

# KD11-Z

ROM DIAG (11/44 UBI)  
CKKFAA0

AH-F640A-MC  
COPYRIGHT 1980  
FICHE 1 OF 1

JAN 1980  
**digital**  
MADE IN USA



.REM %

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  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45

IDENTIFICATION

PRODUCT CODE:	AC-F639A-MC
PRODUCT NAME:	CKKFAAO ROM DIAG(11/44 UBI)
DATE CREATED:	OCT 1979
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	JOHN W. CIUKAJ

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 FAULTS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1979, BY DIGITAL EQUIPMENT CORPORATION

47  
48  
49  
50  
51  
52  
53

HISTORY SECTION

CKKFAAD WAS RELEASED OCT 1979

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111

BOOT ROM DIAGNOSTIC- 11/44

OVERVIEW

THE 11/44 BOOT DIAGNOSTIC ROM IS DESIGNED TO PROVIDE DIAGNOSTIC ROUTINES FOR SOME BASIC CPU,CACHE, & MEMORY GO-NO-GO TESTS.

IT MAY BE IMPLEMENTED IN THE FOLLOWING MANNER:

1. AT POWER UP
2. DEPRESSING THE FRONT PANEL BOOT SWITCH
3. ISSUING THE BOOT COMMAND FROM THE MFM CONSOLE
4. LOAD ADDRESS AND START SEQUENCE

THE 11/44 BOOT DIAGNOSTIC IS AN INTEGRAL PART OF THE UBI BOOTSTRAPPING SYSTEM .THE SYSTEM ALLOWS FOR:

- A. BOOT OR POWER UP TO MFM CONSOLE WITH BOOT DIAGNOSTIC SELECTED,
- B. BOOT OR POWER UP TO MFM CONSOLE WITHOUT BOOT DIAGNOSTIC SELECTED,
- C. BOOT OR POWER UP TO PRIMARY BOOTSTRAP ROM WITH BOOT DIAGNOSTIC SELECTED,
- D. BOOT OR POWER UP TO PRIMARY BOOTSTRAP ROM WITHOUT BOOT DIAGNOSTIC SFLECTED.

2. UBI BOOTSTRAPPING SYSTEM (FIGURE 1)

THE PRIMARY ACTIVATING PROCESSES FOR THE BOOTSTRAP SYSTEM ARE EITHER A POWER UP SEQUENCE OR THE ACTUATING OF THE FRONT PANEL BOOT SWITCH.

TO ACTIVATE THE BOOTSTRAPPING PROCESS ON POWER UP SWITCH S1 (M7095 MODULE) IN THE CPU CONTROL LOGIC MUST BE IN THE ON POSITION. IF THIS SWITCH IS OFF THEN A NORMAL TRAP TO LOCATION 24 TO EXECUTE THE USER POWER UP ROUTINE WILL OCCUR. WHEN THIS SWITCH IS ON, THE CPU MICRO CODE ASSERTS 773024 ON THE BUS, SWITCH E28-1,E28-2 ,TOGETHER WITH E28-3 THRU 10, ON THE UBI WILL DETERMINE WHAT ACTION WILL FOLLOW AS THE POWER UP SEQUENCE CONTINUES.

WHEN THE CONSOLE BOOT SWITCH IS ACTUATED, A SOFTWARE ONLY POWER DOWN FOLLOWED BY A BOOT POWER UP OCCURS. NOTE THAT THE POSITION OF S1 IS IRRELEVANT IN THIS CASE.

773024 WILL BE ASSERTED ON THE UNIBUS ADDRESS LINES. LIKE THE NORMAL POWER UP DESCRIBED ABOVE, SWITCH E28-1, E28-2,TOGETHER WITH E28-3 THRU WILL DETERMINE WHAT ACTION WILL FOLLOW.

112  
113  
114

(CONSULT SHEET 11 M7098 MODULE FOR SWITCH SETTINGS E28-1  
THRU E28-10)

121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164

3. BOOT DIAGNOSTIC ROM

THERE ARE THREE DIFFERENT TYPES OF TESTS INCLUDED IN THE BOOT DIAGNOSTIC ROM :

- 1 CPU TESTS
- 2 MEMORY TEST
- 3 CACHE TEST

THE CPU TESTS ARE TESTS OF MOST UNARY AND DOUBLE OPERAND INSTRUCTIONS USING REGISTER MODES 0,1 AND 4. EXCEPT TESTS THAT USE THE STACK POINTER, MAIN MEMORY IS NOT MODIFIED. IF A FAILURE IS DETECTED, A 'BR .' WILL OCCUR.

THE MEMORY TEST PERFORMS BOTH A DUAL ADDRESSING AND DATA CHECK OF ALL THE AVAILABLE MEMORY FROM ADDRESS 1000 TO 28K WITH CACHE DISABLED. THIS TEST WILL CHANGE ALL OF MEMORY TESTED. WHEN AN ERROR IS DETECTED, 'ERROR' IS PRINTED TO THE CONSOLE, FOLLOWED BY A CPU HALT. MFM CONSOLE MODE WILL BE ENTERED DUE TO THE HALT WITH THE FAILING PC BEING PRINTED. AN MFM CONSOLE 'CONTINUE' WILL ALLOW CONTINUATION OF THE BOOT ROM DIAGNOSTIC. WHEN AN ERROR IS INCURRED, R0 WILL CONTAIN THE ADDRESS AT WHICH THE FAILURE WAS DETECTED. R1 WILL CONTAIN THE FAILING DATA PATTERN AND R0 WILL CONTAIN THE EXPECTED DATA PATTERN.

A CACHE TEST IS PERFORMED TO CHECK BOTH DUAL ADDRESSING AND CACHE DATA MEMORY. THE CACHE DATA STORE AND MAIN MEMORY LOCATIONS WILL BE WRITTEN WITH THEIR OWN ADDRESS. CACHE UPDATES AND HITS USING MAIN MEMORY FROM VIRTUAL ADDRESS 1000 TO HIGHEST AVAILABLE MEMORY LESS THAN 28K ARE ALLOWED. IT MUST BE NOTED THAT THE TEST DOES NOT GUARANTEE THAT DATA RECEIVED IS ACTUALLY COMING FROM CACHE DUE TO THE CACHE CPL CLOCK RESTART OCCURING. TO GET THE CLOSEST ASSURANCE THAT THIS IS HAPPENING, CACHE HITS ARE CHECKED VIA THE 'HIT' MAINTENANCE BIT IN THE CACHE MAINTENANCE REGISTER. WHEN AN ERROR OCCURS AND MFM 'CONTINUE' IS USED, ERROR HANDLING IS SIMILAR TO MEMORY TEST WHEN DATA ERRORS ARE INCURRED. BUT A 'CONTINUE' AFTER A HIT ERROR WILL RESULT WITH THE CACHE TEST BEING ABORTED AND THE BOOT ROM DIAGNOSTIC EXITING.

NOTE: IF CACHE IS NOT FOUND, THE CACHE TEST IS NOT PERFORMED.

166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222

4.0 PRINTOUT EXAMPLES

-----  
CONDITIONS: THE FOLLOWING EXAMPLES ARE ILLUSTRATIONS OF TYPEOUTS TO THE TERMINAL WITH THE UBI MICRO SWITCHES E28-2 THRU 10 SELECTED FOR 'POWERUP/BOOT TO CONSOLE WITH DIAGNOSTIC' (USW=020) (SEE FIG. 1)

- 1. ERROR OCCURS IN BOOT DIAGNOSTIC CACHE TEST WHEN POWERUP OCCURS:

ERROR  
CONSOLE  
17777707 165652  
>>>

- 2. ERROR OCCURS WHEN BOOT COMMAND IS ISSUED FROM CONSOLE

>>>B  
ERROR  
CONSOLE  
17777707 165652  
>>>

- 3. NO ERROR OCCURS; BOOT DIAGNOSTIC RUNS SUCCESSFULLY:

>>>B  
CONSOLE  
17777707 165714  
>>>

5.0 LOAD ADDRESS AND START PROCEDURE

-----  
THE BOOT DIAGNOSTIC MAY BE RUN BY LOADING ADDRESS 165020 USING THE MFM CONSOLE AND STARTING. THE FOLLOWING EXAMPLE EXHIBITS THIS AND ILLUSTRATES SUCCESSFUL COMPLETION OF THE DIAGNOSTIC.

223  
224  
225  
226  
227  
228  
229  
230

>>>S 165020

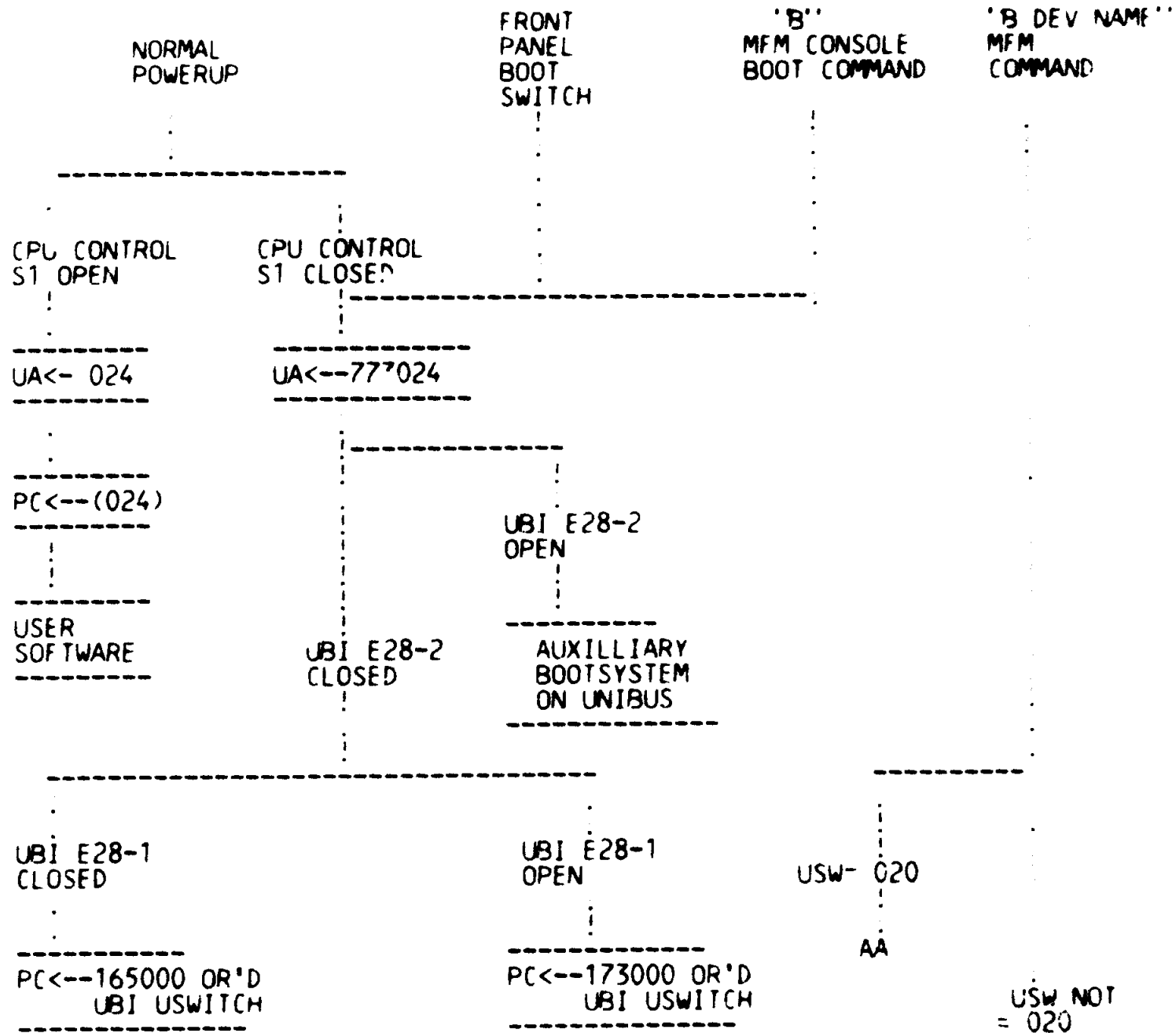
CONSOLE  
17777707 165714  
>>>



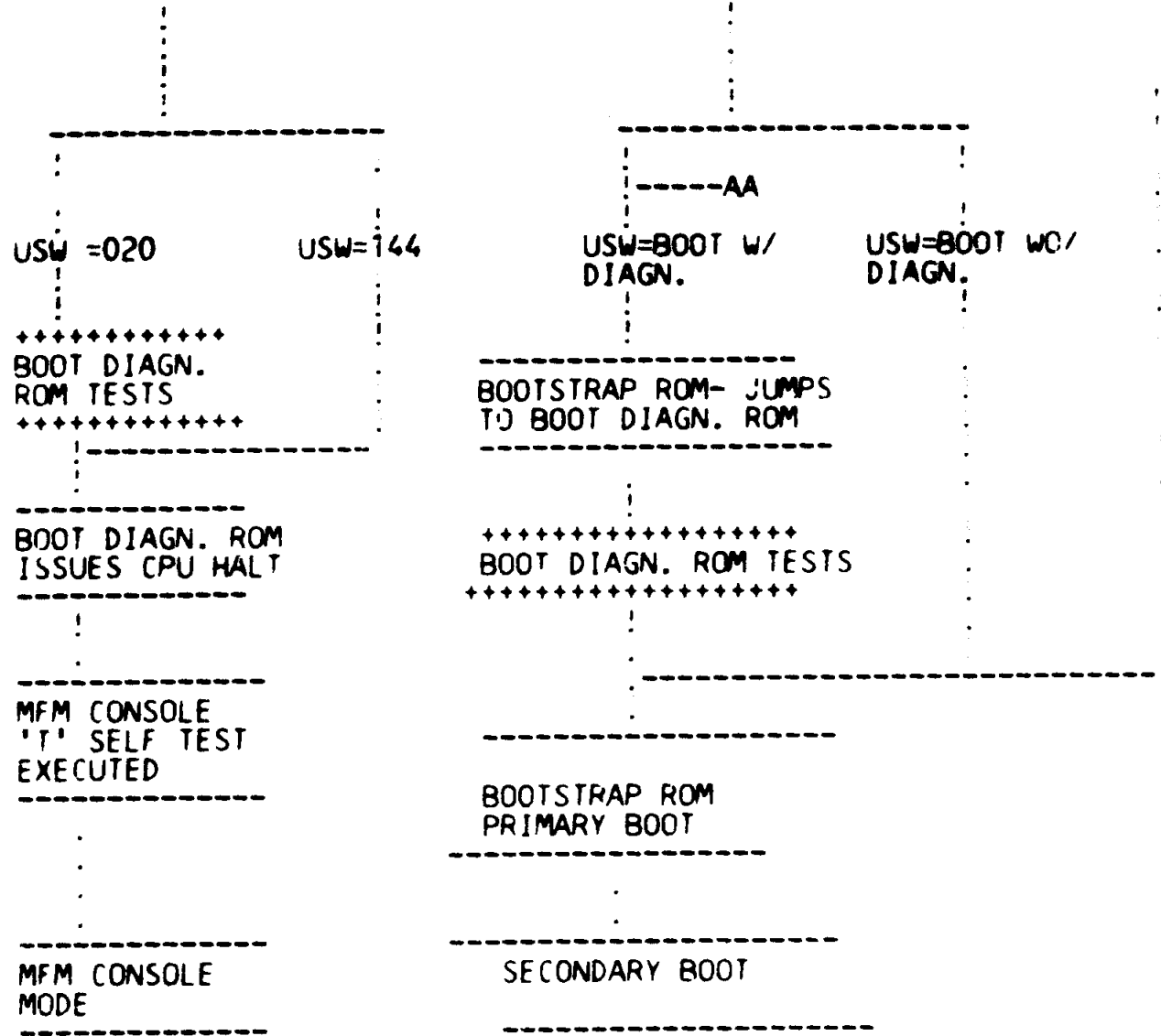
11/44 JBI BOOTSTRAP SYSTEM- FIG. 1

- NOTES: 1. CLOSED=ON  
 2. OPEN =OFF  
 3. USW- UBI E28-3 THRU 10 MICRO SWITCH SETTINGS (LOCATED ON UBI MODULE)  
 4. UA- UNIBUS ADDRESS  
 5. S1 IS LOCATED ON KE44-1 DATA PATH (M7094)

232  
 233  
 234  
 235  
 236  
 237  
 238  
 239  
 240  
 241  
 242  
 243  
 244  
 245  
 246  
 247  
 248  
 249  
 250  
 251  
 252  
 253  
 254  
 255  
 256  
 257  
 258  
 259  
 260  
 261  
 262  
 263  
 264  
 265  
 266  
 267  
 268  
 269  
 270  
 271  
 272  
 273  
 274  
 275  
 276  
 277  
 278  
 279  
 280  
 281  
 282  
 283  
 284  
 285  
 286  
 287  
 288



289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327



```

329
330
331 000000          .ENABL ABS
332 001015          MISS=1015          ;UNCONDITIONAL BYPASS, FORCE MISS LO
333                                     ;FORCE MISS HI, DISABLE CACHE INTERRUPT
334                                     ;USED FOR CACHE CONTROL REGISTER
335          104000  ERROR=104000
336          165000  .-165000
337 165000  012737  177777  000706  ENTBT: MOV    #-1, @#706          ; BOOT/POWERUP TO PERIPHERAL
338                                     ;ROM WITH DIAGNOSTIC. SET FLAG.
339                                     ;JUMPED TO HERE FROM 165564
340 165006  000405          BR      SAVE
341 165010  005037  000706  START: CLR    @#706          ;CLEAR FLAG
342 165014  000402          BR      SAVE
343 165016  000240          NOP
344 165020  000773          BR      START          ;ENTRY: BOOT/POWERUP TO CONSOLE WITH DIAGNOSTIC
345 165022  010037  000700  SAVE:  MOV    R0, @#700          ;SAVE R0, R1, R4 CONTENTS.
346 165026  010137  000702          MOV    R1, @#702
347 165032  010437  000704          MOV    R4, @#704
348 165036  012737  165676  000004          MOV    #CPUERR, @#4          ;SETUP FOR ANY CPU ERRORS
349 165044  012737  000340  000006          MOV    #340, @#6
350 165052  012737  165736  000030          MOV    #SERR, @#30          ;SETUP ERROR PRINT ROUTINE
351 165060  012737  000340  000032          MOV    #340, @#32
352
353          ;*****
354          .SBTTL TEST1  THIS TEST VERIFIES THE UNCONDITIONAL BRANCH
355
356          ;*
357          ;*  THE REGISTERS AND CONDITION CODES ARE ALL UNDEFINED WHEN
358          ;*  THIS TEST IS ENTERED AND THEY SHOULD REMAIN THAT WAY UPON
359          ;*  THE COMPLETION OF THIS TEST.
360          ;*****
361          TST1:
362 165066  000401          BR      TST2          ; * BRANCH ALWAYS
363 165070  000777          BR      .
364
365          ;*****
366          .SBTTL TEST2  TEST 'CLR', MODE '0', AND 'BMI', 'BVS', 'BHI', 'BLT', 'BLOS'
367
368          ;*
369          ;*  THE REGISTERS AND CONDITION CODES ARE ALL UNDEFINED WHEN
370          ;*  THIS TEST IS ENTERED. UPON COMPLETION OF THIS TEST THE 'SP'
371          ;*  (R6) SHOULD BE ZERO AND ONLY THE 'Z' FLIP-FLOP WILL BE SET.
372          ;*****
373          TST2:
374
375 165072  005006          CLR    SP          ;N=0, Z=1, V=0, C=0, SP=000000
376 165074  100404          BMI    1$          ; V BRANCH IF N=1
377 165076  102403          BVS    1$          ; V BRANCH IF V=1
378 165100  101002          BHI    1$          ; V BRANCH IF Z AND C ARE BOTH 0
379 165102  002401          BLT    1$          ; V BRANCH IF (N XOR V)=1
380 165104  101401          BLOS   TST3          ; * BRANCH IF (Z XOR C)=0
381 165106  000777          1$:  BR      .
382
383          ;*****
384          .SBTTL TEST3  TEST 'DEC', MODE '0', AND 'BPL', 'BEQ', 'BGE', 'BLE'
385

```

MACRO M1111 26-SEP-79 10:28 PAGE 7-1  
TEST3 TEST 'DEC', MODE '0', AND 'BPL', 'BEQ', 'BGE', 'BLE'

386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396 165110  
397 165110 005306  
398 165112 1000C3  
399 165114 001402  
400 165116 002001  
401 165120 003401  
402 165122 00C777  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417 165124  
418 165124 006006  
419 165126 102002  
420 165130 103001  
421 165132 001001  
422 165134 000777  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439 165136  
440 165136 012706 125252  
441 165142 000402  
442 165144 000000

UPON ENTERING THIS TEST THE CONDITION CODES ARE:  
N = 0, Z = 1, V = 0, AND C = 0.  
THE REGISTERS ARE: R0 = ?, R1 = ?, R2 = ?  
R3 = ?, R4 = ?, R5 = ?, SP = 000000  
UPON COMPLETION OF THIS TEST THE CONDITION CODES WILL BE:  
N = 1, Z = 0, V = 0, AND C = 0  
THE REGISTERS AFFECTED BY THE TEST ARE:  
SP = 177777

\*\*\*\*\*  
TST3:  
DEC SP ;N=1,Z=0,V=0,C=0,SP=177777  
BPL 1\$ ; V BRANCH IF N=0  
BEQ 1\$ ; V BRANCH IF Z=1  
BGE 1\$ ; V BRANCH IF (N XOR V)=0  
BLE TST4 ; \* BRANCH IF (Z OR (N XOR V)) 1  
1\$: BR .

\*\*\*\*\*  
.SBTTL TEST4 TEST 'ROR', MODE '0', AND 'BVC', 'BHIS', 'BNE'

UPON ENTERING THIS TEST THE CONDITION CODES ARE:  
N = 1, Z = 0, V = 0, AND C = 0.  
THE REGISTERS ARE: R0=?, R1=?, R2 = ?  
R3 = ?, R4 = ?, R5 = ?, SP = 177777  
UPON COMPLETION OF THIS TEST THE CONDITION CODES WILL BE:  
N = 0, Z = 0, V = 1, AND C = 1  
THE REGISTERS AFFECTED BY THE TEST ARE:  
SP = 077777

\*\*\*\*\*  
TST4:  
ROR SP ;N=0,Z=0,V=1,C=1,SP=077777  
BVC 1\$ ; V BRANCH IF V=0  
BHIS 1\$ ; V BRANCH IF C=0  
BNE TST5 ; \* BRANCH IF Z=0  
1\$: BR .

\*\*\*\*\*  
.SBTTL TEST5 TEST REGISTER DATA PATH

WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:  
N = 0, Z = 0, V = 1, AND C = 1.  
THE REGISTERS ARE: R0 = ?, R1 = ?, R2 = ?  
R3 = ?, R4 = ?, R5 = ?, SP = 077777.  
UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:  
N = 0, Z = 1, V = 0 AND C = 0.  
THE REGISTERS ARE LEFT AS FOLLOWS:  
R0 = 125252, R1 = 000000, R2 = 125252, R3 = 125252  
R4 = 125252, R5 = 125252, AND SP = 125252

\*\*\*\*\*  
TST5:  
MOV #125252,SP ;N=0,Z=0,V=0,C=1,SP=125252  
BR .+6  
HALT ;ENTRY: BOOT/POWERUP TO CONSOLE WITHOUT DIAGNOSTIC

```

443 165146 000776 BR      .-2
444 165150 010600 MOV     SP,R0      ;N=0,Z=0,V=0,C=1,R0=125252
445 165152 010001 MOV     R0,R1      ;N=0,Z=0,V=0,C=1,R1=125252
446 165154 010102 MOV     R1,R2      ;N=0,Z=0,V=0,C=1,R2=125252
447 165156 010203 MOV     R2,R3      ;N=0,Z=0,V=0,C=1,R3=125252
448 165160 010304 MOV     R3,R4      ;N=0,Z=0,V=0,C=1,R4=125252
449 165162 010405 MOV     R4,R5      ;N=0,Z=0,V=0,C=1,R5=125252
450 165164 160501 SUB     R5,R1      ;N=0,Z=1,V=0,C=0, AND R1=000000
451 165166 002401 BLT     1$         ; V BRANCH IF (N XOR V)=1
452 165170 001401 BEQ     TST6       ; * BRANCH IF Z=1
453 165172 000777 BR      .

```

1\$:

```

:*****
: .SBTTL TEST6 TEST 'ROL', 'BCC', 'BLT'
:
: *
: * WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
: * N = 0, Z = 1, V = 0, AND C = 0.
: * THE REGISTERS ARE: R0 = 125252, R1 = 000000, R2 = 125252
: * R3 = 125252, R4 = 125252, R5 = 125252, SP = 125252.
: * UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
: * N = 0, Z = 0, V = 1, AND C = 1.
: * THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
: * R2 WHICH SHOULD NOW EQUAL 05252'.
:*****

```

```

468 165174 006102 TST6:  ROL     R2      ;N=0,Z=0,V=1, C=1, AND R2 = 052524
469 165174 103001 BCC     1$         ; V BRANCH IF C=0
470 165176 103001 BLT     TST7       ; * BRANCH IF (N XOR V)-1
471 165200 002401 BR      .
472 165202 000777

```

1\$:

```

:*****
: .SBTTL TEST7 TEST 'ADD', 'INC', 'COM', AND 'BCS', 'BLE'
:
: *
: * WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
: * N = 0, Z = 0, V = 1, AND C = 1.
: * THE REGISTERS ARE: R0 = 125252, R1 = 000000, R2 = 052524
: * R3 = 125252, R4 = 125252, R5 = 125252, SP = 125252
: * UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
: * N = 0, Z = 1, V = 0, AND C = 0.
: * THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
: * R3 WHICH NOW EQUALS 000000, AND R1 WHICH IS ALSO 000000
:*****

```

```

488 165204 060203 TST7:  ADD     R2,R3      ;(R2 = 052524) + (R3 = 125252)
489 165204 006203 INC     R3      ;N=1,Z=0,V=0,C=0, AND R3=177776
490 165206 005203 COM     R3      ;N=1,Z=0,V=0,C=0, AND R3=177777
491 165210 005103 ADD     R3,R1      ;N=0,Z=1,V=0,C=1, AND R3=000000
492 165212 060301 ADD     R3,R1      ;N=0,Z=1,V=0,C=0, AND R1 = 000000
493 165214 103401 BCS     1$         ; V BRANCH IF C=1
494 165216 003401 BLE     TST10      ; * BRANCH IF (Z OR (N XOR V))-1
495 165220 000777 BR      .
496 165220 000777

```

1\$:

```

:*****
: .SBTTL TEST10 TEST 'ROR', 'DEC', 'BIS', 'ADD', AND 'BLO'
:*****

```

497  
498  
499





```

557
558
559
560 165262
561 165262 000301
562 165264 020127 052125
563 165270 001004
564
565 165272 030405
566 165274 003002
567 165276 005105
568 165300 001001
569 165302 000777

```

```

;* R4 = 052525, R5 = 052525, SP = 125252.
;*
;*****
TST12:
SWAB R1 ;N=0,Z=0,V=0,C=0, AND R1 = 052125
CMP R1,#052125 ;N=0,Z=1,V=0,C=0
BNE 1$ ; V BRANCH IF Z=0
;R4 = 052525 R5 = 125252
BIT R4,R5 ;N=0,Z=1,V=0,C=0
BGT 1$ ; V BRANCH IF Z OR (N XOR V) ARE 0
COM R5 ;N=0,Z=0,V=0,C=1, AND R5 = 052525
BNE TST13 ; * BRANCH IF Z=1
1$: BR .

```

```

570
571
572
573
574
575
576
577
578
579
580
581
582
583

```

```

;*****
.SBTTL TEST13 TEST 'MOVB', 'SOB', 'CLR', 'TST' AND 'BPL', 'BNE'
;*
WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
N = 0, Z = 0, V = 0, AND C = 1.
THE REGISTERS ARE: R0 = 052525, R1 = 052125, R2 = 052524
R3 = 000000, R4 = 052525, R5 = 052525, SP = 125252.
UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
N = 0, Z = 1, V = 0, AND C = 0.
R0 IS DECREMENTED BY A SOB INSTRUCTION TO 000000
R1 IS CLEARED AND THEN INCREMENTED AROUND TO 000000
;*****

```

```

584 165304
585 165304 112700 177401
586 165310 100001
587 165312 000777
588 165314 077002
589 165316 005001
590 165320 005201
591 165322 077002
592 165324 005700
593 165326 001002
594 165330 005701
595 165332 001401
596 165334 000777
597
598
599
600
601
602
603
604
605
606
607
608

```

```

TST13:
MOVB #177401,R0 ;N=0,Z=0,V=0,C=1, AND R0 = 000001
BPL 2$ ; * BRANCH IF N=0
1$: BR . ;STOP IF 'BPL' FAILED
2$: SOB R0,1$ ;DO NOT LOOP SINCE (R0 -1) 0
CLR R1 ;N=0, Z=1, V=0, C=0, AND R1 = 000000
3$: INC R1 ;INCREMENT 64K TIMES (2 ** 16)
SOB R0,3$ ;LOOP BACK TO 'INC' 64K TIMES
TST R0 ;N=0,Z=1,V=0,C=0, AND R0 = 000000
BNE 4$ ; V BRANCH IF Z 0
TST R1 ;N=0,Z=1,V=0,C 0, AND R1 = 000000
BEQ TST14
4$: BR .

```

```

609 165336
610 165336 012706 000776
611 165342 004767 000002
612 165346 000777
613 165350 022716 165346

```

```

;*****
.SBTTL TEST14 TEST 'JSR', 'RTS', 'RTI', & 'JMP'
;*
THIS TEST FIRST SETS THE STACK POINTER TO 776,
AND THEN VERIFIES THAT 'JSR', 'RTS', 'RTI', AND 'JMP'
ALL WORK PROPERLY.
ON ENTRY TO THIS TEST THE STACK POINTER 'SP' IS INITIALIZED
TO 00776 AND IS LEFT THAT WAY ON EXIT.
;*****
TST14:
1$: MOV #776,SP ;SET UP THE STACK POINTER
JSR PC,1$ ;TRY TO JSR TO 1$
10$: BR . ;THE 'JSR' MUST HAVE FAILED
1$: CMP #10$(,SP) ;WAS THE CORRECT ADDRESS PUSHE'

```

```

614 165354 001401      BEQ      2$      ;BRANCH IF YES
615 165356 000777      BR      ;WRONG THING PUSHED ON STACK
616 165360 012716 165370 2$:  MOV      #3$, (SP) ;CHANGE THE ADDRESS ON THE STACK
617 165364 000207      RTS      PC      ;TRY TO RETURN TO 3$
618 165366 000777      BR      ;DID NOT RETURN PROPERLY
619 165370 005046      3$:  CLR      -(SP)   ;PUSH A ZERO ON THE STACK
620 165372 012746 165402  MOV      #4$, -(SP) ;PUSH THE RETURN ADDRESS ON STACK
621 165376 000002      RTI      ;SEE IF AN 'RTI' WORKS
622 165400 000777      BR      ;THE 'RTI' FAILED
623 165402 000137 165410 4$:  JMP      @#5$    ;TRY TO 'JMP'
624 165406 000777      BR      ;THE 'JMP' FAILED
625 165410      5$:      ;ADDRESS TO 'JMP' TO
626
627 ;:*****
628 ;.SBTTL TEST15 TEST MAIN MEMORY FROM VIRTUAL 001000 TO LAST ADDR.
629 ;
630 ;THIS TEST WILL TEST MAIN MEMORY WITH THE CACHE DISABLED, FROM
631 ;VIRTUAL ADDRESS 001000 TO LAST ADDR.
632 ;
633 ;IN THIS TEST THE REGISTERS ARE INITIALIZED AS FOLLOWS:
634 ;R0 - 001000, R1 = DATA READ, R2 001000, R3-177746(CACHE CONTROL REGISTER)
635 ;R5 - LAST MEMORY ADDRESS, SP - 000776
636 ;
637 ;R4 IS USED AS POINTER FOR ERROR PRINT MESSAGE
638 ;
639 ;:*****
640 ;TST15:
641 165410      TST15:
642 165410 012705 160000      MOV      #160000,R5 ;FIRST GET SIZE OF MEMORY.
643 165414 005037 000006      CLR      @#6
644 165420 012737 165426 000004 1$:  MOV      #1$, @#4
645 165426 012706 000776      MOV      #776, SP
646 165432 005745      TST      -(R5)
647 ;
648 ;IN END,R5 CONTAINS LAST MEM ADDR.
649 165434      FIX:
650 165434 012703 177746      MOV      #177746,R3 ;CACHE CONTROL REGISTER
651 165440 012737 165466 000004  MOV      #6$, @#4 ;SETUP FOR POTENTIAL TRAP IF CACHE NOT PRESENT
652 165446 012713 001015      MOV      #MISS,(R3) ;ACCESS CACHE BY DISABLING CACHE
653 165452 012737 165662 000114  MOV      #CONT,@#114 ;SET UP PARITY VECTOR
654 165460 005037 000116      CLR      @#116 ;SET PROCESSOR STATUS WORD TO ZERO
655 165464 000403      BR      7$      ;CACHE MUST BE PRESENT
656 165466 012737 165676 000004 6$:  MOV      #CPUERR,@#4 ;RESTORE TRAP VECTOR
657
658 165474 012702 001000      7$:  MOV      #1000,R2 ;FIRST ADDRESS STORAGE
659 165500 010200      MOV      R2,R0 ;SETUP FORST ADDRESS
660 165502 010010      1$:  MOV      R0,(R0) ;LOAD EACH ADDRESS WITH ITS
661 ;OWN ADDRESS
662 165504 005720      TST      (R0)+
663 165506 020005      CMP      R0,R5
664 165510 101774      BLOS    1$
665 165512 010200      MOV      R2,R0 ;SET STARTING ADDRESS IN R0
666 165514 011001      2$:  MOV      (R0),R1 ;GET THE DATA
667 165516 020001      CMP      R0,R1 ;IS IT CORRECT?
668 165520 001402      BEQ     3$      ;BRANCH IF YES
669 165522 104000      ERROR  ;GO TO PRINT ROUTINE
670 165524 000000      HALT   ;DATA ERROR ON READING MEMORY LOCATION

```

MAIN. MACRO M1111 26-SEP-79 10:28 PAGE 7-6  
EST15 TEST MAIN MEMORY FROM VIRTUAL 001000 TO LAST ADDR.

671		
672	165526	005120
673	165530	020005
674	165532	101770
675	165534	014001
676		
677	165536	005101
678	165540	020001
679	165542	001402
680	165544	104000
681	165546	000000
682		
683	165550	020002
684	165552	001370
685		

3\$:	COM	(R0)+
	CMP	R0,R5
	BLOS	2\$
4\$:	MOV	-(R0),R1
	COM	R1
	CMP	R0,R1
	BEQ	5\$
	ERROR	
	HALT	
5\$:	CMP	R0,R2
	BNE	4\$

:R0 = ADDRESS, R1 = DATA RECEIVED, R0 = DATA EXPECTED  
:COMPLEMENT DATA AND INCREMENT ADDRESS

:READ THE DATA (IT SHOULD NOW BE THE  
:COMPLEMENT OF THE ADDRESS)  
:COMPLEMENT BEFORE CHECKING  
:IS THE DATA CORRECT?  
:BRANCH IF YES  
:GO TO PRINT ROUTINE  
:DATA ERROR ON READING MEMORY LOCATION  
:R0=ADDRESS, R1=DATA RECEIVED, R0=DATA EXPECTED

687  
688  
689  
590  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743

165554  
165554 012737 165704 000004  
165562 000402  
165564 000167 177210  
165570 005013  
165572 012737 165676 000004  
165600 012702 001000  
165604 010200  
165606 010010  
165610 005720  
165612 020005  
165614 101774  
165616 010200  
165620 005110  
165622 005110  
165624 021000  
165626 001402  
165630 104000  
165632 000000  
165634 005720  
165636 032737 000400 177750  
165644 001403  
165646 104000  
165650 000000  
165652 000406  
165654 020005  
165656 101760  
165660 000411  
165662 104000  
165664 000000  
165666 000776  
165670  
165670 012713 001015  
165674 000403

```
.....  
.SBTTL TEST16 TEST MEMORY WITH THE DATA CACHE ON  
*  
* THIS TEST CHECKS VIRTUAL MEMORY FROM 001000 THRU LAST ADDRESS  
* TO INSURE THAT YOU CAN GET HITS ALL THE WAY UP THROUGH MAIN  
* MEMORY.  
*  
* UPON ENTRY THE REGISTERS WILL BE SET UP AS FOLLOWS:  
* R0 = 001000 (ADDRESS), R2 = (FIRST ADDRESS), R3=177746(CACHE CONTROL REGISTER)  
* R5 = (LAST MEMORY ADDRESS), SP = 776  
*  
* R4 IS USED AS POINTER FOR ERROR MESSAGE  
*  
.....  
TST16:  
MOV #JUMP,@#4 ;SETUP FOR POSSIBLE TRAP  
BR +6  
JMP ENTBT ;ENTRY: BOOT/POWERUP TO BOOTSTRAP ROM  
; WITH DIAGNOSTIC  
CLR (R3) ;ENABLE ALL CACHE; ALLOW FOR CACHE INTERRUPTS  
MOV #CPUERR,@#4 ;RESTORE VECTORS  
  
MOV #1000,R2 ;SETUP FIRST ADDRESS  
MOV R2,R0 ;FIRST ADDRESS IS 1000 OCTAL  
1$: MOV R0,(R0) ;FILL MEMORY WITH ADDRESSES  
TST (R0)+  
CMP R0,R5  
BLOS 1$  
2$: MOV R2,R0 ;FIRST ADDRESS  
3$: COM (R0) ;DOUBLE COMPLEMENT DATA AND  
COM (R0) ;MAKE SURE IT IS IN THE CACHE.  
CMP (R0),R0 ;CREATE READ HIT;  
BEQ 5$ ;BRANCH IF DATA MATCHES  
ERROR ;GOTO PRINT ROUTINE  
HALT ;DATA DIDN'T MATCH R0 = ADDRESS + 2  
5$: TST (R0)+ ;READ HIT; CLOCK HIT INFO. INTO MAINTENANCE REGISTER  
BIT #400,@#177750 ;WAS THE LAST MEMORY REFERENCE A HIT  
BEQ 4$ ;BRANCH IF YES  
ERROR ;GOTO PRINT ROUTINE  
HALT ;HIT FAILED TO OCCUR R0 = ADDRESS + 2  
BR DSBLCACH ;ABORT REST OF TEST IF 'CONTINUE' PRESSED  
4$: CMP R0,R5  
BLOS 3$  
JMP0: BR JUMP ;EXIT TEST  
  
CONT: ;GOTO PRINT ROUTINE  
ERROR ;STOP HERE IF THERE IS A CACHE PARITY ERROR  
HALT ;OR A MAIN MEMORY PARITY ERROR  
;CHECK CCR, MEMORY REGISTER AND CPU REGISTER  
;TO FIND WHICH ONE  
  
DSBLCACH: BR -2  
MOV #MISS,(R3) ;FORCE MISSES IN BOTH GROUPS OF CACHE  
BR JUMP
```



```

744 65676 104000          CPUERR: ERROR          :GOTO PRINT ROUTINE
745 165700 000000          HALT              :TRAP TO ERROR VECTOR 4 OCCURRED-(CHECK STACK
746                                     BR          .-2          :FOR ORIGIN
747 165702 000776
748
749 165704          JUMP:          TST @#706          :WAS THE DIAGNOSTIC ENTERED FROM PERIPHERAL
750 165704 005737 000706
751                                     BNE 1$          :ROM
752 165710 001002          HALT              :YES
753 165712 000000          BR          .-2          :HALT TO GET TO 11/44 CONSOLE
754 165714 000776
755
756 165716          1$:          MOV          @#700,R0          :NOW RESTORE ALL NEEDED REGISTERS.
757 165716 013700 000700          MOV          @#702,R1          :AND RETURN TO BOOTSTRAP.
758 165722 013701 000702          MOV          @#704,R4
759 165726 013704 000704          JMP          2(R4)          :RETURN TO PERIPHERAL ROM.REGISTER R4
760 165732 000164 000002          :CONTAINS RETURN ADDRESS LOCATION
761
762
763 165736 012704 165762          $ERR: MOV          #ERRMSG,R4
764 165742 105737 177564          PUTCHR: TSTB          @#177564
765 165746 100375          BPL          PUTCHR
766 165750 112437 177566          MOVB          (R4)+,@#177566
767 165754 105714          TSTB          (R4)
768 165756 001371          BNE          PUTCHR
769 165760 000002          RTI
770
771 165762          015          012          105          ERRMSG: .ASCIZ <15><12>/ERROR/
772 165765          122          122          117
773 165770          122          000
774
775 165772          .EVEN
776          .=165772
777          .WORD 0
778 165774 041460          .WORD 41460          :IDENTIFIES ROM AS 'CO': C VERSION, REV 0.
779 165776 124140          .WORD 124140          :CONTAINS CRC-16 WORD FOR LAST 255 WORDS.
780
781
782          000001          .END
  
```

CONT	165662	FIX	165434	START	165010	TST14	165336	TST4	165124
CPUERR	165676	JUMP	165704	TST1	165066	TST15	165410	TST5	165136
DSBLCA	165670	JUMPO	165660	TST10	165222	TST16	165554	TST6	165174
ENTBT	165000	MISS =	001015	TST11	165242	TST2	165072	TST7	165204
ERRMSG	165762	PUTCHR	165742	TST12	165262	TST3	165110	BERR	165736
ERROR =	104000	SAVE	165022	TST13	165304				

. ABS. 166000 000  
 000000 001  
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 376 WORDS ( 2 PAGES)  
 DYNAMIC MEMORY: 2794 WORDS ( 10 PAGES)  
 ELAPSED TIME: 00:00:15  
 CKKFAAO.SEQ/NL:TOC/CRF=CKKFAAO.P11

SYMBOL	VALUE	REFERENCES
CONT	165662	7-653 #8-734
PLERR	165676	7-348 7-656 8-708 #8-744
DSBLCA	165670	8-729 #8-740
ENTBT	165000	#7-337 8-705
ERRMSG	165762	8-763 #8-771
ERROR	= 104000	#7-335 7-669 7-680 8-722 8-727 8-734 8-744
FIX	165434	#7-649
JUMP	165704	8-703 8-732 8-747 #8-749
JUMPO	165660	#8-732
MISS	= 001015	#7-332 7-652 8-741
PIJCHR	165742	#8-764 8-765 8-768
SAVE	165022	7-340 7-342 #7-345
START	165010	#7-341 7-344
TST1	165066	#7-359
TST10	165222	7-495 #7-512
TST11	165242	7-519 #7-536
TST12	165262	7-543 #7-560
TST13	165304	7-568 #7-584
TST14	165336	7-595 #7-609
TST15	165410	#7-641
TST16	165554	#8-702
TST2	165072	7-362 #7-373
TST3	165110	7-380 #7-396
TST4	165124	7-401 #7-417
TST5	165136	7-421 #7-439
TST6	165174	7-452 #7-468
TST7	165204	7-471 #7-488
SFRH	165736	7-350 #8-763