
012430 104000  
012432

\*\*\*\*\*  
: AT THIS POINT, THE ARITHMETIC SHIFT WORKS  
: \*\*\*\*\*  
: \*\*\*\*\*  
: TEST OF NORMALIZE  
: \*\*\*\*\*  
NORMAL:

SCOPE ; TEST OF NORMALIZE  
MOV #0, @MQ ; LOAD MQ WITH 0  
MOV #0, @AC ; LOAD AC WITH 0  
CLR @NOR ; START NORMALIZE  
CMP #0, @AC ; COMPARE AC WITH 0  
BEQ 64\$ ; IF NO. ERROR SKIP HLT  
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, @SC ; COMPARE SC WITH 37  
AND SR= 36  
BEQ 66\$ ; IF NO. ERROR SKIP HLT  
HLT ; CALL ERROR ROUTINE  
66\$:  
SCOPE ; TEST OF NORMALIZE  
MOV #-1, @MQ ; LOAD MQ WITH -1  
MOV #-1, @AC ; LOAD AC WITH -1  
CLR @NOR ; START NORMALIZE  
CMP #140000, @AC ; COMPARE AC WITH 140000  
BEQ 67\$ ; IF NO. ERROR SKIP HLT  
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, @SC ; COMPARE SC WITH 30.  
AND SR= 310



2005	012506	001401			BEQ	69\$			; IF NO.ERROR SKIP HLT
2006	012510	104000			HLT				; CALL ERROR ROUTINE
2007	012512			69\$					
2008	012512	104400			SCOPE				; TEST OF NORMALIZE
2009	012514	012777	000000	165524	MOV	#0, @MQ			; LOAD MQ WITH 0
2010	012522	012777	177777	165514	MOV	#-1, @AC			; LOAD AC WITH -1
2011	012530	005077	165522		CLR	@NOR			; START NORMALIZE
2012	012534	022777	140000	165502	CMP	#140000, @AC			; COMPARE AC WITH 140000
2013	012542	001401			BEQ	70\$			; IF NO.ERROR SKIP HLT
2014	012544	104000			HLT				; CALL ERROR ROUTINE
2015	012546			70\$					
2016	012546	022777	000000	165472	CMP	#0, @MQ			; COMPARE MQ WITH 0
2017	012554	001401			BEQ	71\$			; IF NO.ERROR SKIP HLT
2018	012556	104000			HLT				; CALL ERROR ROUTINE
2019	012560			71\$					
2020	012560	022777	144016	165464	CMP	#144016, @SC			; COMPARE SC WITH 14.
2021									; AND SR= 310
2022	012566	001401			BEQ	72\$			; IF NO.ERROR SKIP HLT
2023	012570	104000			HLT				; CALL ERROR ROUTINE
2024	012572			72\$					
2025									
2026									
2027									
2028	012572	104400			SCOPE				; TEST OF NORMALIZE
2029	012574	012777	000001	165444	MOV	#1, @MQ			; LOAD MQ WITH 1
2030	012602	012777	000000	165434	MOV	#0, @AC			; LOAD AC WITH 0
2031	012610	005077	165442		CLR	@NOR			; START NORMALIZE
2032	012614	022777	040000	165422	CMP	#40000, @AC			; COMPARE AC WITH 40000
2033	012622	001401			BEQ	73\$			; IF NO.ERROR SKIP HLT
2034	012624	104000			HLT				; CALL ERROR ROUTINE
2035	012626			73\$					
2036	012626	022777	000000	165412	CMP	#0, @MQ			; COMPARE MQ WITH 0
2037	012634	001401			BEQ	74\$			; IF NO.ERROR SKIP HLT
2038	012636	104000			HLT				; CALL ERROR ROUTINE
2039	012640			74\$					
2040	012640	022777	004036	165404	CMP	#004036, @SC			; COMPARE SC WITH 30.
2041									; AND SR= 10
2042	012646	001401			BEQ	75\$			; IF NO.ERROR SKIP HLT
2043	012650	104000			HLT				; CALL ERROR ROUTINE
2044	012652			75\$					
2045									
2046									
2047	012652	104400			SCOPE				; TEST OF NORMALIZE
2048	012654	012777	000005	165364	MOV	#5, @MQ			; LOAD MQ WITH 5
2049	012662	012777	000000	165354	MOV	#0, @AC			; LOAD AC WITH 0
2050	012670	005077	165362		CLR	@NOR			; START NORMALIZE
2051	012674	022777	050000	165342	CMP	#50000, @AC			; COMPARE AC WITH 50000

2052	012702	001401			BEQ	76\$		; IF NO.ERROR SKIP HLT
2053	012704	104000			HLT			; CALL ERROR ROUTINE
2054	012706				76\$:			
2055	012706	022777	000000	165332	CMP	#0,AMQ		; COMPARE MQ WITH 0
2056	012714	001401			BEQ	77\$		; IF NO.ERROR SKIP HLT
2057	012716	104000			HLT			; CALL ERROR ROUTINE
2058	012720				77\$:			
2059	012720	022777	004034	165324	CMP	#004034,ASC		; COMPARE SC WITH 28.
2060								; AND SR= 10
2061	012726	001401			BEQ	78\$		; IF NO.ERROR SKIP HLT
2062	012730	104000			HLT			; CALL ERROR ROUTINE
2063	012732				78\$:			
2064								
2065	012732	104400			SCOPE			; TEST OF NORMALIZE
2066	012734	012777	000001	165304	MOV	#1,AMQ		; LOAD MQ WITH 1
2067	012742	012777	100000	165274	MOV	#100000,AC		; LOAD AC WITH 100000
2068	012750	005077	165302		CLR	ANOR		; START NORMALIZE
2069	012754	022777	100000	165262	CMP	#100000,AC		; COMPARE AC WITH 100000
2070	012762	001401			BEQ	79\$		; IF NO.ERROR SKIP HLT
2071	012764	104000			HLT			; CALL ERROR ROUTINE
2072	012766				79\$:			
2073	012766	022777	000001	165252	CMP	#1,AMQ		; COMPARE MQ WITH 1
2074	012774	001401			BEQ	80\$		; IF NO.ERROR SKIP HLT
2075	012776	104000			HLT			; CALL ERROR ROUTINE
2076	013000				80\$:			
2077	013000	022777	140000	165244	CMP	#140000,ASC		; COMPARE SC WITH 0
2078								; AND SR= 300
2079	013006	001401			BEQ	81\$		; IF NO.ERROR SKIP HLT
2080	013010	104000			HLT			; CALL ERROR ROUTINE
2081	013012				81\$:			
2082								
2083								
2084								
2085	013012	104400			SCOPE			; TEST OF NORMALIZE
2086	013014	012777	125252	165224	MOV	#125252,AMQ		; LOAD MQ WITH 125252
2087	013022	012777	170000	165214	MOV	#170000,AC		; LOAD AC WITH 170000
2088	013030	005077	165222		CLR	ANOR		; START NORMALIZE
2089	013034	022777	100005	165202	CMP	#100005,AC		; COMPARE AC WITH 100005
2090	013042	001401			BEQ	82\$		; IF NO.ERROR SKIP HLT
2091	013044	104000			HLT			; CALL ERROR ROUTINE
2092	013046				82\$:			
2093	013046	022777	052520	165172	CMP	#52520,AMQ		; COMPARE MQ WITH 52520
2094	013054	001401			BEQ	83\$		; IF NO.ERROR SKIP HLT
2095	013056	104000			HLT			; CALL ERROR ROUTINE
2096	013060				83\$:			
2097	013060	022777	140003	165164	CMP	#140003,ASC		; COMPARE SC WITH 3
2098								; AND SR= 300
2099	013066	001401			BEQ	84\$		; IF NO.ERROR SKIP HLT
2100	013070	104000			HLT			; CALL ERROR ROUTINE
2101	013072				84\$:			

2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141

013072 004767 004324  
013076 032777 000002 165070  
013104 001402  
013106 000167 003042  
  
013112  
  
013112 104400  
013114 012777 000000 165124  
013122 012777 000000 165120  
013130 022777 000000 165106  
013136 001401  
013140 104000  
013142  
013142 022777 000000 165076  
013150 001401  
013152 104000  
013154  
013154 122777 000036 165072

\*\*\*\*\*  
: AT THIS POINT NORMALIZE WORKS  
\*\*\*\*\*

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

JSR %7,CKSWR ;CHECK FOR 1G  
BIT #2,QSWR  
BEQ .DIV  
JMP .DEV

\*\*\*\*\*  
: TEST OF MULTIPLY  
: .DIV:  
\*\*\*\*\*

SCOPE ;TEST OF MULTIPLY  
MOV #0,QMQ ;LOAD MQ WITH 0  
MOV #0,QMUL ;LOAD MUL WITH 0 AND MULTIPLY  
CMP #0,QAC ;COMPARE AC WITH 0  
BEQ 64\$ ;IF NO.ERROR SKIP HLT  
HLT ;CALL ERROR ROUTINE  
  
64\$: CMP #0,QMQ ;COMPARE MQ WITH 0  
BEQ 65\$ ;IF NO.ERROR SKIP HLT  
HLT ;CALL ERROR ROUTINE  
  
65\$: CMPB #36,QSR ;COMPARE SR WITH 36

```

2142 013162 001401      BEQ      66$      ; IF NO.ERROR SKIP HLT
2143 013164 104000      HLT      ; CALL ERROR ROUTINE
2144 013166      66$:
2145
2146
2147 013166 104400      SCOPE      ; TEST OF MULTIPLY
2148 013170 012777 177777 165050  MOV      #-1,2MQ      ; LOAD MQ WITH -1
2149 013176 012777 000001 165044  MOV      #1,2MUL      ; LOAD MUL WITH 1 AND MULTIPLY
2150 013204 022777 177777 165032  CMP      #-1,2AC      ; COMPARE AC WITH -1
2151 013212 001401      BEQ      67$      ; IF NO.ERROR SKIP HLT
2152 013214 104000      HLT      ; CALL ERROR ROUTINE
2153
2154 013216 022777 177777 165022 67$:      CMP      #-1,2MQ      ; COMPARE MQ WITH -1
2155 013224 001401      BEQ      68$      ; IF NO.ERROR SKIP HLT
2156 013226 104000      HLT      ; CALL ERROR ROUTINE
2157
2158 013230 122777 000342 165016 68$:      CMPB     #342,2SR     ; COMPARE SR WITH 342
2159 013236 001401      BEQ      69$      ; IF NO.ERROR SKIP HLT
2160 013240 104000      HLT      ; CALL ERROR ROUTINE
2161
2162
2163
2164
2165 013242 104400      SCOPE      ; TEST OF MULTIPLY
2166 013244 012777 125252 164774  MOV      #125252,2MQ   ; LOAD MQ WITH 125252
2167 013252 012777 000002 164770  MOV      #2,2MUL      ; LOAD MUL WITH 2 AND MULTIPLY
2168 013260 022777 177777 164756  CMP      #-1,2AC      ; COMPARE AC WITH -1
2169 013266 001401      BEQ      70$      ; IF NO.ERROR SKIP HLT
2170 013270 104000      HLT      ; CALL ERROR ROUTINE
2171
2172 013272 022777 052524 164746 70$:      CMP      #52524,2MQ   ; COMPARE MQ WITH 52524
2173 013300 001401      BEQ      71$      ; IF NO.ERROR SKIP HLT
2174 013302 104000      HLT      ; CALL ERROR ROUTINE
2175 013304
2176 013304 122777 000340 164742 71$:      CMPB     #340,2SR     ; COMPARE SR WITH 340

```

2177	013312	001401			BEQ	72\$		; IF NO.ERROR SKIP HLT
2178	013314	104000			HLT			; CALL ERROR ROUTINE
2179	013316				72\$:			
2180	013316	104400			SCOPE			; TEST OF MULTIPLY
2181	013320	012777	052525	164720	MOV	#52525, @MQ		; LOAD MQ WITH 52525
2182	013326	012777	000002	164714	MOV	#2, @MUL		; LOAD MUL WITH 2 AND MULTIPLY
2183	013334	022777	000000	164702	CMP	#0, @AC		; COMPARE AC WITH 0
2184	013342	001401			BEQ	73\$		; IF NO.ERROR SKIP HLT
2185	013344	104000			HLT			; CALL ERROR ROUTINE
2186	013346				73\$:			
2187	013346	022777	125252	164672	CMP	#125252, @MQ		; COMPARE MQ WITH 125252
2188	013354	001401			BEQ	74\$		; IF NO.ERROR SKIP HLT
2189	013356	104000			HLT			; CALL ERROR ROUTINE
2190	013360				74\$:			
2191	013360	122777	000020	164666	CMPB	#20, @SR		; COMPARE SR WITH 20
2192	013366	001401			BEQ	75\$		; IF NO.ERROR SKIP HLT
2193	013370	104000			HLT			; CALL ERROR ROUTINE
2194	013372				75\$:			
2195								
2196								
2197								
2198	013372	104400			SCOPE			; TEST OF MULTIPLY
2199	013374	012777	125252	164644	MOV	#125252, @MQ		; LOAD MQ WITH 125252
2200	013402	012777	040000	164640	MOV	#40000, @MUL		; LOAD MUL WITH 40000 AND MULTIPLY
2201	013410	022777	165252	164626	CMP	#165252, @AC		; COMPARE AC WITH 165252
2202	013416	001401			BEQ	76\$		; IF NO.ERROR SKIP HLT
2203	013420	104000			HLT			; CALL ERROR ROUTINE
2204	013422				76\$:			
2205	013422	022777	100000	164616	CMP	#100000, @MQ		; COMPARE MQ WITH 100000
2206	013430	001401			BEQ	77\$		; IF NO.ERROR SKIP HLT
2207	013432	104000			HLT			; CALL ERROR ROUTINE
2208	013434				77\$:			
2209	013434	122777	000300	164612	CMPB	#300, @SR		; COMPARE SR WITH 300
2210	013442	001401			BEQ	78\$		; IF NO.ERROR SKIP HLT
2211	013444	104000			HLT			; CALL ERROR ROUTINE
2212	013446				78\$:			
2213								
2214								
2215	013446	104400			SCOPE			; TEST OF MULTIPLY
2216	013450	012777	100000	164570	MOV	#100000, @MQ		; LOAD MQ WITH 100000
2217	013456	012777	100000	164564	MOV	#100000, @MUL		; LOAD MUL WITH 100000 AND MULTIPLY
2218	013464	022777	040000	164552	CMP	#40000, @AC		; COMPARE AC WITH 40000
2219	013472	001401			BEQ	79\$		; IF NO.ERROR SKIP HLT
2220	013474	104000			HLT			; CALL ERROR ROUTINE
2221	013476				79\$:			
2222	013476	022777	000000	164542	CMP	#0, @MQ		; COMPARE MQ WITH 0
2223	013504	001401			BEQ	80\$		; IF NO.ERROR SKIP HLT
2224	013506	104000			HLT			; CALL ERROR ROUTINE
2225	013510				80\$:			
2226	013510	122777	000010	164536	CMPB	#10, @SR		; COMPARE SR WITH 10
2227	013516	001401			BEQ	81\$		; IF NO.ERROR SKIP HLT
2228	013520	104000			HLT			; CALL ERROR ROUTINE
2229	013522				81\$:			

```

2230 ;*****
2231 ; TEST OF DIVIDE
2232 ;*****
2233 DIVIDE:
2234 013522 104400 SCOPE ; TEST OF DIVIDE
2235 013524 005077 164516 CLR @MQ ; LOAD MQ WITH 0
2236 013530 005077 164506 CLR @DIV ; LOAD DIV WITH 0 AND DIVIDE
2237 013534 106177 164514 ROLB @SR ; SHIFT OVERFLOW BIT INTO PS
2238 013540 102401 BVS 1$ ; SKIP HALT IF GOOD
2239 013542 104000 HLT ; HALT ON ERROR
2240 013544
2241 013544 005777 164474 1$: TST @AC ; CHECK AC'S SIGN
2242 013550 100405 BMI .MIN ;
2243 013552 106177 164476 ROLB @SR ; SET APROPRIATE N AND V BITS
2244 013556 100006 BPL .CONT ;
2245 013560 104000 HLT ; WRONG SIGN
2246 013562 000404 BR .CONT ;
2247 013564 106177 164464 .MIN: ROLB @SR ; SET APROPRIATE N AND V BITS
2248 013570 100401 BMI .CONT ;
2249 013572 104000 .HLT: HLT ;
2250
2251 .CONT:
2252 013574 104400 SCOPE ; TEST OF DIVIDE
2253 013576 005077 164444 CLR @MQ ; CLEAR THE MQ
2254 013602 012777 052525 164434 MOV #52525,@AC ; LOAD AC WITH 52525
2255 013610 012777 052525 164424 MOV #52525,@DIV ; LOAD DIV WITH 52525 AND DIVIDE
2256 013616 106177 164432 ROLB @SR ; SHIFT OVERFLOW BIT INTO PS
2257 013622 102401 BVS 1$ ; SKIP HALT IF GOOD
2258 013624 104000 HLT ; HALT ON ERROR
2259 013626
2260 013626 005777 164412 1$: TST @AC ; CHECK AC'S SIGN
2261 013632 100405 BMI .MIN1 ;
2262 013634 106177 164414 ROLB @SR ; SET APROPRIATE N AND V BITS
2263 013640 100006 BPL .CONT1 ;
2264 013642 104000 HLT ; WRONG SIGN
2265 013644 000404 BR .CONT1 ;
2266 013646 106177 164402 .MIN1: ROLB @SR ; SET APROPRIATE N AND V BITS
2267 013652 100401 BMI .CONT1 ;
2268 013654 104000 .HLT1: HLT ;
2269 013656 .CONT1:
2270 013656 104400 SCOPE ; TEST OF DIVIDE
2271 013660 012777 177777 164360 MOV #-1,@MQ ; LOAD MQ WITH -1
2272 013666 012777 177777 164350 MOV #-1,@AC ; LOAD AC WITH -1
2273 013674 012777 000001 164340 MOV #1,@DIV ; LOAD DIV WITH 1 AND DIVIDE
2274 013702 022777 000000 164334 CMP #0,@AC ; COMPARE AC WITH 0 (REMAINDER)
2275 013710 001401 BEQ 64$ ; IF NO ERROR SKIP HLT
2276 013712 104000 HLT ; CALL ERROR ROUTINE
2277 013714
2278 013714 022777 177777 164324 64$: CMP #-1,@MQ ; COMPARE MQ WITH -1 (QUOTIANT)
2279 013722 001401 BEQ 65$ ; IF NO ERROR SKIP HLT
2280 013724 104000 HLT ; CALL ERROR ROUTINE
2281 013726
2282 013726 122777 000320 164320 65$: CMPB #320,@SR ; COMPARE SR WITH 320
2283 013734 001401 BEQ 66$ ; IF NO ERROR SKIP HLT
2284 013736 104000 HLT ; CALL ERROR ROUTINE
2285 013740 66$:

```



2342	014166	104400			SCOPE				
2343	014170	012777	177777	164050	MOV	#-1, @MQ			: TEST OF DIVIDE
2344	014176	012777	177777	164040	MOV	#-1, @AC			: LOAD MQ WITH -1
2345	014204	012777	177777	164030	MOV	#-1, @DIV			: LOAD AC WITH -1
2346	014212	022777	000000	164024	CMP	#0, @AC			: LOAD DIV WITH -1 AND DIVIDE
2348	014220	001401			BEQ	76\$			: COMPARE AC WITH 0 (REMAINDER)
2349	014222	104000			HLT				: IF NO ERROR SKIP HLT
2350	014224							76\$:	: CALL ERROR ROUTINE
2351	014224	022777	000001	164014	CMP	#1, @MQ			: COMPARE MQ WITH 1 (QUOTIANT)
2352	014232	001401			BEQ	77\$			: IF NO ERROR SKIP HLT
2353	014234	104000			HLT				: CALL ERROR ROUTINE
2354	014236							77\$:	
2355	014236	122777	000022	164010	CMPB	#22, @SR			: COMPARE SR WITH 22
2356	014244	001401			BEQ	78\$			: IF NO ERROR SKIP HLT
2357	014246	104000			HLT				: CALL ERROR ROUTINE
2358	014250							78\$:	
2359									
2360									
2361									
2362	014250	104400			SCOPE				: TEST OF DIVIDE
2363	014252	012777	000000	163766	MOV	#0, @MQ			: LOAD MQ WITH 0
2364	014260	012777	025253	163756	MOV	#25253, @AC			: LOAD AC WITH 25253
2365	014266	012777	125252	163746	MOV	#125252, @DIV			: LOAD DIV WITH 125252 AND DIVIDE
2366	014274	022777	000000	163742	CMP	#0, @AC			: COMPARE AC WITH 0 (REMAINDER)
2367	014302	001401			BEQ	79\$			: IF NO ERROR SKIP HLT
2368	014304	104000			HLT				: CALL ERROR ROUTINE
2369	014306							79\$:	
2370	014306	022777	100000	163732	CMP	#100000, @MQ			: COMPARE MQ WITH 100000 (QUOTIANT)
2371	014314	001401			BEQ	80\$			: IF NO ERROR SKIP HLT
2372	014316	104000			HLT				: CALL ERROR ROUTINE
2373	014320							80\$:	
2374	014320	122777	000320	163726	CMPB	#320, @SR			: COMPARE SR WITH 320
2375	014326	001401			BEQ	81\$			: IF NO ERROR SKIP HLT
2376	014330	104000			HLT				: CALL ERROR ROUTINE
2377	014332							81\$:	
2378	014332	104400			SCOPE				: TEST OF DIVIDE
2379	014334	012777	000001	163704	MOV	#1, @MQ			: LOAD MQ WITH 1
2380	014342	012777	025253	163674	MOV	#25253, @AC			: LOAD AC WITH 25253
2381	014350	012777	125252	163664	MOV	#125252, @DIV			: LOAD DIV WITH 125252 AND DIVIDE
2382	014356	022777	000001	163660	CMP	#1, @AC			: COMPARE AC WITH 1 (REMAINDER)
2383	014364	001401			BEQ	82\$			: IF NO ERROR SKIP HLT
2384	014366	104000			HLT				: CALL ERROR ROUTINE
2385	014370							82\$:	
2386	014370	022777	100000	163650	CMP	#100000, @MQ			: COMPARE MQ WITH 100000 (QUOTIANT)
2387	014376	001401			BEQ	83\$			: IF NO ERROR SKIP HLT
2388	014400	104000			HLT				: CALL ERROR ROUTINE
2389	014402							83\$:	
2390	014402	122777	000300	163644	CMPB	#300, @SR			: COMPARE SR WITH 300
2391	014410	001401			BEQ	84\$			: IF NO ERROR SKIP HLT
2392	014412	104000			HLT				: CALL ERROR ROUTINE
2393	014414							84\$:	
2394									
2395									
2396									
2397	014414	104400			SCOPE				: TEST OF DIVIDE



2398	014416	012777	077777	163622	MOV	#77777,AMQ	;LOAD MQ WITH 77777
2399	014424	012777	037777	163612	MOV	#37777,AC	;LOAD AC WITH 37777
2400	014432	012777	077777	163602	MOV	#77777,ADIV	;LOAD DIV WITH 77777 AND DIVIDE
2401	014440	022777	077776	163576	CMP	#77776,AC	;COMPARE AC WITH 77776 (REMAINDER)
2402	014446	001401			BEQ	85\$	;IF NO.ERROR SKIP HLT
2403	014450	104000			HLT		;CALL ERROR ROUTINE
2404	014452						
2405	014452	022777	077777	163566	CMP	#77777,AMQ	;COMPARE MQ WITH 77777 (QUOTIANT)
2406	014460	001401			BEQ	86\$	;IF NO.ERROR SKIP HLT
2407	014462	104000			HLT		;CALL ERROR ROUTINE
2408	014464						
2409	014464	122777	000000	163562	CMPB	#0,SR	;COMPARE SR WITH 0
2410	014472	001401			BEQ	87\$	;IF NO.ERROR SKIP HLT
2411	014474	104000			HLT		;CALL ERROR ROUTINE
2412	014476						
2413							
2414							
2415							
2416	014476	104400			SCOPE		;TEST OF DIVIDE
2417	014500	012777	100000	163540	MOV	#100000,AMQ	;LOAD MQ WITH 100000
2418	014506	012777	000000	163530	MOV	#0,AC	;LOAD AC WITH 0
2419	014514	012777	000002	163520	MOV	#2,ADIV	;LOAD DIV WITH 2 AND DIVIDE
2420	014522	022777	000000	163514	CMP	#0,AC	;COMPARE AC WITH 0 (REMAINDER)
2421	014530	001401			BEQ	88\$	;IF NO.ERROR SKIP HLT
2422	014532	104000			HLT		;CALL ERROR ROUTINE
2423	014534						
2424	014534	022777	040000	163504	CMP	#40000,AMQ	;COMPARE MQ WITH 40000 (QUOTIANT)
2425	014542	001401			BEQ	89\$	;IF NO.ERROR SKIP HLT
2426	014544	104000			HLT		;CALL ERROR ROUTINE
2427	014546						
2428	014546	122777	000022	163500	CMPB	#22,SR	;COMPARE SR WITH 22
2429	014554	001401			BEQ	90\$	;IF NO.ERROR SKIP HLT
2430	014556	104000			HLT		;CALL ERROR ROUTINE
2431	014560						
2432	014560	104400			SCOPE		;TEST OF DIVIDE
2433	014562	012777	100001	163456	MOV	#100001,AMQ	;LOAD MQ WITH 100001
2434	014570	012777	000000	163446	MOV	#0,AC	;LOAD AC WITH 0
2435	014576	012777	000002	163436	MOV	#2,ADIV	;LOAD DIV WITH 2 AND DIVIDE
2436	014604	022777	000001	163432	CMP	#1,AC	;COMPARE AC WITH 1 (REMAINDER)
2437	014612	001401			BEQ	91\$	;IF NO.ERROR SKIP HLT
2438	014614	104000			HLT		;CALL ERROR ROUTINE
2439	014616						
2440	014616	022777	040000	163422	CMP	#40000,AMQ	;COMPARE MQ WITH 40000 (QUOTIANT)
2441	014624	001401			BEQ	92\$	;IF NO.ERROR SKIP HLT
2442	014626	104000			HLT		;CALL ERROR ROUTINE
2443	014630						
2444	014630	122777	000000	163416	CMPB	#0,SR	;COMPARE SR WITH 0
2445	014636	001401			BEQ	93\$	;IF NO.ERROR SKIP HLT
2446	014640	104000			HLT		;CALL ERROR ROUTINE
2447	014642						
2448							
2449							
2450							
2451	014642	104400			SCOPE		;TEST OF DIVIDE
2452	014644	012777	037776	163374	MOV	#37776,AMQ	;LOAD MQ WITH 37776
2453	014652	012777	020000	163364	MOV	#20000,AC	;LOAD AC WITH 20000

2454	014660	012777	077777	163354	MOV	#77777, @DIV	; LOAD DIV WITH 77777 AND DIVIDE
2455	014666	022777	077776	163350	CMP	#77776, @AC	; COMPARE AC WITH 77776 (REMAINDER)
2456	014674	001401			BEQ	94\$	; IF NO.ERROR SKIP HLT
2457	014676	104000			HLT		; CALL ERROR ROUTINE
2458	014700						
2459	014700	022777	040000	163340	CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2460	014706	001401			BEQ	95\$	; IF NO.ERROR SKIP HLT
2461	014710	104000			HLT		; CALL ERROR ROUTINE
2462	014712						
2463	014712	122777	000000	163334	CMPB	#0, @SR	; COMPARE SR WITH 0
2464	014720	001401			BEQ	96\$	; IF NO.ERROR SKIP HLT
2465	014722	104000			HLT		; CALL ERROR ROUTINE
2466	014724						
2467							
2468							
2469	014724	104400			SCOPE		; TEST OF DIVIDE
2470	014726	012777	077777	163312	MOV	#77777, @MQ	; LOAD MQ WITH 77777
2471	014734	012777	177777	163302	MOV	#177777, @AC	; LOAD AC WITH 177777
2472	014742	012777	177776	163272	MOV	#177776, @DIV	; LOAD DIV WITH 177776 AND DIVIDE
2473	014750	022777	177777	163266	CMP	#177777, @AC	; COMPARE AC WITH 177777 (REMAINDER)
2474	014756	001401			BEQ	97\$	; IF NO.ERROR SKIP HLT
2475	014760	104000			HLT		; CALL ERROR ROUTINE
2476	014762						
2477	014762	022777	040000	163256	CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2478	014770	001401			BEQ	98\$	; IF NO.ERROR SKIP HLT
2479	014772	104000			HLT		; CALL ERROR ROUTINE
2480	014774						
2481	014774	122777	000040	163252	CMPB	#40, @SR	; COMPARE SR WITH 40
2482	015002	001401			BEQ	99\$	; IF NO.ERROR SKIP HLT
2483	015004	104000			HLT		; CALL ERROR ROUTINE
2484	015006						
2485	015006	104400			SCOPE		; TEST OF DIVIDE
2486	015010	012777	100001	163230	MOV	#100001, @MQ	; LOAD MQ WITH 100001
2487	015016	012777	157777	163220	MOV	#157777, @AC	; LOAD AC WITH 157777
2488	015024	012777	100000	163210	MOV	#100000, @DIV	; LOAD DIV WITH 100000 AND DIVIDE
2489	015032	022777	100001	163204	CMP	#100001, @AC	; COMPARE AC WITH 100001 (REMAINDER)
2490	015040	001401			BEQ	100\$	; IF NO.ERROR SKIP HLT
2491	015042	104000			HLT		; CALL ERROR ROUTINE
2492	015044						
2493	015044	022777	040000	163174	CMP	#40000, @MQ	; COMPARE MQ WITH 40000 (QUOTIANT)
2494	015052	001401			BEQ	101\$	; IF NO.ERROR SKIP HLT
2495	015054	104000			HLT		; CALL ERROR ROUTINE
2496	015056						
2497	015056	122777	000000	163170	CMPB	#0, @SR	; COMPARE SR WITH 0
2498	015064	001401			BEQ	102\$	; IF NO.ERROR SKIP HLT
2499	015066	104000			HLT		; CALL ERROR ROUTINE
2500	015070						
2501							
2502							
2503							
2504	015070	104400			SCOPE		; TEST OF DIVIDE
2505	015072	012777	052525	163146	MOV	#52525, @MQ	; LOAD MQ WITH 52525
2506	015100	012777	000000	163136	MOV	#0, @AC	; LOAD AC WITH 0
2507	015106	012777	052525	163126	MOV	#52525, @DIV	; LOAD DIV WITH 52525 AND DIVIDE
2508	015114	022777	000000	163122	CMP	#0, @AC	; COMPARE AC WITH 0 (REMAINDER)
2509	015122	001401			BEQ	103\$	; IF NO.ERROR SKIP HLT

2510	015124	104000			HLT				;CALL ERROR ROUTINE
2511	015126								
2512	015126	022777	000001	163112	103\$: CMP	#1, @MQ			;COMPARE MQ WITH 1 (QUOTIANT)
2513	015134	001401			BEQ	104\$			;IF NO.ERROR SKIP HLT
2514	015136	104000			HLT				;CALL ERROR ROUTINE
2515	015140								
2516	015140	122777	000022	163106	104\$: CMPB	#22, @SR			;COMPARE SR WITH 22
2517	015146	001401			BEQ	105\$			;IF NO.ERROR SKIP HLT
2518	015150	104000			HLT				;CALL ERROR ROUTINE
2519	015152								
2520	015152	104400			105\$: SCOPE				;TEST OF DIVIDE
2521	015154	012777	052524	163064	MOV	#52524, @MQ			;LOAD MQ WITH 52524
2522	015162	012777	000000	163054	MOV	#0, @AC			;LOAD AC WITH 0
2523	015170	012777	052525	163044	MOV	#52525, @DIV			;LOAD DIV WITH 52525 AND DIVIDE
2524	015176	022777	052524	163040	CMP	#52524, @AC			;COMPARE AC WITH 52524 (REMAINDER)
2525	015204	001401			BEQ	106\$			;IF NO.ERROR SKIP HLT
2526	015206	104000			HLT				;CALL ERROR ROUTINE
2527	015210								
2528	015210	022777	000000	163030	106\$: CMP	#0, @MQ			;COMPARE MQ WITH 0 (QUOTIANT)
2529	015216	001401			BEQ	107\$			;IF NO.ERROR SKIP HLT
2530	015220	104000			HLT				;CALL ERROR ROUTINE
2531	015222								
2532	015222	122777	000010	163024	107\$: CMPB	#10, @SR			;COMPARE SR WITH 10
2533	015230	001401			BEQ	108\$			;IF NO.ERROR SKIP HLT
2534	015232	104000			HLT				;CALL ERROR ROUTINE
2535	015234								
2536	015234	104400			108\$: SCOPE				;TEST OF DIVIDE
2537	015236	012777	000000	163002	MOV	#0, @MQ			;LOAD MQ WITH 0
2538	015244	012777	000000	162772	MOV	#0, @AC			;LOAD AC WITH 0
2539	015252	012777	125252	162762	MOV	#125252, @DIV			;LOAD DIV WITH 125252 AND DIVIDE
2540	015260	022777	000000	162756	CMP	#0, @AC			;COMPARE AC WITH 0 (REMAINDER)
2541	015266	001401			BEQ	109\$			;IF NO.ERROR SKIP HLT
2542	015270	104000			HLT				;CALL ERROR ROUTINE
2543	015272								
2544	015272	022777	000000	162740	109\$: CMP	#0, @MQ			;COMPARE MQ WITH 0 (QUOTIANT)
2545	015300	001401			BEQ	110\$			;IF NO.ERROR SKIP HLT
2546	015302	104000			HLT				;CALL ERROR ROUTINE
2547	015304								
2548	015304	122777	000036	162742	110\$: CMPB	#36, @SR			;COMPARE SR WITH 36
2549	015312	001401			BEQ	111\$			;IF NO.ERROR SKIP HLT
2550	015314	104000			HLT				;CALL ERROR ROUTINE
2551	015316								
2552	015316	104400			111\$: SCOPE				;TEST OF SUCCESSIVE MULTIPLIES
2553	015320	012777	000001	162720	MOV	#1, @MQ			
2554									
2555	015326	012777	000002	162714	MOV	#2, @MUL			
2556	015334	012777	000002	162706	MOV	#2, @MUL			
2557	015342	012777	000002	162700	MOV	#2, @MUL			
2558	015350	012777	000002	162672	MOV	#2, @MUL			
2559	015356	012777	000002	162664	MOV	#2, @MUL			
2560	015364	012777	000002	162656	MOV	#2, @MUL			
2561	015372	012777	000002	162650	MOV	#2, @MUL			
2562	015400	012777	000002	162642	MOV	#2, @MUL			
2563	015406	012777	000002	162634	MOV	#2, @MUL			
2564	015414	012777	000002	162626	MOV	#2, @MUL			
2565	015422	012777	000002	162620	MOV	#2, @MUL			

2566	015430	012777	000002	162612	MOV	#2, @MUL	
2567	015436	012777	000002	162604	MOV	#2, @MUL	
2568	015444	012777	000002	162576	MOV	#2, @MUL	
2569							
2570	015452	022777	040000	162566	CMP	#40000, @MQ	
2571	015460	001401			BEQ	112\$	; IF NO.ERROR SKIP HLT
2572	015462	104000			HLT		; CALL ERROR ROUTINE
2573	015464						
2574	015464	005777	162554		TST	@AC	
2575	015470	001401			BEQ	113\$	; IF NO.ERROR SKIP HLT
2576	015472	104000			HLT		; CALL ERROR ROUTINE
2577	015474						
2578	015474	122777	000022	162552	CMPB	#22, @SR	; CHECK STATUS 22
2579	015502	001401			BEQ	114\$	; IF NO.ERROR SKIP HLT
2580	015504	104000			HLT		; CALL ERROR ROUTINE
2581	015506						
2582	015506	104400					; TEST OF SUCCESSIVE DIVIDES
2583	015510	012777	040000	162530	SCOPE		
2584					MOV	#40000, @MQ	
2585	015516	012777	000002	162516	MOV	#2, @DIV	
2586	015524	012777	000002	162510	MOV	#2, @DIV	
2587	015532	012777	000002	162502	MOV	#2, @DIV	
2588	015540	012777	000002	162474	MOV	#2, @DIV	
2589	015546	012777	000002	162466	MOV	#2, @DIV	
2590	015554	012777	000002	162460	MOV	#2, @DIV	
2591	015562	012777	000002	162452	MOV	#2, @DIV	
2592	015570	012777	000002	162444	MOV	#2, @DIV	
2593	015576	012777	000002	162436	MOV	#2, @DIV	
2594	015604	012777	000002	162430	MOV	#2, @DIV	
2595	015612	012777	000002	162422	MOV	#2, @DIV	
2596	015620	012777	000002	162414	MOV	#2, @DIV	
2597	015626	012777	000002	162406	MOV	#2, @DIV	
2598	015634	012777	000002	162400	MOV	#2, @DIV	
2599							
2600	015642	005777	162376		TST	@AC	
2601	015646	001401			BEQ	115\$	; IF NO.ERROR SKIP HLT
2602	015650	104000			HLT		; CALL ERROR ROUTINE
2603	015652						
2604	015652	022777	000001	162366	CMP	#1, @MQ	
2605	015660	001401			BEQ	116\$	; IF NO.ERROR SKIP HLT
2606	015662	104000			HLT		; CALL ERROR ROUTINE
2607	015664						
2608	015664	122777	000022	162362	CMPB	#22, @SR	; CHECK STATUS 22
2609	015672	001401			BEQ	117\$	; IF NO.ERROR SKIP HLT
2610	015674	104000			HLT		; CALL ERROR ROUTINE
2611	015676						
2612	015676	104400					; TEST OR ALTERNATE MUL AND DIV
2613	015700	012777	052525	162340	SCOPE		
2614					MOV	#52525, @MQ	
2615	015706	012777	040000	162334	MOV	#40000, @MUL	
2616	015714	012777	040000	162320	MOV	#40000, @DIV	
2617	015722	012777	040000	162320	MOV	#40000, @MUL	
2618	015730	012777	040000	162304	MOV	#40000, @DIV	
2619	015736	012777	040000	162304	MOV	#40000, @MUL	
2620	015744	012777	040000	162270	MOV	#40000, @DIV	
2621	015752	012777	040000	162270	MOV	#40000, @MUL	

2622	015760	012777	040000	162254	MOV	#40000, @DIV	
2623	015766	012777	040000	162254	MOV	#40000, @MUL	
2624	015774	012777	040000	162240	MOV	#40000, @DIV	
2625							
2626	016002	022777	052525	162236	CMP	#52525, @MQ	
2627	016010	001401			BEQ	118\$	; IF NO.ERROR SKIP HLT
2628	016012	104000			HLT		; CALL ERROR ROUTINE
2629	016014						
2630	016014	005777	162224		TST	@AC	
2631	016020	001401			BEQ	119\$	; IF NO.ERROR SKIP HLT
2632	016022	104000			HLT		; CALL ERROR ROUTINE
2633	016024						
2634	016024	122777	000022	162222	CMPB	#22, @SR	; CHECK STATUS 22
2635	016032	001401			BEQ	120\$	; IF NO.ERROR SKIP HLT
2636	016034	104000			HLT		; CALL ERROR ROUTINE
2637	016036						
2638	016036	104400			SCOPE		; TEST OF FAST PROCESSING OF DATA
2639	016040	016700	162202		MOV	MQ, %0	; SET UP POINTER
2640	016044	012720	125252		MOV	#125252, (0)+	; LOAD MQ
2641	016050	012710	040000		MOV	#40000, (0)	; LOAD MUL
2642	016054	014001			MOV	-(0), %1	; SAVE MQ
2643	016056	014002			MOV	-(0), %2	; SAVE AC
2644	016060	005720			TST	(0)+	
2645	016062	020127	100000		CMP	%1, #100000	; CHECK MQ
2646	016066	001401			BEQ	121\$	; IF NO.ERROR SKIP HLT
2647	016070	104000			HLT		; CALL ERROR ROUTINE
2648	016072						
2649	016072	020227	165252		CMP	%2, #165252	; CHECK AC
2650	016076	001401			BEQ	122\$	; IF NO.ERROR SKIP HLT
2651	016100	104000			HLT		; CALL ERROR ROUTINE
2652	016102						
2653							
2654	016102	104400			SCOPE		; SAVE WITH DIVIDE
2655	016104	016700	162136		MOV	MQ, %0	
2656	016110	012710	000001		MOV	#1, (0)	; LOAD MQ WITH 1
2657	016114	012740	025253		MOV	#25253, -(0)	; LOAD AC WITH 25253
2658	016120	012740	125252		MOV	#125252, -(0)	; DIVIDE
2659	016124	005720			TST	(0)+	
2660	016126	012001			MOV	(0)+, %1	; SAVE THE AC IN R1
2661	016130	011002			MOV	(0), %2	; SAVE THE MQ IN R2
2662	016132	020127	000001		CMP	%1, #1	; TEST THE AC
2663	016136	001401			BEQ	123\$	; IF NO.ERROR SKIP HLT
2664	016140	104000			HLT		; CALL ERROR ROUTINE
2665	016142						
2666	016142	020227	100000		CMP	%2, #100000	; TEST THE MQ
2667	016146	001401			BEQ	124\$	; IF NO.ERROR SKIP HLT
2668	016150	104000			HLT		; CALL ERROR ROUTINE
2669	016152						
2670	016152	104400			SCOPE		
2671							

```

2672 C16154
2673
2674
2675
2676
2677
2678
2679
2680 016154 004767 001242
2681 016160 032777 002000 162006
2682 016166 001021
2683 016170 012737 000207 177566
2684 016176 105777 000502
2685 016202 100375
2686 016204 012705 020031
2687 016210 004767 001710
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697 016214 013700 000042
2698 016220 001404
2699 016222 004710
2700 016224 000240
2701 016226 000240
2702 016230 000240
2703 016232
2704
2705
2706
2707
2708
2709
2710 016232 004767 001164
2711 016236 032777 010000 161730
2712 016244 001417
2713 016246 005767 000104
2714 016252 001411
2715 016254 016767 000076 161532
2716 016252 016767 000072 161526
2717 016270 042767 000020 161500
2718 016276 000167 162056
2719 016302 000000
2720
2721
2722 016304 016767 161504 000044
2723 016312 016767 161500 000040
2724 016320 012767 016362 161466
2725 016326 005067 161464
2726 016332 005067 161440
2727 016336 005167 177740

```

```

.DEV:
;*****
; AT THIS POINT MULT AND DIV ARE OK
;*****
;*****
; BELL ON PASS COMPLETE
;*****
        JSR      %7,CKSWR          ;CHECK FOR CONT-G
        BIT      #2000,%SWR
        BNE      TRTRAP
        MOV      #207,%207,%177566
        TSTB    %TPS              ; IS TTY READY FOR NEXT CHARACTER?
        BPL      64$              ; IF READY BIT (BIT 7)=0--NO, LOOP
        MOV      #SEAE,%5         ; PRINT EAE OK
        JSR      %7,TTOUT

;*****
; END OF PASS CODE AND CHECK FOR AUTO MODE
;*****

SENDAD: MOV      #42,%0
        BEQ      DOAGN
        JSR      %7,(%0)
        NOP
        NOP
        NOP
DOAGN:

;*****
; ROUTINE TO CHECK FOR TRACE TRAP TO BE RUN WITH PROGRAM
;*****
TRTRAP: JSR      %7,CKSWR          ;CHECK FOR CONT-G
        BIT      #10000,%SWR      ;SHOULD WE RUN WITH TRACE TRAP
        BEQ      YESTR            ;YES
        TST      YESTR1          ;NO HAVE WE RAN WITH TRACE TRAP ON
        BEQ      TRPA            ;IF SO RESTORE PREVIOUS CONTENTS
        MOV      YESTR1,14
        MOV      YESTR2,16
        BIC      #20,%PSW        ;CLEAR TRACE TRAP
        JMP      BEGIN           ;START OF TEST WITH TRACE OFF
TRPA:   0
TRPB:   0
; YESTR: SAVE OLD CONTENTS, SET UP FOR TRACE TRAP
        MOV      14,YESTR1       ;SAVE ODT PC
        MOV      16,YESTR2       ;SAVE ODT STATUS
        MOV      #YESTR,14       ;NEW TRAP VECTOR
        CLR      16              ;NEW CONDITION CODES
        CLR      %PSW
        COM      TRPB

```

```

2728 016342 100403      BMI      1$
2729 016344 052767 000020 161424      BIS      #20,PSW      ;SET TRACE 09P
2730 016352      1$:
2731 016352 000167 162002      JMP      BEGIN      ;START OF TEST WITH TRACE ON
2732
2733 016350 000000      YESTR1: 0      ;STORAGE FOR ODT PC
2734 016360 000000      YESTR2: 0      ;STORAGE FOR ODT STATUS
2735 016362 000002      YESRT:  ;RETURN TO PROGRAM FROM TRAP
2736 016364 000000      HALT      ;RTI FAILED
2737      ;
2738      ; ENTERED WITH SYSTEM TRAP CALL(HLT)
2739 016366 004767 001030      PRINT: JSR      %7,CKSWR      ;PRINT OUT THE ERROR PC AND STATUS REGISTER
2740 016372 032777 002000 161574      BIT      #2000,ASWR      ;CHECK FOR CONT-G
2741 016400 001406      BEQ      PRNT
2742 016402 012737 000007 177566      MOV      #7,#177566
2743 016410 105777 000270      64$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2744 016414 100375      BPL      64$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2745 016416 004767 001000      PRNT: JSR      %7,CKSWR      ;CHECK FOR CONT-G
2746 016422 037727 161546 020000      BIT      ASWR,#20000      ;TEST FOR INHIBIT PRINT OUT
2747 016430 001401      BEQ      1$
2748 016432 000002      RTI
2749 016434 012667 000246      1$: MOV      (6)+,SAVPC      ;PC OF FAILING ROUTINE
2750 016440 012667 000244      MOV      (6)+,SAVCC      ;CC OF ERROR CONDITION
2751 016444 024446      CMP      -(6),-(6)      ;REPOSITION THE STACK
2752 016446 012777 000215 000226      MOV      #215,@TPB      ;CR
2753 016454 105777 000224      64$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2754 016460 100375      BPL      64$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2755 016462 012777 000212 000212      MOV      #212,@TPB      ;LINE FEED
2756 016470 105777 000210      65$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2757 016474 100375      BPL      65$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2758 016476 010267 000172      MOV      %2,SAVR2      ;SAVE R2
2759 016502 010367 000170      MOV      %3,SAVR3      ;SAVE R3
2760 016506 010467 000166      MOV      %4,SAVR4      ;SAVE R4
2761 016512 016702 000170      MOV      SAVPC,%2
2762 016516 004767 000174      JSR      %7,PRTAB      ;PRINT OCTAL NUMBER
2763 016522 012777 000240 000152      MOV      #240,@TPB
2764 016530 105777 000150      66$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2765 016534 100375      BPL      66$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2766 016536 017702 161502      MOV      @AC,%2
2767 016542 004767 000150      JSR      %7,PRTAB      ;PRINT OCTAL NUMBER
2768 016546 012777 000240 000126      MOV      #240,@TPB
2769 016554 105777 000124      67$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2770 016560 100375      BPL      67$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2771 016562 017702 161460      MOV      @MQ,%2
2772 016566 004767 000124      JSR      %7,PRTAB      ;PRINT OCTAL NUMBER
2773 016572 012777 000240 000102      MOV      #240,@TPB
2774 016600 105777 000100      68$: TSTB     @TPS      ;IS TTY READY FOR NEXT CHARACTER?
2775 016604 100375      BPL      68$      ;IF READY BIT (BIT 7)=0--NO, LOOP
2776 016606 017702 161440      MOV      @SC,%2
2777 016612 004767 000100      JSR      %7,PRTAB      ;PRINT OCTAL NUMBER
2778 016616 016702 000052      MOV      SAVR2,%2      ;RESTORE REGISTERS
2779 016622 015703 000050      MOV      SAVR3,%3
2780 016626 016704 000046      MOV      SAVR4,%4
2781 016632 004767 000564      JSR      %7,CKSWR      ;CHECK FOR CONT-G
2782 016636 005777 161332      TST      ASWR      ;CHECK SR FOR HALT SWITCH
2783 016642 100004      BPL      3$

```





2840	017132	012767	000005	000044	XLIST:	MOV	#5,ASCNT		;SEND 5 CHAR TO TTY
2841	017140				WAIT2:				
2842	017140	105777	177540		64\$:	TSTB	@TPS		; IS TTY READY FOR NEXT CHARACTER?
2843	017144	100375				BPL	64\$		; IF READY BIT (BIT 7)=0--NO, LOOP
2844	017146	014477	177530			MOV	-(4),@TPB		
2845	017152	005367	000026			DEC	ASCNT		
2846	017156	001401				BEQ	HDFHM		
2847	017160	000767				BR	WAIT2		
2848	017162				HDFHM:				
2849	017162	105777	177516		64\$:	TSTB	@TPS		; IS TTY READY FOR NEXT CHARACTER?
2850	017166	100375				BPL	64\$		; IF READY BIT (BIT 7)=0--NO, LOOP
2851	017170	000207				RTS	%7		;HEAD FOR HOME
2852	017172	000000			TOODLE:	0			
2853	017174	000000			SEVEN:	0			
2854	017176	000070			DECML:	0			
2855	017200	000000			WGTC:	0			
2856	017202	000000			BINCT:	0			
2857	017204	000000			ASCNT:	0			
2858	017206	000000			LIST:	0			
2859	017210	000000				0			
2860	017212	000000				0			
2861	017214	000000				0			
2862	017216	000000				0			

```

2863 ;*****
2864 ; SCOPE LOOP
2865 ; ENTERED BY USER TRAP
2866 ;*****
2867
2868 017220 SCOPEA: JSR %7,CKSWR ;CHECK FOR CONT-G
2869 017220 004767 000176 BIT #40000,@SWR
2870 017224 032777 040000 160742 BNE SCOPEB ;SCOPE BIT IS A ONE
2871 017232 001003 MOV @%6,RETURN ;NO - SAVE %7 FOR NEXT TIME
2872 017234 011667 000076 RTI ;RETURN IN SEQUENCE
2873 017240 000002 SCOPEB: CMP (6)+,%6 ;REPOSITION THE STACK
2874 017242 022606 MOV (6)+,PSW
2875 017244 012667 160526 JMP @RETURN ;SCOPE RETURN
2876 017250 000177 000062
2877
2878 017254 SCOPEC: JSR %7,CKSWR ;CHECK FOR CONT-G
2879 017260 032777 040000 160706 BIT #40000,@SWR
2880 ;TEST SR FOR SCOPE
2881 017266 001365 BNE SCOPEB ;SCOPE OR/AND ITERATION LOOP FOR EACH TEST 400
2882 017270 004767 000126 JSR %7,CKSWR ;YES SCOPE
2883 017274 032777 004000 160672 BIT #4000,@SWR ;CHECK FOR CONT-G
2884 017302 001007 BNE SCOPEC ;NO - TEST FOR ITERATION
2885 017304 026727 000024 004000 CMP SCOPEF,#4000 ;INHIBIT ITERATION
2886 017312 001403 BEQ SCOPEF
2887 017314 005267 000014 INC SCOPEF ;EXIT - DONE
2888 017320 000750 BR SCOPEB ;INCREMENT COUNT
2889 017322 005067 000006 SCOPEG: CLR SCOPEF ;LOOP SOME MORE
2890 017326 011667 000004 MOV @%6,RETURN ;CLEAR COUNT
2891 017332 000002 RTI ;SAVE SCOPE RETURN POINTER
2892 017334 000000 SCOPEF: 0 ;RETURN INLINE-NEXT TEST
2893 017336 000360 RETURN: BEGIN ;COUNT LOCATION FOR ITERATION LOOP
2894 017340 000 EOMK: .BYTE 0 ;ADDRESS OF LAST TEST
2895 017342 000167 160632 .EVEN
2896 017346 000040 JMP 200
2897 017406 000000 .BLKB 40
2898 017410 000000 BUFF: 0 ;FOR STACK POINTER 40 LOCATIONS
2899 017410 000000 CP: 0
2900
2901 ;*****
2902 ; CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ↑G TO ALLOW CHANGING
2903 ; OF LOC.176.
2904 ; LOCATIONS USED:
2905 ;*****
2906
2907
2908
2909
2910 017412 000000 TEMPST: .WORD 0
2911 017414 000000 COUNT: .WORD 0
2912 017416 000000 RDSW: .WORD 0
2913 017420 000000 TIB: .WORD 0
2914

```

2915	017422				CKSWR:	CMP	#SWREG, SWR		; SOFTWARE SWITCH REGISTER PRESENT
2916	017422	022767	000176	160544		BNE	OUT		; NO GET OUT
2917	017430	001131				TSTB	@TKS		; YES WAIT FOR
2918	017432	105777	177254			BPL	OUT		; READY, GET CHARACTER
2919	017436	100126				MOV	@TKB, TIB		; AND STRIP OFF
2920	017440	017767	177250	177752		BIC	#177600, TIB		; THE GARBAGE
2921	017446	042767	177600	177744		CMP	#7, TIB		; IS IT A <IG>
2922	017454	022767	000007	177736		BNE	OUT		
2923	017462	001114				MOV	#SCNTG, %5		
2924	017464	012705	017770			JSR	%7, TTOUT		
2925	017470	004767	000430		CNTLU:	MOV	#MSWR, %5		
2926	017474	012705	020002			JSR	%7, TTOUT		
2927	017500	004767	000420			MOV	@SWR, R2		
2928	017504	017702	160464			JSR	%7, PRTAB		
2929	017510	004767	177202			MOV	#SNEW, %5		
2930	017514	012705	020012			JSR	%7, TTOUT		
2931	017520	004767	000400		\$READ:	CLR	TEMPST		
2932	017524	005067	177662			MOV	#7, COUNT		
2933	017530	012767	000007	177656	1\$:	JSR	%7, TTIN		; GO READ A CHARACTER
2934	017536	004767	000154			BIC	#177600, TIB		; STRIP OFF GARBAGE
2935	017542	042767	177600	177650		CMF	#25, TIB		; IS IT A IU?
2936	017550	122767	000025	177642		BNE	2\$		; BRANCH IF NOT
2937	017556	001001			3\$:	BR	CNTLU		; START OVER
2938	017560	000745			2\$:	CMPB	#15, TIB		; IS IT A <CR>?
2939	017562	122767	000015	177630		BNE	4\$		; BRANCH IF NOT
2940	017570	001011				MOV	#SCRLF, %5		
2941	017572	012705	017776			JSR	%7, TTOUT		
2942	017576	004767	000322			CMP	#7, COUNT		; WAS IT FIRST CHARACTER
2943	017602	022767	000007	177604		BNE	7\$		; CHANGE SWR IF NOT FIRST ONE
2944	017610	001036				BR	OUT		; GET OUT
2945	017612	000440			8\$:	BR	OUT		
2946	017614	122767	000060	177576	4\$:	CMPB	#60, TIB		
2947	017622	003004				BGT	5\$		
2948	017624	122767	000067	177566		CMPB	#67, TIB		
2949	017632	002005				BGE	6\$		
2950	017634	012705	020023		5\$:	MOV	#SQUEST, %5		
2951	017640	004767	000260			JSR	%7, TTOUT		
2952	017644	000745				BR	3\$		; START OVER IF NOT LEGAL CHARACTER
2953	017646	006367	177540		6\$:	ASL	TEMPST		
2954	017652	006367	177534			ASL	TEMPST		
2955	017656	006367	177530			ASL	TEMPST		
2956	017662	142767	000060	177530		BICB	#60, TIB		; GET NITTY-GRITTY
2957	017670	156767	177524	177514		BISB	TIB, TEMPST		
2958	017676	005367	177512			DEC	COUNT		; ONLY WANT 6 DIGITS
2959	017702	001754				BEQ	5\$		
2960	017704	000714				BR	1\$		
2961	017706	016777	177500	160260	7\$:	MOV	TEMPST, @SWR		; CHANGE SWITCH REGISTER CONTENTS
2962	017714	000207			OUT:	RTS	%7		; RETURN TO PROGRAM



3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025  
3026  
3027  
3028  
3029  
3030  
3031  
3032  
3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041

020124  
020124 105715  
020126 001403  
020130 122715 000046  
020134 001005  
020136 042777 000100 176540  
020144 005005  
020146 000207  
020150 122715 000137  
020154 001411  
020156 122715 000041  
020162 001414  
020164  
020164 105777 176514  
020170 100375  
020172 112577 176504  
020176 000752  
020200 005205  
020202 010567 000020  
020206 012705 020222  
020212 000767  
020214 016705 000006  
020220 000741  
020222 015 012 041  
020226  
000000  
000001

```
*****  
TTY ASCII OUTPUT ROUTINE  
*****  
TTOUT:  
TSTB (5) ; CHECK FOR NULL CHARACTER  
BEQ 15 ; IF NOT, TYPE THE CHARACTER  
CMPB #'&(5) ; CHECK FOR TERMINATOR  
BNE .EMPTY  
BIC #100,@TPS  
CLR %5 ; CLEAR POINTER TO CHARACTER  
RTS ; RETURN  
RTS ; CRLF CHAR?  
.EMPTY: CMPB #'+'(5)  
BEQ .RET ; CHECK FOR RETURN TERMINATOR  
CMPB #'!'(5)  
BEQ .REST  
15:  
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 ; SET UP NEW POINTER  
MOV %5.SAV  
MOV #.RETR,%5  
BR .1  
.REST: MOV .SAV,%5  
BR TTOUT  
3038 020222 015 012 041 .RETR: .BYTE 15,12,'!  
3039 020226 .EVEN  
3040 000000 .SAV: 0  
3041 000001 .END
```

AC	000244	71*	102*	150*	151	162*	163	181*	182	188	194	204	214	220
		225	261*	287	304*	316*	328*	341*	353*	366*	380*	392*	404*	417*
		430*	443*	455*	468*	481*	494*	510*	512	521*	523	534*	536	547*
		549	558*	560	571*	573	583*	585	596*	598	607*	609	620*	622
		633*	635	645*	647	656*	658	669*	671	682*	684	695*	697	712*
		905*	907	917*	923	930*	936	943*	949	954*	960	967*	973	980*
		986	993*	999	1005*	1011	1017*	1023	1030*	1036	1043*	1049	1055*	1061
		1068*	1074	1080*	1086	1097*	1099	1116*	1118	1133*	1135	1149*	1151	1168*
		1170	1186*	1188	1203*	1205	1221*	1223	1239*	1241	1256*	1258	1275*	1277
		1293*	1295	1309*	1311	1325*	1327	1341*	1343	1357*	1359	1375*	1377	1393*
		1395	1409*	1411	1431*	1433	1449*	1451	1465*	1467	1484*	1486	1503*	1505
		1520*	1522	1537*	1539	1555*	1557	1571*	1573	1589*	1591	1607*	1609	1623*
		1625	1641*	1643	1659*	1661	1675*	1677	1693*	1695	1711*	1713	1740*	1742
		1759*	1761	1777*	1779	1793*	1795	1812*	1814	1834*	1836	1850*	1852	1867*
		1869	1884*	1886	1900*	1902	1917*	1919	1933*	1935	1950*	1952	1974*	1976
		1993*	1995	2010*	2012	2030*	2032	2049*	2051	2067*	2069	2087*	2089	2133
		2150	2168	2183	2201	2218	2241	2254*	2260	2272*	2274	2291*	2293	2310*
		2312	2326*	2328	2345*	2347	2364*	2366	2380*	2382	2399*	2401	2418*	2420
		2434*	2436	2453*	2455	2471*	2473	2487*	2489	2506*	2508	2522*	2524	2538*
		2540	2574	2600	2630	2766								
ACMQ	007416	1426*												
ACIL	002444	507*												
ACIOL	004530	902*												
ASCNT	017204	2808*	2833*	2840*	2845*	2857*								
ASH	000262	78*	1741*	1760*	1778*	1794*	1813*	1835*	1851*	1868*	1885*	1901*	1918*	1934*
		1951*												
ASL	011532	1831*												
ASR	011140	1735*												
BEGIN	000360	97	99	101*	2718	2731	2893							
BINCT	017202	2804*	2825*	2828*	2829	2832*	2856*							
BUFF	017406	61	2898*											
CKSWR	017422	250	2120	2680	2710	2739	2745	2781	2786	2792	2869	2878	2882	2915*
CNTLU	017474	100	2926*	2938										
COUNT	017414	2911*	2933*	2943	2958*									
C	017410	173*	174*	176	177	181	182	2899*						
CP1	000636	174*	183											
DECML	017176	2810*	2824	2830*	2854*									
DIV	000242	70*	2236*	2255*	2273*	2292*	2311*	2327*	2346*	2365*	2381*	2400*	2419*	2435*
		2454*	2472*	2488*	2507*	2523*	2539*	2585*	2586*	2587*	2588*	2589*	2590*	2591*
		2592*	2593*	2594*	2595*	2596*	2597*	2598*	2616*	2618*	2620*	2622*	2624*	
		2233*												
DIVIDE	013522	2698	2703*											
DOAGN	016232	2894*												
EOMK	017340	2846	2848*											
HDFHM	017162	52*												
HLT =	104000	190	113	122	130	135	142	147	153	158	165	169	179	184
		289	196	202	206	212	216	222	227	231	237	243	269	273
		375	293	308	312	320	324	332	336	345	349	357	361	371
		459	384	388	396	400	408	412	421	425	434	438	447	451
		542	463	472	476	485	489	498	502	514	518	525	529	538
		624	551	555	562	566	575	579	587	591	600	604	611	615
		703	628	637	641	649	653	660	664	673	677	686	690	699
		787	716	720	729	733	741	745	753	757	764	768	775	779
		861	791	799	803	810	814	822	826	834	838	846	850	857
		947	869	873	881	885	893	897	909	913	921	925	934	938
		1025	951	958	962	971	975	984	988	997	1001	1009	1013	1021
			1034	1038	1047	1051	1059	1063	1072	1076	1084	1088	1101	1105

1109	1120	1124	1128	1137	1141	1145	1153	1157	1161	1172	1176	1180
1190	1194	1198	1207	1211	1215	1225	1229	1233	1243	1247	1251	1260
1264	1268	1279	1283	1287	1297	1301	1305	1313	1317	1321	1329	1333
1337	1345	1349	1353	1361	1365	1369	1379	1383	1387	1397	1401	1405
1413	1417	1421	1435	1439	1443	1453	1457	1461	1469	1473	1477	1488
1492	1496	1507	1511	1515	1524	1528	1532	1541	1545	1549	1559	1563
1567	1575	1579	1583	1593	1597	1601	1611	1615	1619	1627	1631	1635
1645	1649	1653	1663	1667	1671	1679	1683	1687	1697	1701	1705	1715
1719	1723	1744	1748	1752	1763	1767	1771	1781	1785	1789	1797	1801
1805	1816	1820	1824	1838	1842	1846	1854	1858	1862	1871	1875	1879
1888	1892	1896	1904	1908	1912	1921	1925	1929	1937	1941	1945	1954
1958	1962	1978	1982	1987	1997	2001	2006	2014	2018	2023	2034	2038
2043	2053	2057	2062	2071	2075	2080	2091	2095	2100	2135	2139	2143
2152	2156	2160	2170	2174	2178	2185	2189	2193	2203	2207	2211	2220
2224	2228	2239	2245	2249	2258	2264	2268	2276	2280	2284	2295	2299
2303	2314	2318	2322	2330	2334	2338	2349	2353	2357	2368	2372	2376
2384	2388	2392	2403	2407	2411	2422	2426	2430	2438	2442	2446	2457
2461	2465	2475	2479	2483	2491	2495	2499	2510	2514	2518	2526	2530
2534	2542	2546	2550	2572	2576	2580	2602	2606	2610	2628	2632	2636
2647	2651	2664	2668									
2806	2858*											
264*	276											
284*	296											
77*	266*	286*	305*	317*	329*	342*	354*	367*	381*	393*	405*	418
431*	444*	456*	469*	482*	495*	511*	522*	535*	548*	559*	572*	584*
597*	608*	621*	634*	646*	657*	670*	683*	696*	713*	726*	738*	750*
761*	772*	794*	796*	807*	819*	831*	843*	854*	866*	878*	890*	906*
918*	931*	944*	955*	968*	981*	994*	1006*	1018*	1031*	1044*	1056*	1069*
1081*	1098*	1117*	1134*	1150*	1169*	1187*	1204*	1222*	1240*	1257*	1276*	1294*
1310*	1326*	1342*	1358*	1376*	1394*	1410*	1432*	1450*	1466*	1485*	1504*	1521*
1538*	1556*	1572*	1590*	1608*	1624*	1642*	1660*	1676*	1694*	1712*		
2815	2818*											
2824*	2827											
2836*	2838											
72*	103*	127*	128	139*	140	176*	177	187*	193*	199*	200	209*
210	219*	224*	229	262*	267	303*	306	315*	318	327*	330	340*
343	352*	355	365*	369	379*	382	391*	394	403*	406	416*	419
429*	432	442*	445	454*	457	467*	470	480*	483	493*	496	509*
711*	714	725*	731	737*	743	749*	755	760*	766	771*	777	783*
789	795*	801	806*	812	818*	824	830*	836	842*	848	853*	859
865*	871	877*	883	889*	895	904*	1096*	1103	1115*	1122	1132*	1139
1148*	1155	1167*	1174	1185*	1192	1202*	1209	1220*	1227	1238*	1245	1255*
1262	1274*	1281	1292*	1299	1308*	1315	1324*	1331	1340*	1347	1356*	1363
1374*	1381	1392*	1399	1408*	1415	1430*	1437	1448*	1455	1464*	1471	1483*
1490	1502*	1509	1519*	1526	1536*	1543	1554*	1561	1570*	1577	1588*	1595
1606*	1613	1622*	1629	1640*	1647	1658*	1665	1674*	1681	1692*	1699	1710*
1717	1739*	1746	1758*	1765	1776*	1783	1792*	1799	1811*	1818	1833*	1840
1849*	1856	1866*	1873	1883*	1890	1899*	1906	1916*	1923	1932*	1939	1949*
1956	1973*	1980	1992*	1999	2009*	2016	2029*	2036	2048*	2055	2066*	2073
2086*	2093	2131*	2137	2148*	2154	2166*	2172	2181*	2187	2199*	2205	2216*
2222	2235*	2253*	2271*	2278	2290*	2297	2309*	2316	2325*	2332	2344*	2351
2363*	2370	2379*	2386	2398*	2405	2417*	2424	2433*	2440	2452*	2459	2470*
2477	2486*	2493	2505*	2512	2521*	2528	2537*	2544	2553*	2570	2583*	2604
2613*	2626	2639	2655	2771								
1093*												
252	259*											

LIST 017206  
LOOP 001172  
LOOP1 001242  
LSH 000260

MINUS 017004  
MKNUM 017034  
MOADD 017116  
MQ 000246

MQAC 005530  
MQOL 001154







B.4	2#	112	121	129	134	141	146	152	157	164	168	178	189	195	201
	205	211	215	221	226	230	236	242	268	272	288	292	307	311	319
	323	331	335	344	348	356	360	370	374	383	387	395	399	407	411
	420	424	433	437	446	450	458	462	471	475	484	488	497	501	513
	517	524	528	537	541	550	554	561	565	574	578	586	590	599	603
	610	614	623	627	636	640	648	652	659	663	672	676	685	689	698
	702	715	719	728	732	740	744	752	756	763	767	774	778	786	790
	798	802	809	813	821	825	833	837	845	849	856	860	868	872	880
	884	892	896	908	912	920	924	933	937	946	950	957	961	970	974
	983	987	36	1000	1008	1012	1020	1024	1033	1037	1046	1050	1058	1062	1071
	1075	1083	1087	1100	1104	1108	1119	1123	1127	1136	1140	1144	1152	1156	1160
	1171	1175	1179	1189	1193	1197	1206	1210	1214	1224	1228	1232	1242	1246	1250
	1253	1263	1267	1278	1282	1286	1296	1300	1304	1312	1316	1320	1328	1332	1336
	1344	1348	1352	1360	1364	1368	1378	1382	1386	1396	1400	1404	1412	1416	1420
	1434	1438	1442	1452	1456	1460	1468	1472	1476	1487	1491	1495	1506	1510	1514
	1523	1527	1531	1540	1544	1548	1558	1562	1566	1574	1578	1582	1592	1596	1600
	1610	1614	1618	1626	1630	1634	1644	1648	1652	1662	1666	1670	1678	1682	1686
	1696	1700	1704	1714	1718	1722	1743	1747	1751	1762	1766	1770	1780	1784	1788
	1796	1800	1804	1815	1819	1823	1837	1841	1845	1853	1857	1861	1870	1874	1878
	1887	1891	1895	1903	1907	1911	1920	1924	1928	1936	1940	1944	1953	1957	1961
	1977	1981	1986	1996	2000	2005	2013	2017	2022	2033	2037	2042	2052	2056	2061
	2070	2074	2079	2090	2094	2099	2134	2138	2142	2151	2155	2159	2169	2173	2177
	2184	2188	2192	2202	2206	2210	2219	2223	2227	2275	2279	2283	2294	2298	2302
	2313	2317	2321	2329	2333	2337	2348	2352	2356	2367	2371	2375	2383	2387	2391
	2402	2406	2410	2421	2425	2429	2437	2441	2445	2456	2460	2464	2474	2478	2482
	2490	2494	2498	2509	2513	2517	2525	2529	2533	2541	2545	2549	2571	2575	2579
	2601	2605	2609	2627	2631	2635	2646	2650	2663	2667	2671	2675	2683	2687	2691
STARS	2#	3	17	32	82	84	105	107	246	249	255	258	279	281	298
	300	504	506	705	707	899	901	1090	1092	1423	1425	1726	1728	1730	1734
	1828	1830	1964	1966	1967	1969	2105	2107	2114	2117	2124	2126	2230	2232	2673
	2675	2676	2678	2691	2693	2707	2709	2863	2866	2903	2907	2966	2968	3009	3011
WATTPS	2#	2684	2743	2753	2756	2764	2769	2774	2812	2842	2849	2981	3027		
.HEADE	24	5													

. ABS. 020230 000

ERRORS DETECTED: 0

DZKEBB.BIN,DZKEBB.LST/CRF/SOL/NL:TOC=DZKEBB.P11  
RUN-TIME: 3 7 .7 SECONDS  
RUN-TIME RATIO: 49/11=4.1  
CORE USED: 7K (13 PAGES)

