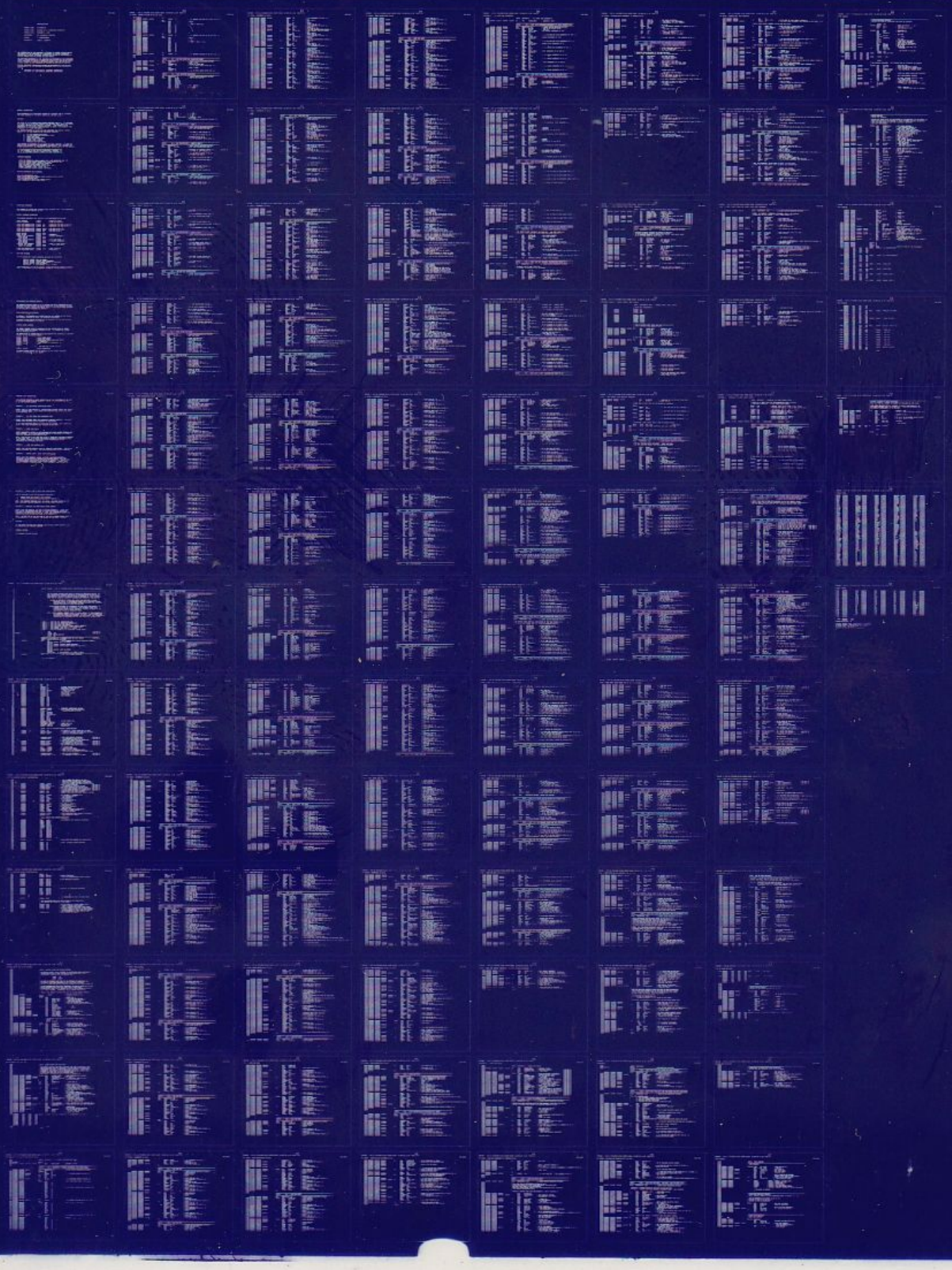
SBC11/Z1
11/Z1+

KXT-11 PROC DIAG

CNKXABO

RH-S943B-MC
FICHE 01 OF 01

APR 1984
digital
Made In USA



IDENTIFICATION

PRODUCT CODE: AC-S942B-MC

PRODUCT NAME: CNKXAB0 KXT-11 PROCESSOR DIAGNOSTIC

PRODUCT DATE: DECEMBER 5, 1983

MAINTAINER: ISS DIAGNOSTIC SERVICES

AUTHOR: G. PASQUANTONIO

REVISED BY: JAMES S. DOUCETTE

1.0 GENERAL INFORMATION

THIS DIAGNOSTIC IS A RAM-BASED VERSION OF "CIKXACO" AND IS INTENDED FOR FIELD SUPPORT OF THE KXT-11 SBC-11/21 AND SBC-11/21+.

1.1 ABSTRACT

THE KXT11 IS A T11-BASED SINGLE BOARD COMPUTER (SBC-11). IT INCLUDES A T11 CPU, UP TO 12 KW RAM, SOCKETS FOR OPTIONAL ROM, TWO 8 BIT SERIAL I/O PORTS, AND ONE 8 BIT PARALLEL I/O PORT. THIS PROGRAM IS AN ASSORTMENT OF 147 (OCTAL) TESTS DESIGNED TO VERIFY THE INTEGRITY AND OPERABILITY OF THOSE COMPONENTS.

THE PROGRAM RESIDES IN Q-BUS RAM AND EXERCISES THE VARIOUS LOGICAL SEGMENTS OF THE SBC BOARD IN THE FOLLOWING SEQUENCE:

1. T11 CPU INSTRUCTION TESTS.
2. T11 CPU TRAPS AND INTERRUPTS TESTS.
3. LOCAL RAM TESTS.
4. LOCAL ROM ADDRESS TEST.
5. SERIAL LINE 1 TESTS.
6. SERIAL LINE 2 TESTS.
7. PARALLEL I/O PORT TESTS.

THIS PROGRAM IS DESIGNED TO RUN UNDER THE XXDP+ MONITOR. IT DOES NOT REQUIRE THE SERVICES OF THE DIAGNOSTIC SUPERVISOR. ALL TERMINAL I/O (ERROR REPORTS ETC.) ARE HANDLED BY DIRECT CALLS (EMT'S) TO THE MONITOR.

```
**********************************************************
*** THIS PROGRAM IS NOT ACT/APT COMPATABLE, HOWEVER  ***
*** IT IS CHAINABLE DIRECTLY UNDER THE XXDP+ MONITOR ***
**********************************************************
```

1.2 HARDWARE REQUIRED

KXT11-AA (M8063-AA) MODULE UNDER TEST, WITH MACRO-ODT ROM -OR-
KXT11-AB (M7676) MODULE UNDER TEST, WITH MACRO-ODT ROM.
KXT11-LP (M3275) DUAL LOOP-BACK CONNECTOR.
16KW (OR MORE) Q-BUS MEMORY (MSV11-D).
XXDP+ LOAD DEVICE AND MEDIA (RX02, RL02, ETC.).
CONSOLE TERMINAL (VT52, VT100, ETC.).
Q-BUS EXERCISER (G5201) MODULE (OPTIONAL).

1.3 RELATED DOCUMENTS AND STANDARDS

KXT11-AA ENGINEERING SPEC.
M8063 FALCON SBC-11/21 SINGLE BOARD COMPUTER USER'S GUIDE.
KXT11-AB ENGINEERING SPEC.
M7676 FALCON-PLUS SBC-11/21+ USER'S GUIDE.
XXDP+ DIAGNOSTIC SYSTEM USERS GUIDE.

2.0 OPERATING PROCEDURE

THE PROGRAM IS LOADED AND STARTED USING STANDARD XXDP+ LOAD PROCEDURES.
THE START/RESTART ADDRESS IS 200.

2.1 DEFAULT PROGRAM PARAMETERS

THE PROGRAM REQUIRES THAT THE KXT-11 BE CONFIGURED AS FOLLOWS:

START ADDRESS	172000		(POWER-UP START)
RESTART ADDRESS	172004		(TIME-OUT RESTART)
LOCAL ROM ADDRESS/SIZE	170000	1KW	(MACRO-ODT FOR SBC-11/21) OR
LOCAL ROM ADDRESS/SIZE	164000	2KW	(MACRO-ODT FOR SBC-11/21+)
LOCAL RAM ADDRESS/SIZE	160010	2KW-4	(FOR SBC-11/21) OR
LOCAL RAM ADDRESS/SIZE	100000	12KW	(FOR SBC-11/21+) OR
LOCAL RAM ADDRESS/SIZE	140000	4KW	(FOR SBC-11/21+)
BEVNT VECTOR	000100	PRI6	
BHALT VECTOR	000140	PRI7	(MASKABLE)
SERIAL LINE 1 ADDRESS	177560		(CONSOLE TERMINAL)
RCVR VECTOR	000060	PRI4	
XMTR VECTOR	000064	PRI4	
SERIAL LINE 2 ADDRESS	176540		(WITH LOOP-BACK...)
RCVR VECTOR	000120	PRI5	(...CLOCK DISABLED)
XMTR VECTOR	000124	PRI5	
PARALLEL PORT ADDRESS	176200		(MODE 1, A IN, B OUT...)
OUTPUT (B) VECTOR	000130	PRI5	(...WITH LOOP-BACK)
INPUT (A) VECTOR	000134	PRI5	
Q-BUS RAM ADDRESS/SIZE	000000	16KW	(XXDP+ MINIMUM)
Q-BUS EXERCISER CSR1	174020		(OPTIONAL)

2.2 RUN TIME OPTIONS

THE FOLLOWING SOFT SWITCH-REGISTER OPTIONS ARE SUPPORTED:

SWR<15>	100000	HALT ON ERROR.
SWR<13>	020000	INHIBIT ERROR MESSAGES.
SWR<12>	010000	PRINT ERROR SUMMARY AT END-PASS.
SWR<09>	001000	LOOP ON ERROR.
SWR<02:00>	N	RUN SEGMENT N (1 - 7) ONLY.

THE PROGRAM WILL ASK FOR AN INITIAL SWITCH SETTING AT START OR RESTART TIME. THEREAFTER, YOU MAY REQUEST A CHANGE BY TYPING CONTROL G <↑G>.

3.0 PERFORMANCE AND PROGRESS REPORTS

THE PROGRAM REQUIRES FROM 10 TO 20 SECONDS PER PASS (DEPENDING ON THE HARDWARE) AND WILL REPORT THE END OF EACH PASS ON THE CONSOLE TERMINAL, INCLUDING A TOTAL (CUMULATIVE) ERROR COUNT. AN OPTIONAL ERROR SUMMARY MAY BE INCLUDED IF ENABLED VIA SWR<12>.

4.0 ERROR REPORTING AND RECOVERY

BY DEFAULT, THE PROGRAM WILL REPORT AND LOG ALL ERRORS AS THEY OCCUR AND PROCEED (FOREVER OR UNTIL THE OPERATOR INTERVENES).

ALTERNATE ACTION AND/OR RECOVERY MAY BE SELECTED VIA THE SOFT SWITCH REGISTER AS DESCRIBED IN 2.2 ABOVE.

4.1 DYNAMIC ERROR LOGGING

THE ERROR LOGGING FACILITY PROVIDES FOR THE ACCUMULATION OF ERROR STATISTICS OVER AN EXTENDED PERIOD OF TIME. ALL ERRORS ARE LOGGED, WHETHER REPORTED OR NOT.

THE ERROR DATA IS ACCUMULATED IN Q-BUS SCRATCH RAM STARTING AT LOCATION 004000, IN THE FOLLOWING FORMAT:

```
004000  XXXXXX          ; TOTAL ERROR COUNT.
004002  YYYYYY          ; TOTAL PASS COUNT.
004004  ZZZZZZ          ; CONSECUTIVE ERROR-FREE PASS COUNT.
004006  PCPCPC          ; 1ST ERROR PC...
004010       N          ;...AND NUMBER OF OCCURANCES.
004012  PCPCPC          ; 2ND...
004014       N
  ::                    ;...AND ROOM FOR A TOTAL OF 510.
  ::
007776  177777          ; TABLE TERMINATOR.
```

AN ERROR SUMMARY REPORT MAY BE ENABLED VIA THE SWITCH REGISTER SWR<12> AS DESCRIBED IN 2.2 ABOVE.

5.0 HARDWARE TEST DESCRIPTION

THIS SECTION PROVIDES A BRIEF DESCRIPTION OF THE FUNCTIONALITY OF EACH OF THE SEVEN MAJOR PROGRAM SEGMENTS. REFER TO THE PROGRAM LISTING (6.0) FOR FURTHER DETAILS.

5.1 SEGMENT 1 -- T11 CPU BASIC INSTRUCTION TESTS

VERIFY THAT T11 CAN EXECUTE ALL INSTRUCTIONS (EXCEPT "WAIT" AND "HALT") IN ALL ADDRESS MODES. ALSO CHECKS FOR EXPECTED SIDE EFFECTS (CC BITS, AUTO-INCR/DECR, ETC...).

5.2 SEGMENT 2 -- T11 CPU TRAPS AND INTERRUPTS TEST

VERIFY THAT SOFTWARE TRAPS AND INTERRUPTS EXECUTE CORRECTLY, AND THAT PROPER STACK POINTER AND PSW CONTEXT IS MAINTAINED.

IF A Q-BUS EXERCISOR MODULE IS INSTALLED (AT 174020), THEN POWER-FAIL, BEVNT, AND Q-BUS INTERRUPTS WILL ALSO BE TESTED HERE.

5.3 SEGMENT 3 -- LOCAL RAM TESTS

VERIFY READ/WRITE ACCESS TO THE ON-BOARD RAM USING VARIOUS ADDRESS AND DATA PATTERNS. IF A Q-BUS-EXERCISER MODULE IS INSTALLED (AT 174020), THEN TEST READ/WRITE DMA ACCESS AS WELL.

NOTE: FOR THE SBC-11/21 (NOT SBC-11/21+), MACRO-ODT UTILIZES A PORTION OF THE LOCAL RAM. IF A BREAK-AND-PROCEED SEQUENCE IS SERVICED DURING THIS SEGMENT, ERRORS MAY OCCUR DUE TO THE OVERWRITING OF RAM BACKGROUND DATA PATTERNS (BY ODT).

5.4 SEGMENT 4 -- LOCAL ROM ADDRESS TEST

VERIFY THAT THE ROM ADDRESS SPACE IS CORRECTLY CONFIGURED. THIS IS AN ADDRESS RESPONSE TEST ONLY. ROM DATA VERIFICATION IS NOT ATTEMPTED.

5.5 SECTION 5 -- SERIAL LINE 1 TESTS (DEC DC319 DLART)

VERIFY THAT THE "CONSOLE" SERIAL I/O PORT FUNCTIONS CORRECTLY, INCLUDING THE DETECTION OF FRAMING AND OVERRUN ERRORS, AND THAT INTERRUPTS OCCUR THRU VECTORS 60 AND 64 AT PRIORITY 4. VERIFY THAT THE PROGRAMMABLE BAUD-RATE FEATURE WORKS CORRECTLY. VERIFY THAT "BREAK" INVOKES A MASKABLE LEVEL 7 HALT INTERRUPT THRU VECTOR 140.

5.6 SECTION 6 -- SERIAL LINE 2 TESTS (DEC DC319 DLART)

SAME AS SECTION 5 WITH THE FOLLOWING EXCEPTIONS:

1. "BREAK" DOES NOT INVOKE A HALT REQUEST.
2. INTERRUPTS VECTOR THRU 120 AND 124 AT PRIORITY 5.

NOTE: THIS SEGMENT REQUIRES THAT THE SERIAL LOOP-BACK SWITCH ON THE DUAL LOOP-BACK CARD BE CORRECTLY SET TO THE "LOOP" POSITION.

5.7 SECTION 7 -- PARALLEL I/O PORT TESTS (INTEL 8255A)

VERIFY THAT THE PARALLEL I/O PORT FUNCTIONS CORRECTLY. CHECK THAT DATA CAN BE TRANSMITTED FROM PORT B. THRU THE PARALLEL LOOP-BACK CONNECTOR, AND RECEIVED IN PORT A. CHECK THAT INTERRUPTS OCCUR THRU VECTORS 130 (PORT B) AND 134 (PORT A) AT PRIORITY 5.

NOTE: THE RED LED ON THE TOP EDGE OF THE KXT-11 BOARD SHOULD LIGHT AT THE BEGINNING OF THIS SEGMENT, AND REMAIN ON FOR ABOUT 1 SECOND.

5.8 END PASS

AT THIS POINT AND END-PASS MESSAGE (AND OPTIONAL ERROR SUMMARY) WILL BE PRINTED ON THE CONSOLE TERMINAL.

6.0 PROGRAM LISTING

THE PROGRAM LISTING FOLLOWS.

```
                        .TITLE  CNKXAB -- KXT-11 (FALCON(-PLUS)) VERIFICATION TESTS (FIELD VERSION).
                        ;
                        ;-------------------------------------------------------------------------------
                        ;                                                                              !
                        ;       THE FOLLOWING CHANGES WERE MADE TO CNKXAA IN PRODUCING CNKXAB. THE      !
                        ;       MOST SIGNIFICANT CHANGES WERE MADE TO ACCOMODATE THE FALCON-PLUS        !
                        ;       ARCHITECTURE. REVISION B RELEASE DATE IS DECEMBER 5, 1983. CHANGES      !
                        ;       WERE MADE BY JAMES S. DOUCETTE (CHANGES ARE MARKED BY ;JSD REV B).      !
                        ;                                                                              !
                        ;       1. CHANGES TO SECTION 3 (LOCAL RAM TESTS) AND SECTION 4 (LOCAL         !
                        ;               PROM/RAM TESTS) TO ACCOMODATE MEMORY MAP 0 FOR BOTH FALCON     !
                        ;               AND FALCON-PLUS. ALSO RENAMED LITERALS DEFINING THE STARTING   !
                        ;               ADDRESS AND SIZE OF LOCAL RAM AND ROM.                         !
                        ;                                                                              !
                        ;       2. MINOR CODING CHANGES TO ACCOMODATE VAX/VMS HOST DEVELOPMENT, AS      !
                        ;               OPPOSED TO TOPS-20. (EXAMPLES: .ABS BECOMES .ENABL ABS,        !
                        ;               INSTRUCTIONS USING @<N_N> (WHERE N = NUMBER) CHANGED TO THE     !
                        ;               EQUIVALENT VALUE IN IMMEDIATE MODE).                           !
                        ;                                                                              !
                        ;       NOTE: THREE ASSEMBLY ERRORS (TYPE "Z") WILL RESULT IF THIS DIAGNOSTIC  !
                        ;               IS ASSEMBLED UNDER VAX/VMS USING "MCR MAC". THESE ERRORS ARE    !
                        ;               NON-FATAL AND SHOULD BE IGNORED. IN SOME CASES, THE "ERRONEOUS" !
                        ;               INSTRUCTION IS REQUIRED FOR THE TEST WHICH IS TAKING PLACE.     !
                        ;                                                                              !
                        ;-------------------------------------------------------------------------------
                        ;
                        ; PART 1 -- TEST THE T11 INSTRUCTION SET.
                        ; PART 2 -- TEST THE T11 TRAPS AND INTERRUPTS.
                        ; PART 3 -- TEST THE LOCAL RAM MEMORY.
                        ; PART 4 -- TEST THE LOCAL ROM/RAM MEMORY (EMPTY SOCKETS).
                        ; PART 5 -- TEST SERIAL LINE UNIT 1 (DEC DC319 DLART).
                        ; PART 6 -- TEST SERIAL LINE UNIT 2 (    DITTO    ).
                        ; PART 7 -- TEST THE PARALLEL I/O INTERFACE (INTEL 8255).
                        ;
                        ;       .ABS                                                    ;JSD REV B
000000                  ;       .ENABL  ABS                                             ;JSD REV B
                        ;       .NLIST  MD,MC,CND,TOC
                        ;       .LIST   ME
                        ;*******************************************************************************
                        ;PRGSIZ=<^H<@LSTAD>>+1  ; PROGRAM SIZE IN 1/8 K UNITS.          ;JSD REV B
                        .SBTTL  SWITCH REGISTER OPTIONS
000176                  SWR=    176             ; THE STANDARD SOFT SWITCH REGISTER ADDRESS.
                        ;       SWR<15> 100000  HALT ON ERROR.
                        ;       SWR<14>
                        ;       SWR<13> 020000  INHIBIT ERROR MESSAGES.
                        ;       SWR<12> 010000  PRINT ERROR SUMMARY AT END-PASS.
                        ;       SWR<11>
                        ;       SWR<10>
                        ;       SWR<09> 001000  LOOP ON ERROR.
                        ;       SWR<08>
                        ;       SWR<02:00>   N  RUN SEGMENT N (1 THRU 7) ONLY.
                        ;*******************************************************************************
000000                  TN@=    0               ; INIT TEST NUMBER SEQUENCE.
                        ;
                        ;
                        ; T11 CPU EQUATES.
                        ;
```

```
SWITCH REGISTER OPTIONS

     83          000006                  R6=     SP
     84          000007                  R7=     PC
     85          000004                  ERRVEC= 4               ; ILLEGAL INSTRUCTION VECTOR.
     86          000010                  RESVEC= 10              ; RESERVED OPCODE VECTOR.
     87          000014                  BPTVEC= 14              ; BREAKPOINT VECTOR...
     88          000014                  TRTVEC= BPTVEC          ;...TRACE TRAP TOO.
     89          000020                  IOTVEC= 20              ; IOT VECTOR.
     90          000024                  PWRVEC= 24              ; POWER FAIL VECTOR.
     91          000030                  EMTVEC= 30              ; EMT VECTOR.
     92          000034                  TRPVEC= 34              ; TRAP VECTOR.
     93          000340                  PRI7=   340
     94          000300                  PRI6=   300
     95          000240                  PRI5=   240
     96          000200                  PRI4=   200
     97          000140                  PRI3=   140
     98          000100                  PRI2=   100
     99          000040                  PRI1=   040
    100          000000                  PRI0=   000
    101
    102          000010                  RESRVD= 000010          ; RESERVED (UNDEFINED) OPCODE.
    103          000007                  MFPT=   000007          ; MFPT OPCODE (NEW INSTRUCTION)...
    104          000004                  KXT11=  4               ;...SHOULD RETURN 04 IN R0.
    105          000254                  CLNZ=   CLN!CLZ
    106          000263                  SEVC=   SEV!SEC
    107          000273                  SENVC=  SEN!SEV!SEC
    108          000260                  NOP1=   260
    109          000401                  SKP1=   BR+1
    110          000402                  SKP2=   BR+2
    111          000403                  SKP3=   BR+3
    112          104400                  CHECKCC= TRAP           ; CC BIT CHECK CALL.
    113          104420                  BEGINTEST= TRAP+20
    114          104421                  CHECKSWR= TRAP+21       ; CHECK FOR <^G>.
    115          104422                  GETSWR= TRAP+22         ; GET NEW SWR.
    116          104423                  ERROR=  TRAP+23         ; ERROR CALL.
    117                                  ;
    118                                  .SBTTL  KXT-11 (FALCON) DEFAULT CONFIGURATION
    119                                  ;
    120          172000                  START=  ODT             ; ON POWER-UP, START MACRO-ODT IN PROM...
    121          172004                  RESTART= ODTRE          ;...TRAP TO "RESTART" ON TIME-OUT OR HALT...
    122          000000                  EHALT=  HALT            ;...AND HALT ON PROGRAM ERROR.
    123
    124
    125                                  ;ROMADR=170000          ; LOCAL ROM SOCKETS STRAPPED...         ;JSD REV B
    126                                  ;ROMSIZ=1024.           ;...TO RESPOND AT 30K TO 31K...         ;JSD REV B
    127                                                          ;...MACRO-ODT PROM MUST BE INSTALLED.
    128          170000                  ROMAD1=170000           ; LOCAL ROM SOCKETS STRAPPED...         ;JSD REV B
    129          002000                  ROMSZ1=1024.            ;...TO RESPOND AT 30K TO 31K...         ;JSD REV B
    130          164000                  ROMAD2=164000           ; ROM ADDRESS FOR FALCON-PLUS           ;JSD REV B
    131          004000                  ROMSZ2=2048.            ; 2KW OF ROM SPACE FOR FALCON-PLUS      ;JSD REV B
    132
    133          172000                  ODT=    172000          ; IT STARTS HERE...
    134          172004                  ODTRE=  172004          ;...AND RESTARTS HERE.
    135          170000                  ODTBRK= 170000          ; "BREAK" HALT IS SERVICED HERE...
    136          000140                  ODTIN=  BHALT           ;...AND THE VECTOR SERVES AS A HANDY FLAG.
    137
    138                                  ;RAMADR=160010          ; LOCAL 2K RAM STRAPPED...              ;JSD REV B
    139                                  ;RAMSIZ=2048.-4.-8.     ;...TO RESPOND AT 28K(+4) TO 30K...     ;JSD REV B
```

```
                                        ;...1ST 4 WORDS ARE NON-EX SO THAT...
                                        ;...CONVENTIONAL SIZERS WILL WORK RIGHT...
                                        ;...AND LAST 8 WORDS BELONG TO MACRO-ODT...
167776                  TRAP4= 167776   ;...(THE LAST BEING THE TRAP4 EMULATE FLAG).
160010                  RAMAD1=160010   ; LOCAL 2K RAM STRAPPED...              ;JSD REV B
003764                  RAMSZ1=2048.-4.-8.  ;...TO RESPOND AT 28K(+4) TO 30K...  ;JSD REV B
100000                  RAMAD2=100000   ; RAM START ADDRESS FOR FALCON-PLUS      ;JSD REV B
024000                  RAMSZ2=10240.   ; MAX RAM SPACE (WORDS) FOR FALCON-PLUS ;JSD REV B
                                        ;...(AVOID MONITOR CODE AT 150000-OCT)  ;JSD REV B
000100                  BEVNT= 100      ; "LINE CLOCK" (ON SLU2) VECTOR.
000140                  BHALT= 140      ; "BREAK" (ON SLU1) OR Q-BUS "BHALT" VECTOR.

177560                  SLU1=  177560   ; SLU (DLART) 1 (CONSOLE TTY)...
000060                  SLU1V= 60       ;...1ST VECTOR (OF 2)...
000200                  SLU1P= PRI4     ;...AND PRIORITY.
176540                  SLU2=  176540   ; SLU (DLART) 2...
000120                  SLU2V= 120      ;...1ST VECTOR (OF 2)...
000240                  SLU2P= PRI5     ;...AND PRIORITY.

176200                  PIOA=  176200   ; PIO PORT A (INPUT)...
000134                  PIOAV= 134      ;...AND VECTOR.
176202                  PIOB=  176202   ; PIO PORT B (OUTPUT)...
000130                  PIOBV= 130      ;...AND VECTOR.
000240                  PIOP=  PRI5     ; INTERRUPT PRIORITY (FOR BOTH).
176204                  PIOC=  176204   ; PIO PORT C (PORT CONTROL)...
176206                  PIOMC= 176206   ;...AND MODE CONTROL.

174020                  QBXCSR= 174020  ; Q-BUS EXERCISER CSR1 ADDRESS (OPTIONAL)...
000110                  QBXVEC= 110     ;...AND VECTOR (AT PRI4).
                        ;
                        ; NOW, EQUATE SOME GENERAL REGISTERS.
                        ;
001000                  ADR=   1000     ; INSTRUCTION TEST SCRATCH LOCATIONS.
001002                  ADR1=  ADR+2
001004                  ADR2=  ADR+4
001006                  TEMP=  ADR+6
001010                  TEMP1= ADR+10
001012                  TEMP2= ADR+12
001014                  TEMP3= ADR+14
001016                  K0=    ADR+16
001020                  K1=    ADR+20
001022                  K2=    ADR+22
001024                  K3=    ADR+24
001026                  K4=    ADR+26
001030                  K5=    ADR+30
001032                  K6=    ADR+32
001034                  K7=    ADR+34

001036                  QBXA=  ADR+36   ; Q-BUS EXER CSR ADDRESS (IF INSTALLED).

001016                  SPD0=  K0       ; DLART, VARIABLE SPEED COUNTERS.
001020                  SPD1=  K1
001022                  SPD2=  K2
001024                  SPD3=  K3
001026                  SPD4=  K4
001030                  SPD5=  K5
001032                  SPD6=  K6
```

K1

KXT-11 (FALCON) DEFAULT CONFIGURATION

```
    197          001C34                  SPD7=   K7
    198          001040                  DFPBR=  ADR+40          ; DEFAULT PBR BITS...
    199          001042                  DFSPD=  ADR+42          ;...AND THE CORRESPONDING SPEED COUNTER.
    200
    201          001044                  RCSR=   ADR+44          ; DLART REGISTER POINTERS.
    202          001046                  RBUF=   ADR+46
    203          001050                  XCSR=   ADR+50
    204          001052                  XBUF=   ADR+52
    205          001054                  RVEC=   ADR+54
    206          001056                  XVEC=   ADR+56
    207
    208          001060                  PPA=    ADR+60          ; PARALLEL I/O REGISTER POINTERS.
    209          001062                  PPB=    ADR+62
    210          001064                  PPC=    ADR+64
    211          001066                  PPCW=   ADR+66
    212          001070                  PPAV=   ADR+70
    213          001072                  PPBV=   ADR+72
    214                                                  ;  \\
    215                                                  ;   >> REMAINDER OF 1ST K IS OUR STACK.
    216          003776                  STACK=  3776    ;  //
    217                                  ;
    218                                  ; NOW THE ENTIRE 2ND K OF RAM IS USED AS AN "ERROR LOG".
    219                                  ; ALL ERRORS ARE LOGGED AS THEY OCCUR.
    220                                  ;
    221          004000                  ERRCNT= 4000            ; THE 1ST WORD IS A TOTAL ERROR COUNT...
    222          004002                  TPASS=  4002            ;...THEN A "TOTAL" PASS COUNT...
    223          004004                  GPASS=  4004            ;...AND A "CONSECUTIVE-GOOD" PASS COUNT.
    224          004006                  EPCTBL= 4006            ; THE REST FORMS A TABLE OF 2 WORD ENTRIES...
    225                                                          ;...SHOWING ERROR PC (WD1) AND THE NUMBER...
    226                                                          ;...OF OCCURRANCES OF THAT ERROR (WD2).
    227          007776                  LOGEND= 7776            ; END OF ERROR LOG TABLE.
```

```
                                .SBTTL  INITIAL START AND CLEAR VECTORS.
                                ;
                                ; ON INITIAL START, WE'LL ATTEMPT TO INITIALIZE THE Q-BUS MEMORY
                                ; VECTOR SPACE WITH A TRAP CATCHER IN THE TRADITIONAL STYLE.
                                ; LOCATIONS 0 THRU 776 WILL BE COATED WITH:
                                ;
                                ;               .WORD   .+2
                                ;               .WORD   EHALT
                                ;
                                ; WE HAVE TO PRESUME THAT THE T11 IS CAPABLE OF EXECUTING THE
                                ; FEW BASIC FUNCTIONS REQUIRED TO SET THE TRAP CATCHERS.
                                ; IF NOT -- THEN HOW DID WE EVER GET STARTED IN THE FIRST PLACE ???
                                ;
                                ; IN THE ENGINEERING AND FIELD VERSIONS, WE HAVE A MACRO-ODT IN ROM
                                ; WHICH PROVIDES "BUS TIME-OUT" AND "HALT" EMULATION.
                                ; ON A RESTART TRAP, WE'LL REMAIN IN ODT IF THE STACKED PC POINTS
                                ; TO A "HALT" OPCODE.  OTHERWISE, WE'LL VECTOR THRU LOC 4.
                                ;
                                ; "BHALT" AND "BREAK" VECTOR THRU 140 TO THE ODT BREAK HANDLER.
                                ;
  010000                                .= 10000                ; PROGRAM STARTS AT 2K.
010000                          BEGIN:
010000  000240                  INIT:   NOP                     ; RESET ???
010002  013746  000030                  MOV     @#30,-(SP)      ; SAVE XXDP EMT VECTOR...
010006  013746  000040                  MOV     @#40,-(SP)      ;...LOAD DEV ID AND UNIT...
010012  013746  000042                  MOV     @#42,-(SP)      ;...AND CHAIN RETURN.
010016  012700  000002                  MOV     #2,R0           ; COAT THE ENTIRE VECTOR AREA...
010022  005001                          CLR     R1              ;...WITH A TRADITIONAL TRAP CATCHER.
010024  012702  000200                  MOV     #200,R2
010030  010021                  1$:     MOV     R0,(R1)+        ;       .+2
010032  012721  000000                  MOV     #EHALT,(R1)+    ;       EHALT
010036  022020                          CMP     (R0)+,(R0)+
010040  077205                          SOB     R2,1$
010042  012637  000042                  MOV     (SP)+,@#42      ; RESTORE CHAIN RETURN...
010046  012637  000040                  MOV     (SP)+,@#40      ;...LOAD DEV...
010052  012637  000030                  MOV     (SP)+,@#30      ;...AND EMT VECTOR.
010056  012737  000137  000200          MOV     #137,@#200      ; SET A STANDARD "RESTART" ADDRESS.
010064  012737  010072  000202          MOV     #INIT1,@#202
                                ;
                                ; IF WE GOT THIS FAR WE'RE IN PRETTY GOOD SHAPE !.!.!
                                .SBTTL
                                .SBTTL  RESTART
                                ;
010072  012700  001000          INIT1:  MOV     #1000,R0        ; "200G" RESTARTS HERE.
010076  005020                  1$:     CLR     (R0)+           ; CLEAR ALL THE REST OF RAM UP TO...
010100  020027  010000                  CMP     R0,#BEGIN       ;...THE PROGRAM START POINT.
010104  103774                          BLO     1$
010106  012706  003776                  MOV     #STACK,SP       ; SET A STACK POINTER.
010112  012737  170000  000140          MOV     #ODTBRK,@#BHALT ; ENABLE "BREAKS" TO ODT...
010120  012737  000340  000142          MOV     #PRI7,@#BHALT+2 ;...THRU VECTOR 140.
010126  012737  000001  167776          MOV     #1,@#TRAP4      ; ENABLE ODT'S "TRAP 4" EMULATOR.
                                ;       .IIF P2, .PRINT .-2     ; *** DEBUG, 100000 ***         ;JSD REV B
010134  106427  000000                  MTPS    #PRIO           ; AND LOWER THE CPU LEVEL.
```

```
                                        .SBTTL  QUICK-VERIFY EMT AND TRAP.
                                        ;
                                        ; NOW TRAPS (104400) AND EMTS (104000) ARE USED EXTENSIVELY FOR
                                        ; ERROR HANDLING, AND TTY SERVICE.  THE EMT HAS TO BE FUNCTIONAL
                                        ; OTHERWISE, THE XXDP+ MONITOR WOULD HAVE CRASHED.  TRAPS ARE
                                        ; UNKNOWN AT THIS POINT.  SO TO BE SAFE, WE'LL DO A QUICK VERIFY
                                        ; OF BOTH THE TRAP AND THE EMT OPCODES HERE.
                                        ; ADDITIONAL TESTING WILL BE ATTEMPTED IN SEGMENT 2.
                                        ;
010140  016746  167664        EMTTRP: MOV     EMTVEC,-(SP)    ; SAVE MONITORS EMT VECTOR.
010144  012767  010204  167656        MOV     #4$,EMTVEC      ; SET EMT...
010152  012767  000036  167654        MOV     #TRPVEC+2,TRPVEC ;...AND RESET TRAP VECTORS.
010160  012700  104000                MOV     #EMT+0,R0       ; TEST EMT'S FIRST...
010164  000402                        BR      2$
010166  012700  104400        1$:     MOV     #TRAP+0,R0      ;...THEN TRAP'S.
010172  010067  000000        2$:     MOV     R0,3$
010176  104000                3$:     EMT+0                   ; EXECUTE EMT/TRAP.
010200  000240                        NOP                     ; DIDN'T WORK -- BIG TROUBLE...
010202  000000                        HALT                    ;...(R0) HOLDS THE OFFENDER !!!
010204  012716  010216        4$:     MOV     #5$,(SP)        ; EMT/TRAP SEEMS OK, TWEAK RETURN PC...
010210  000002                        RTI                     ;...AND TEST THE "RTI" TOO.
010212  000240                        NOP                     ; W-T-F ???
010214  000000                        HALT                    ; RTI DOESN'T WORK !!!
010216  105200        5$:     INCB    R0                      ; WE'RE HERE IF EVERYTHING WORKED...
010220  001364                        BNE     2$              ;...BUMP THE OPCODE AND LOOP 'TIL DONE.
010222  020027  104400                CMP     R0,#TRAP        ; OPCODE = TRAP ??
010226  001407                        BEQ     6$              ; WE'RE DONE IF SO.
010230  012767  000032  167572        MOV     #EMTVEC+2,EMTVEC ; OTHERWISE, RESET EMT...
010236  012767  010204  167570        MOV     #4$,TRPVEC      ;...AND SET TRAP VECTORS...
010244  000750                        BR      1$              ;...AND GO 'ROUND ONCE.

010246  012667  167556        6$:     MOV     (SP)+,EMTVEC    ; DONE, RESTORE EMT FOR MONITOR CALLS...
010252  012767  035464  167554        MOV     #TRAPHAN,TRPVEC ;...AND SET-UP OUR OWN TRAP HANDLER.
010260  012700  010300                MOV     #HELLO,R0
010264  104003                        EMT+3                   ; "GREETINGS".
010266  104422                        GETSWR                  ; GET INITIAL SWITCH SETTING...
010270  016700  025516                MOV     INSEG,R0        ; SINGLE SEGMENT SELECTED ??
010274  001420                        BEQ     DOALL           ;...IF NOT, RUN 'EM ALL...
010276  010007                        MOV     R0,PC           ;...ELSE, GO THERE.

010300     015     012     116  HELLO:  .ASCIZ  <15><12>'NKXAB0 KXT-11 DIAGNOSTIC'<15><12>
010303     113     130     101
010306     102     060     040
010311     113     130     124
010314     055     061     061
010317     040     104     111
010322     101     107     116
010325     117     123     124
010330     111     103     015
010333     012     000
                                        .EVEN
                                .SBTTL
```

```
                                        .SBTTL  SECTION 1 -- T11 BASIC INSTRUCTION TESTS.

                                        DOALL:
010336                                  S1:     MOV     #SEG1,SEGN      ; SEGMENT 1 TEXT.
                                        ;*****************************************************************************
                                        .SBTTL      T1 -- BRANCHES, FORWARD, BACKWARD, AND CONDITIONAL
                                        ;*****************************************************************************
010344  104420  000001                  T1:     BEGINTEST, 1            ;; CHECK SWITCH REGISTER OPTIONS.
010350  000405                          BRANCH: BR      2$
010352  000405                          1$:     BR      3$
010354  104423  010344                          ERROR,  T1              ;; *** FORWARD BRANCH FAILED ***
010360  012707  010374                  4$:     MOV     #5$,PC
010364  000772                          2$:     BR      1$
010366  000774                          3$:     BR      4$
010370  104423  010344                          ERROR,  T1              ;; *** BACKWARD BRANCH FAILED ***
010374                                  5$:
010374  000257                          NOBIT:  CCC                     ; ZERO CONDITION CODES, NZVC=0000
010376  103414                                  BCS     1$
010400  102413                                  BVS     1$
010402  001412                                  BEQ     1$
010404  100411                                  BMI     1$
010406  000260                                  NOP1                    ; CHECK NOP1 INSTRUCTION (OPCODE 260).
010410  103407                                  BCS     1$
010412  102406                                  BVS     1$
010414  001405                                  BEQ     1$
010416  100404                                  BMI     1$
010420  002403                                  BLT     1$
010422  003402                                  BLE     1$
010424  101401                                  BLOS    1$
010426  101002                                  BHI     2$
010430                                  1$:
010430  104423  010344                          ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 0000 ***
010434                                  2$:
010434  000270                          NBIT:   SEN                     ; NBIT IS SET, NZVC=1000
010436  100007                                  BPL     1$
010440  001406                                  BEQ     1$
010442  002005                                  BGE     1$
010444  003004                                  BGT     1$
010446  103403                                  BCS     1$
010450  101402                                  BLOS    1$
010452  103401                                  BLO     1$
010454  003402                                  BLE     2$
010456                                  1$:
010456  104423  010344                          ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1000 ***
010462                                  2$:
010462  000270                          VBIT:   SEN                     ; SET N...
010464  000262                                  SEV                     ;...AND V, NZVC = 1010
010466  102010                                  BVC     1$
010470  001407                                  BEQ     1$
010472  100006                                  BPL     1$
010474  103405                                  BCS     1$
010476  002404                                  BLT     1$
010500  003403                                  BLE     1$
010502  101402                                  BLOS    1$
010504  103401                                  BLO     1$
010506  003002                                  BGT     2$
```

PC 10456 = BRANCH FAILURE OR CC NOT 1000.

```
389 010510                                  1$:
    010510  104423  010344                           ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1010 ***
390 010514                                  2$:
391 010514  000270                          CBIT:   SEN                     ; SET N...
392 010516  000262                                  SEV                     ;...V...
393 010520  000261                                  SEC                     ;...AND C, NZVC=1011
394 010522  001406                                  BEQ     1$
395 010524  100005                                  BPL     1$
396 010526  102004                                  BVC     1$
397 010530  002403                                  BLT     1$
398 010532  003402                                  BLE     1$
399 010534  101001                                  BHI     1$
400 010536  002002                                  BGE     2$
401 010540                                  1$:
    010540  104423  010344                           ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1011 ***
402 010544                                  2$:
403 010544  000270                          ZBIT:   SEN                     ; SET N...
404 010546  000262                                  SEV                     ;...V...
405 010550  000261                                  SEC                     ;...C...
406 010552  000264                                  SEZ                     ;...AND Z, NZVC = 1111.
407 010554  001007                                  BNE     1$
408 010556  100006                                  BPL     1$
409 010560  102005                                  BVC     1$
410 010562  103004                                  BCC     1$
411 010564  002403                                  BLT     1$
412 010566  003002                                  BGT     1$
413 010570  101001                                  BHI     1$
414 010572  001402                                  BEQ     2$
415 010574                                  1$:
    010574  104423  010344                           ERROR,  T1              ;; *** BRANCH FAILURE OR CC NOT 1111 ***
416 010600                                  2$:
417
418                                         ;*****************************************************************************
                                            .SBTTL      T2 -- SET AND CLEAR ALL CONDITION CODES
                                            ;*****************************************************************************
    010600  104420  000002                 T2:     BEGINTEST, 2            ;; CHECK SWITCH REGISTER OPTIONS.
419 010604  000277                          ALLCC:  SCC                     ; NZVC = 1111
420 010606  104417                                  CHECKCC+17              ; CHECK ALL BITS SET.
421 010610  104423  010600                          ERROR,  T2              ;; *** CC BITS WRONG AFTER SCC ***
422
423 010614  000257                                  CCC                     ; NZVC = 0000
424 010616  104400                                  CHECKCC+0
425 010620  104423  010600                          ERROR,  T2              ;; *** CC BITS WRONG AFTER CCC ***
426
427                                         ;*****************************************************************************
                                            .SBTTL      T3 -- TEST REGISTER SELECTION
                                            ;*****************************************************************************
    010624  104420  000003                 T3:     BEGINTEST, 3            ;; CHECK SWITCH REGISTER OPTIONS.
428 010630  012700  000001                 REGS:   MOV     #1,R0           ; LOAD UP R0 THRU R4.
429 010634  012701  000004                         MOV     #4,R1
430 010640  012702  000020                         MOV     #20,R2
431 010644  012703  000100                         MOV     #100,R3
432 010650  012704  000400                         MOV     #400,R4
433 010654  012705  001000                         MOV     #1000,R5
434 010660  060600                                 ADD     R6,R0           ; R0 + #STACK...
435 010662  060500                                 ADD     R5,R0           ;...+ 1000...
436 010664  060400                                 ADD     R4,R0           ;...+ 400...
```

```
                                        ADD     R3,R0           ;...+ 100...
    438 010670  060200                  ADD     R2,R0           ;...+ 20...
    439 010672  060100                  ADD     R1,R0           ;...+ 4...
    440 010674  020027  005523          CMP     R0,@STACK+1525  ;...SHOULD = THIS.
    441 010700  001402                  BEQ     1$              ; BR IF OK.
    442 010702  104423  010624          ERROR,  T3              ;; *** REGISTER SELECTION FAILURE ***
    443 010706                      1$:
    444
    445                             ;*****************************************************************************
                                    .SBTTL      T4 -- JMP INSTRUCTION -- MODE 1
                                    ;*****************************************************************************
        010706  104420  000004     T4:     BEGINTEST, 4            ;; CHECK SWITCH REGISTER OPTIONS.
    446 010712  012700  010726     JMP1:   MOV     #1$,R0          ; TEST JUMP INSTRUCTION MODE 1
    447 010716  000277                      SCC
    448 010720  000110                      JMP     (R0)
    449 010722  104423  010706              ERROR,  T4              ;; *** MODE 1 JUMP FAILED ***
    450 010726  104417              1$:     CHECKCC+17
    451 010730  104423  010706              ERROR,  T4              ;; *** CC BITS ALTERED ON JMP ***
    452 010734  020027  010726              CMP     R0,#1$
    453 010740  001402                      BEQ     2$              ; CONTINUE IF R0 IS OK
    454 010742  104423  010706              ERROR,  T4              ;; *** R0 INCORRECT AFTER JUMP ***
    455 010746                      2$:
    456
    457                             ;*****************************************************************************
                                    .SBTTL      T5 -- JMP INSTRUCTION -- MODES 2 AND 3
                                    ;*****************************************************************************
        010746  104420  000005     T5:     BEGINTEST, 5            ;; CHECK SWITCH REGISTER OPTIONS.
    458 010752  012700  010766     JMP2:   MOV     #1$,R0          ; TEST JUMP INSTRUCTION MODE 2
    459 010756  000277                      SCC
Z   460 010760  000120                      JMP     (R0)+           ; POSSIBLE "Z" ASSEMBLY ERROR O.K.      ;JSD REV B
    461 010762  104423  010746              ERROR,  T5              ;; *** MODE 2 JUMP FAILED ***
    462 010766  104417              1$:     CHECKCC+17
    463 010770  104423  010746              ERROR,  T5              ;; *** CC BITS ALTERED AFTER JMP ***
    464 010774  020027  010770              CMP     R0,#1$+2        ; IS THERE AUTO INC.?
    465 011000  001402                      BEQ     JMP3
    466 011002  104423  010746              ERROR,  T5              ;; *** MODE 2 FAILED ON JUMP ***
    467
    468 011006  012767  011036  167772 JMP3:  MOV     #3$,TEMP        ; TEST JUMP INSTRUCTION MODE 3
    469 011014  012767  011052  167766       MOV     #4$,TEMP+2      ;
    470 011022  012700  001006               MOV     #TEMP,R0
    471 011026  000277                       SCC
    472 011030  000130                       JMP     @(R0)+          ;
    473 011032  104423  010746               ERROR,  T5              ;; *** MODE 3 JUMP FAILED ***
    474 011036  027067  000000  000006 3$:   CMP     @(R0),4$        ; IS THERE AUTO INC.?
    475 011044  001402                       BEQ     4$
    476 011046  104423  010746               ERROR,  T5              ;; *** MODE 3 JUMP FAILED ***
    477 011052                       4$:
    478
    479                             ;*****************************************************************************
                                    .SBTTL      T6 -- JMP INSTRUCTION -- MODES 4 AND 5
                                    ;*****************************************************************************
        011052  104420  000006     T6:     BEGINTEST, 6            ;; CHECK SWITCH REGISTER OPTIONS.
    480 011056  012700  011074     JMP4:   MOV     #3$,R0          ; TEST JUMP INSTRUCTION MODE 4
    481 011062  000277                      SCC
    482 011064  000140                      JMP     -(R0)
    483 011066  104423  011052              ERROR,  T6              ;; *** DIDN'T JUMP AT ALL ***
    484 011072  000402                      BR      4$              ; JUMP SHOULD LAND HERE
```

PC 11066 = DIDN'T JUMP AT ALL.

```
    485 011074                          3$:
        011074  104423  011052                  ERROR,  T6              ;; *** MODE 4 JUMP FAILED. ***
    486 011100  022700  011072          4$:     CMP     #3$-2,R0        ; CHECK R0
    487 011104  001402                          BEQ     JMP5
    488 011106  104423  011052                  ERROR,  T6              ;; *** R0 INCORRECT AFTER JUMP ***
    489
    490 011112  012767  011140  167670  JMP5:   MOV     #3$,TEMP1       ; TEST JUMP INSTRUCTION MODE 5
    491 011120  012700  001010                  MOV     #TEMP1,R0
    492 011124  012767  011144  167654          MOV     #4$,TEMP1-2     ;
    493 011132  000150                          JMP     @-(R0)
    494 011134  104423  011052                  ERROR,  T6              ;; *** MODE 5 JUMP DIDN'T ***
    495 011140                          3$:
        011140  104423  011052                  ERROR,  T6              ;; *** AUTO-DECR FAILED ON MODE 5 JUMP ***
    496 011144  022700  001006          4$:     CMP     #TEMP1-2,R0     ; CHECK R0
    497 011150  001402                          BEQ     5$
    498 011152  104423  011052                  ERROR,  T6              ;; *** R0 INCORRECT AFTER JMP5 ***
    499 011156                          5$:
    500
    501                                 ;*******************************************************************************
                                        .SBTTL      T7 -- JMP INSTRUCTION -- MODES 6 AND 7
                                        ;*******************************************************************************
        011156  104420  000007         T7:     BEGINTEST, 7            ;; CHECK SWITCH REGISTER OPTIONS.
    502 011162  012703  011204          JMP6:   MOV     #1$+6,R3
    503 011166  000163  177772                  JMP     -6(R3)
    504 011172  104423  011156                  ERROR,  T7              ;; *** MODE 6 JUMP FAILED ***
    505 011176  020327  011204          1$:     CMP     R3,#1$+6        ; CHECK R3
    506 011202  001402                          BEQ     2$
    507 011204  104423  011156                  ERROR,  T7              ;; *** R3 WRONG AFTER JMP6 ***
    508 011210  000167  000004          2$:     JMP     3$-.-4(PC)      ; TEST JUMP INSTRUCTION MODE 6
    509 011214  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
    510
    511 011220  012703  011234          3$:     MOV     #JMP7,R3        ; JUMP SHOULD LAND HERE
    512 011224  000163  000000                  JMP     0(R3)
    513 011230  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
    514
    515 011234  012703  001006          JMP7:   MOV     #TEMP,R3
    516 011240  012713  011254                  MOV     #1$,(R3)
    517 011244  000173  000000                  JMP     @(R3)
    518 011250  104423  011156                  ERROR,  T7              ;; *** MODE 7 JUMP FAILED ***
    519 011254  012713  011274          1$:     MOV     #3$,(R3)        ; TEST JUMP INSTRUCTION MODE 7
    520 011260  012700  001002                  MOV     #TEMP-4,R0
    521 011264  000170  000004                  JMP     @4(R0)
    522 011270  104423  011156                  ERROR,  T7              ;; *** JUMP FAILED ***
    523 011274  012767  011316  167504  3$:     MOV     #4$,TEMP        ; CONTINUE
    524 011302  012700  001006                  MOV     #TEMP,R0
    525 011306  000170  000000                  JMP     @0(R0)
    526 011312  104423  011156                  ERROR,  T7              ;; *** MODE 7 JUMP FAILED. ***
    527 011316                          4$:
    528
    529                                 ;*******************************************************************************
                                        .SBTTL      T10 -- JSR AND RTS INSTRUCTIONS
                                        ;*******************************************************************************
        011316  104420  000010         T10:    BEGINTEST, 10           ;; CHECK SWITCH REGISTER OPTIONS.
    530 011322  012706  003776          JSRTST: MOV     #STACK,SP       ; SET UP STACK POINTER.
    531 011326  000277                          SCC
    532 011330  004767  000004                  JSR     PC,2$
    533 011334                          1$:
```

PC 11334 = JSR INSTRUCTION FAILED.

```
     011334  104423  011316                    ERROR,  T10             ;; *** JSR INSTRUCTION FAILED ***
 534 011340  104417                    2$:     CHECKCC+17
 535 011342  104423  011316                    ERROR,  T10             ;; *** CC BITS CHANGED ON JSR ***
 536 011346  022706  003774            4$:     CMP     #STACK-2,SP
 537 011352  001402                            BEQ     5$              ; BR IF STACK IS RIGHT.
 538 011354  104423  011316                    ERROR,  T10             ;; *** STACK WRONG AFTER JSR ***
 539 011360  022716  011334            5$:     CMP     #1$,(SP)
 540 011364  001402                            BEQ     6$              ; BR IF STACKED PC IS RIGHT.
 541 011366  104423  011316                    ERROR,  T10             ;; *** SP DID NOT HAVE CORRECT RETURN ADDRESS ***
 542
 543 011372  012716  011404            6$:     MOV     #7$,(SP)        ; CHANGE RETURN PC.
 544 011376  000207                            RTS     PC
 545 011400  104423  011316                    ERROR,  T10             ;; *** RTS INSTRUCTION FAILED ***
 546 011404  022706  003776            7$:     CMP     #STACK,SP
 547 011410  001402                            BEQ     8$              ; BR IF STACK PROPERLY RESTORED.
 548 011412  104423  011316                    ERROR,  T10             ;; *** SP WAS NOT RESTORED BY RTS INSTRUCTION ***
 549 011416                            8$:
 550                                   ;
 551                                   .SBTTL
 552                                   .SBTTL  DESTINATION MODE 0
 553                                   ;
 554                                   ;*************************************************************************
                                       .SBTTL      T11 -- TSTB, CLRB, AND MOVB
                                       ;*************************************************************************
     011416  104420  000011            T11:    BEGINTEST, 11           ;; CHECK SWITCH REGISTER OPTIONS.
 555 011422  012706  003776            TSTB0:  MOV     #STACK,SP       ; INIT THE STACK.
 556 011426  000277                            SCC
 557 011430  105000                            CLRB    R0              ; CLEAR THE REGISTER
 558 011432  104404                            CHECKCC+4               ; CHECK FOR CC = 4
 559 011434  104423  011416                    ERROR,  T11             ;; *** CLRB FAILED ***
 560 011440  105700                            TSTB    R0              ; CHECK IT
 561 011442  104404                            CHECKCC+4               ; CHECK FOR CC = 4
 562 011444  104423  011416                    ERROR,  T11             ;; *** TSTB FAILED. ***
 563 011450  112701  000377                    MOVB    #377,R1         ; LOAD THE REGISTER
 564 011454  104410                            CHECKCC+10              ; CHECK FOR CC = 10
 565 011456  104423  011416                    ERROR,  T11             ;; *** INCORRECT CC BITS ***
 566 011462  105701                            TSTB    R1              ; CHECK IT
 567 011464  104410                            CHECKCC+10              ; CHECK FOR CC = 10
 568 011466  104423  011416                    ERROR,  T11             ;; *** INCORRECT CC BITS ***
 569
 570                                   ;*************************************************************************
                                       .SBTTL      T12 -- CMPB AND BISB
                                       ;*************************************************************************
     011472  104420  000012            T12:    BEGINTEST, 12           ;; CHECK SWITCH REGISTER OPTIONS.
 571 011476  000277                    CMPB0:  SCC
 572 011500  152702  000377                    BISB    #377,R2         ; LOAD REGISTER
 573 011504  104411                            CHECKCC+11              ; CHECK FOR CC = 11
 574 011506  104423  011472                    ERROR,  T12             ;; *** INCORRECT CC BITS ***
 575 011512  122702  000377                    CMPB    #377,R2         ; CHECK COMPARE
 576 011516  001402                            BEQ     2$              ; CONTINUE IF OK
 577 011520  104423  011472                    ERROR,  T12             ;; *** BISB OR CMPB INSTRUCTION FAILED ***
 578 011524  112700  000077            2$:     MOVB    #77,R0
 579 011530  120002                            CMPB    R0,R2           ; CHECK IT AGAIN
 580 011532  100002                            BPL     3$              ; CONTINUE IF OK
 581 011534  104423  011472                    ERROR,  T12             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
 582 011540  120200                    3$:     CMPB    R2,R0           ; ONCE MORE
 583 011542  100402                            BMI     4$              ; CONTINUE IF OK
```

PC 11544 = CMPB INSTRUCTION FAILED (WRONG CC).

```
    584 011544  104423  011472                  ERROR.  T12             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
    585 011550  112702  000377          4$:     MOVB    #377,R2         ; LOAD REGISTER, SIGN EXTEND
    586 011554  122702  000377                  CMPB    #377,R2         ; CHECK IF BYTE INSTRUCTION
    587 011560  001402                          BEQ     5$              ; CONTINUE IF OK
    588 011562  104423  011472                  ERROR.  T12             ;; *** CMPB BECAME CMP INSTRUCTION ***
    589 011566  112702  000377          5$:     MOVB    #377,R2         ; LOAD REGISTER, SIGN EXTEND
    590 011572  120227  000377                  CMPB    R2,#377         ; CHECK IF BYTE INSTRUCTION
    591 011576  001402                          BEQ     6$              ; CONTINUE IF OK
    592 011600  104423  011472                  ERROR.  T12             ;; *** MOVB OR CMPB FAILED ***
    593 011604                          6$:
    594
    595                             ;*************************************************************************
                                    .SBTTL      T13 -- BICB AND BITB
                                    ;*************************************************************************
        011604  104420  000013      T13:    BEGINTEST, 13           ;; CHECK SWITCH REGISTER OPTIONS.
    596 011610  112703  000377      BICB0:  MOVB    #377,R3         ; LOAD REGISTER
    597 011614  112700  000252              MOVB    #252,R0         ; PLACE #252 IN R0
    598 011620  000277                      SCC
    599 011622  140003                      BICB    R0,R3           ; CLEAR EVERY OTHER BIT
    600 011624  104401                      CHECKCC+1               ; CHECK FOR CC = 1
    601 011626  104423  011604              ERROR.  T13             ;; *** INCORRECT CC BITS ***
    602 011632  130003                      BITB    R0,R3           ; CHECK IT
    603 011634  001402                      BEQ     4$              ; CONTINUE IF OK
    604 011636  104423  011604              ERROR.  T13             ;; *** BICB OR BITB INSTRUCTION FAILED ***
    605 011642  132703  000125          4$:     BITB    #125,R3         ; CHECK IT
    606 011646  104401                      CHECKCC+1               ; CHECK FOR CC = 1
    607 011650  104423  011604              ERROR.  T13             ;; *** INCORRECT CC BITS ***
    608 011654  150003                      BISB    R0,R3           ; SET THE BITS THAT WERE CLEARED
    609 011656  100402                      BMI     6$
    610 011660  104423  011604              ERROR.  T13             ;; *** BISB INSTRUCTION FAILED ***
    611 011664  142703  000177      6$:     BICB    #177,R3         ; CLEAR ALL THE BITS EXCEPT FOR SIGN
    612 011670  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    613 011672  104423  011604              ERROR.  T13             ;; *** INCORRECT CC BITS ***
    614 011676  132703  000377              BITB    #377,R3         ; CHECK IT
    615 011702  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    616 011704  104423  011604              ERROR.  T13             ;; *** INCORRECT CC BITS ***
    617
    618
                                    ;*************************************************************************
                                    .SBTTL      T14 -- INCB AND DECB
                                    ;*************************************************************************
        011710  104420  000014      T14:    BEGINTEST, 14           ;; CHECK SWITCH REGISTER OPTIONS.
    619 011714  112704  000177      INCB0:  MOVB    #177,R4         ; R4 = 177
    620 011720  000261                      SEC
    621 011722  105204                      INCB    R4              ; ADD ONES INTO REG. 4
    622 011724  104413                      CHECKCC+13              ; CHECK FOR CC = 13
    623 011726  104423  011710              ERROR.  T14             ;; *** INCORRECT CC BITS ***
    624 011732  112704  000376              MOVB    #376,R4
    625 011736  105204                      INCB    R4
    626 011740  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    627 011742  104423  011710              ERROR.  T14             ;; *** INCORRECT CC BITS ***
    628 011746  105204                      INCB    R4
    629 011750  104405                      CHECKCC+5               ; CHECK FOR CC = 5
    630 011752  104423  011710              ERROR.  T14             ;; *** INCORRECT CC BITS ***
    631 011756  105204                      INCB    R4
    632 011760  104401                      CHECKCC+1               ; CHECK FOR CC = 1
    633 011762  104423  011710              ERROR.  T14             ;; *** INCORRECT CC BITS ***
    634 011766  122704  000001              CMPB    #1,R4           ; CHECK IT
```

```
PC 11762 = INCORRECT CC BITS.

    635 011772  001402                      BEQ     2$              ; CONTINUE IF OK
    636 011774  104423  011710              ERROR,  T14             ;; *** INCB INSTRUCTION FAILED ***
    637 012000  000261              2$:     SEC
    638 012002  105304                      DECB    R4              ; SUBTRACT ONES FROM REG. 4
    639 012004  104405                      CHECKCC+5               ; CHECK FOR CC = 5
    640 012006  104423  011710              ERROR,  T14             ;; *** INCORRECT CC BITS ***
    641 012012  105304                      DECB    R4
    642 012014  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    643 012016  104423  011710              ERROR,  T14             ;; *** INCORRECT CC BITS ***
    644 012022  012704  000200              MOV     #200,R4
    645 012026  105304                      DECB    R4
    646 012030  104403                      CHECKCC+3               ; CHECK FOR CC = 3
    647 012032  104423  011710              ERROR,  T14             ;; *** INCORRECT CC BITS ***
    648 012036  105304                      DECB    R4
    649 012040  104401                      CHECKCC+1               ; CHECK FOR CC = 1
    650 012042  104423  011710              ERROR,  T14             ;; *** INCORRECT CC BITS ***
    651
    652                         ;*****************************************************************************
                                .SBTTL      T15 -- COMB AND NEGB
                                ;*****************************************************************************
        012046  104420  000015  T15:       BEGINTEST, 15           ;; CHECK SWITCH REGISTER OPTIONS.
    653 012052  112703  000252  COMB0:     MOVB    #252,R3         ; LOAD EVERY OTHER BIT
    654 012056  000277                      SCC
    655 012060  105103                      COMB    R3              ; 1'S COMPLEMENT
    656 012062  104401                      CHECKCC+1               ; CHECK FOR CC = 1
    657 012064  104423  012046              ERROR,  T15             ;; *** INCORRECT CC BITS ***
    658 012070  122703  000125              CMPB    #125,R3         ; CHECK IT
    659 012074  001402                      BEQ     2$              ; CONTINUE IF OK
    660 012076  104423  012046              ERROR,  T15             ;; *** COMB INSTRUCTION FAILED ***
    661 012102  000277              2$:     SCC
    662 012104  105103                      COMB    R3              ; COMPLEMENT BACK
    663 012106  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    664 012110  104423  012046              ERROR,  T15             ;; *** INCORRECT CC BITS ***
    665 012114  122703  000252              CMPB    #252,R3         ; CHECK IT
    666 012120  001402                      BEQ     3$              ; CONTINUE IF OK
    667 012122  104423  012046              ERROR,  T15             ;; *** COMB INSTRUCTION FAILED ***
    668 012126  012703  000377  3$:         MOV     #377,R3
    669 012132  000277                      SCC
    670 012134  105103                      COMB    R3
    671 012136  104405                      CHECKCC+5               ; CHECK FOR CC = 5
    672 012140  104423  012046              ERROR,  T15             ;; *** INCORRECT CC BITS ***
    673
    674 012144  112700  000001  NEGB0:     MOVB    #1,R0           ; LOAD THE REGISTER
    675 012150  105400                      NEGB    R0              ; 2'S COMPLEMENT
    676 012152  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    677 012154  104423  012144              ERROR,  NEGB0           ;; *** INCORRECT CC BITS ***
    678 012160  122700  000377              CMPB    #377,R0         ; CHECK IT
    679 012164  001402                      BEQ     2$              ; CONTINUE IF OK
    680 012166  104423  012144              ERROR,  NEGB0           ;; *** NEGB INSTRUCTION FAILED ***
    681 012172  012700  000200  2$:         MOV     #200,R0
    682 012176  105400                      NEGB    R0              ; 2'S COMPLEMENT
    683 012200  104413                      CHECKCC+13              ; CHECK FOR CC = 13
    684 012202  104423  012144              ERROR,  NEGB0           ;; *** INCORRECT CC BITS ***
    685 012206  122700  000200              CMPB    #200,R0         ; CHECK IT
    686 012212  001402                      BEQ     3$              ; CONTINUE IF OK
    687 012214  104423  012144              ERROR,  NEGB0           ;; *** NEGB FAILED ***
    688 012220                      3$:
```

PC 12214 = NEGB FAILED.

```
689
690
                                ;*****************************************************************************
                                .SBTTL      T16 -- ROLB AND RORB
                                ;*****************************************************************************
    012220  104420  000016     T16:    BEGINTEST, 16           ;; CHECK SWITCH REGISTER OPTIONS.
691 012224  112701  000040     ROLB0:  MOVB    #40,R1          ; LOAD REGISTER
692 012230  000257                     CCC                     ; CLEAR FLAGS
693 012232  106101                     ROLB    R1              ; SHIFT
694 012234  106101                     ROLB    R1              ;
695 012236  104412                     CHECKCC+12              ; CHECK FOR CC = 12
696 012240  104423  012220             ERROR,  T16             ;; *** INCORRECT CC BITS ***
697 012244  122701  000200             CMPB    #200,R1         ; CHECK IT
698 012250  001402                     BEQ     1$              ; CONTINUE IF OK
699 012252  104423  012220             ERROR,  T16             ;; *** ROLB INSTRUCTION FAILED ***
700 012256  106101             1$:     ROLB    R1              ; SHIFT
701 012260  104407                     CHECKCC+7               ; CHECK FOR CC = 7
702 012262  104423  012220             ERROR,  T16             ;; *** INCORRECT CC BITS ***
703 012266  106101                     ROLB    R1              ; SHIFT
704 012270  122701  000001             CMPB    #1,R1           ; CHECK IT
705 012274  001402                     BEQ     2$              ; CONTINUE IF OK
706 012276  104423  012220             ERROR,  T16             ;; *** ROLB FAILED. ***
707 012302                     2$:
708 012302  112702  000004     RORB0:  MOVB    #4,R2           ; LOAD REGISTER
709 012306  000257                     CCC                     ; CLEAR FLAGS
710 012310  106002                     RORB    R2              ; SHIFT
711 012312  106002                     RORB    R2              ;
712 012314  122702  000001             CMPB    #1,R2           ; CHECK IT
713 012320  001402                     BEQ     1$              ; CONTINUE IF OK
714 012322  104423  012302             ERROR,  RORB0           ;; *** RORB INSTRUCTION FAILED ***
715 012326  106002             1$:     RORB    R2              ; SHIFT
716 012330  104407                     CHECKCC+7               ; CHECK FOR CC = 7
717 012332  104423  012302             ERROR,  RORB0           ;; *** INCORRECT CC BITS ***
718 012336  106002                     RORB    R2              ; SHIFT
719 012340  104412                     CHECKCC+12              ; CHECK FOR CC = 12
720 012342  104423  012302             ERROR,  RORB0           ;; *** INCORRECT CC BITS ***
721 012346  122702  000200             CMPB    #200,R2         ; CHECK IT
722 012352  001402                     BEQ     2$              ; CONTINUE IF OK
723 012354  104423  012302             ERROR,  RORB0           ;; *** RORB FAILED ***
724 012360                     2$:
725
726
                                ;*****************************************************************************
                                .SBTTL      T17 -- ASLB AND ASRB
                                ;*****************************************************************************
    012360  104420  000017     T17:    BEGINTEST, 17           ;; CHECK SWITCH REGISTER OPTIONS.
727 012364  112703  000040     ASLB0:  MOVB    #40,R3          ; LOAD REGISTER
728 012370  000257                     CCC                     ; CLEAR FLAGS
729 012372  106303                     ASLB    R3              ; SHIFT
730 012374  106303                     ASLB    R3              ;
731 012376  104412                     CHECKCC+12              ; CHECK FOR CC = 12
732 012400  104423  012360             ERROR,  T17             ;; *** INCORRECT CC BITS ***
733 012404  122703  000200             CMPB    #200,R3         ; CHECK IT
734 012410  001402                     BEQ     4$              ; CONTINUE IF OK
735 012412  104423  012360             ERROR,  T17             ;; *** ASLB INSTRUCTION FAILED ***
736 012416  106303             4$:     ASLB    R3              ; SHIFT
737 012420  104407                     CHECKCC+7               ; CHECK FOR CC = 7
738 012422  104423  012360             ERROR,  T17             ;; *** INCORRECT CC BITS ***
739 012426  106303                     ASLB    R3              ; SHIFT
```

```
PC 12422 = INCORRECT CC BITS.

    740 012430  104404                      CHECKCC#4               ; CHECK FOR CC = 4
    741 012432  104423  012360              ERROR,  T17             ;; *** INCORRECT CC BITS ***
    742
    743 012436  112704  000004  ASRB0:     MOVB    #4,R4           ; LOAD REGISTER
    744 012442  000257                      CCC                     ; CLEAR FLAGS
    745 012444  106204                      ASRB    R4              ; SHIFT
    746 012446  106204                      ASRB    R4              ;
    747 012450  122704  000001              CMPB    #1,R4           ; CHECK IT
    748 012454  001402                      BEQ     2$              ; CONTINUE IF OK
    749 012456  104423  012436              ERROR,  ASRB0           ;; *** ASRB INSTRUCTION FAILED ***
    750 012462  106204          2$:         ASRB    R4              ; SHIFT
    751 012464  104407                      CHECKCC#7               ; CHECK FOR CC = 7
    752 012466  104423  012436              ERROR,  ASRB0           ;; *** INCORRECT CC BITS ***
    753 012472  106204                      ASRB    R4              ; SHIFT
    754 012474  104404                      CHECKCC#4               ; CHECK FOR CC = 4
    755 012476  104423  012436              ERROR,  ASRB0           ;; *** INCORRECT CC BITS ***
    756 012502  112703  000202              MOVB    #202,R3         ; LOAD REGISTER
    757 012506  106203                      ASRB    R3              ; SHIFT
    758 012510  106203                      ASRB    R3              ;
    759 012512  104411                      CHECKCC#11              ; CHECK FOR CC = 11
    760 012514  104423  012436              ERROR,  ASRB0           ;; *** INCORRECT CC BITS ***
    761 012520  122703  000340              CMPB    #340,R3         ; CHECK IT
    762 012524  001402                      BEQ     3$              ; CONTINUE IF OK
    763 012526  104423  012436              ERROR,  ASRB0           ;; *** ASRB FAILED ***
    764 012532                  3$:
    765
    766                                     ;*************************************************************************
                                            .SBTTL      T20 -- ADCB AND SBCB
                                            ;*************************************************************************
        012532  104420  000020  T20:       BEGINTEST, 20           ;; CHECK SWITCH REGISTER OPTIONS.
    767 012536  105000          ADCB0:      CLRB    R0              ; CLEAR THE REGISTER
    768 012540  000257                      CCC                     ; CLEAR FLAGS
    769 012542  105500                      ADCB    R0              ; ADD C BIT = 0
    770 012544  104404                      CHECKCC#4               ; CHECK FOR CC = 4
    771 012546  104423  012532              ERROR,  T20             ;; *** INCORRECT CC BITS ***
    772 012552  000261                      SEC                     ; C=1
    773 012554  105500                      ADCB    R0              ; ADD C BIT=1
    774 012556  000261                      SEC                     ; C=1
    775 012560  105500                      ADCB    R0              ; AGAIN
    776 012562  104400                      CHECKCC#0               ; CHECK FOR CC = 0
    777 012564  104423  012532              ERROR,  T20             ;; *** INCORRECT CC BITS ***
    778 012570  122700  000002              CMPB    #2,R0           ; CHECK IT
    779 012574  001402                      BEQ     4$              ; CONTINUE IF OK
    780 012576  104423  012532              ERROR,  T20             ;; *** ADCB INSTRUCTION FAILED ***
    781 012602  112700  000177  4$:         MOVB    #177,R0         ; LOAD LARGEST POSITVE NUMBER
    782 012606  000261                      SEC                     ; C=1
    783 012610  105500                      ADCB    R0              ; ADD C BIT=1
    784 012612  104412                      CHECKCC#12              ; CHECK FOR CC = 12
    785 012614  104423  012532              ERROR,  T20             ;; *** INCORRECT CC BITS ***
    786 012620  122700  000200              CMPB    #200,R0         ; CHECK IT
    787 012624  001402                      BEQ     6$              ; CONTINUE IF OK
    788 012626  104423  012532              ERROR,  T20             ;; *** ADCB INSTRUCTION FAILED ***
    789 012632  112700  000377  6$:         MOVB    #377,R0         ; LOAD -1
    790 012636  000261                      SEC                     ; C=1
    791 012640  105500                      ADCB    R0              ; ADD C BIT=1
    792 012642  104405                      CHECKCC#5               ; CHECK FOR CC = 5
    793 012644  104423  012532              ERROR,  T20             ;; *** INCORRECT CC BITS ***
```

PC 12644 = INCORRECT CC BITS.

```
    794
    795 012650  112701  000003          SBCB0:  MOVB    #3,R1           ; LOAD REGISTER
    796 012654  000257                          CCC                     ; CLEAR FLAGS
    797 012656  105601                          SBCB    R1              ; SUBTRACT C BIT=0
    798 012660  104400                          CHECKCC+0               ; CHECK FOR CC = 0
    799 012662  104423  012650                  ERROR.  SBCB0           ;; *** INCORRECT CC BITS ***
    800 012666  122701  000003                  CMPB    #3,R1           ; CHECK IT
    801 012672  001402                          BEQ     2$              ; CONTINUE IF OK
    802 012674  104423  012650                  ERROR.  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
    803 012700  000261                  2$:     SEC                     ; C=1
    804 012702  105601                          SBCB    R1              ; SUBTRACT C BIT=1
    805 012704  000261                          SEC                     ; C=1
    806 012706  105601                          SBCB    R1              ;
    807 012710  104400                          CHECKCC+0               ; CHECK FOR CC = 0
    808 012712  104423  012650                  ERROR.  SBCB0           ;; *** INCORRECT CC BITS ***
    809 012716  122701  000001                  CMPB    #1,R1           ; CHECK IT
    810 012722  001402                          BEQ     3$              ; CONTINUE IF OK
    811 012724  104423  012650                  ERROR.  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
    812 012730  000261                  3$:     SEC                     ; C=1
    813 012732  105601                          SBCB    R1              ; SUBTRACT C BIT=1
    814 012734  104404                          CHECKCC+4               ; CHECK FOR CC = 4
    815 012736  104423  012650                  ERROR.  SBCB0           ;; *** INCORRECT CC BITS ***
    816 012742  000261                          SEC                     ; C=1
    817 012744  105601                          SBCB    R1              ; SUBTRACT C BIT = 1
    818 012746  104411                          CHECKCC+11              ; CHECK FOR CC = 11
    819 012750  104423  012650                  ERROR.  SBCB0           ;; *** INCORRECT CC BITS ***
    820 012754  122701  000377                  CMPB    #377,R1         ; CHECK IT
    821 012760  001402                          BEQ     4$              ; CONTINUE IF OK
    822 012762  104423  012650                  ERROR.  SBCB0           ;; *** SBCB INSTRUCTION FAILED ***
    823 012766  112701  000200          4$:     MOVB    #200,R1         ; LOAD R1
    824 012772  000261                          SEC                     ; C=1
    825 012774  105601                          SBCB    R1              ; SUBTRACT C BIT = 1
    826 012776  104402                          CHECKCC+2               ; CHECK FOR CC = 2
    827 013000  104423  012650                  ERROR.  SBCB0           ;; *** INCORRECT CC BITS ***
    828
    829
                                        ;*****************************************************************************
                                        .SBTTL      T21 -- TST, CLR, AND MOV
                                        ;*****************************************************************************
        013004  104420  000021          T21:    BEGINTEST, 21           ;; CHECK SWITCH REGISTER OPTIONS.
    830 013010  000277                  MOV0:   SCC
    831 013012  005000                          CLR     R0              ; CLEAR THE REGISTER
    832 013014  104404                          CHECKCC+4               ; CHECK FOR CC = 4
    833 013016  104423  013004                  ERROR.  T21             ;; *** INCORRECT CC BITS ***
    834 013022  005700                          TST     R0              ; CHECK IT
    835 013024  104404                          CHECKCC+4               ; CHECK FOR CC = 4
    836 013026  104423  013004                  ERROR.  T21             ;; *** INCORRECT CC BITS ***
    837 013032  012704  177777                  MOV     #177777,R4      ; LOAD THE REGISTER
    838 013036  010401                          MOV     R4,R1
    839 013040  104410                          CHECKCC+10              ; CHECK FOR CC = 10
    840 013042  104423  013004                  ERROR.  T21             ;; *** INCORRECT CC BITS ***
    841 013046  005701                          TST     R1              ; CHECK IT
    842 013050  104410                          CHECKCC+10              ; CHECK FOR CC = 10
    843 013052  104423  013004                  ERROR.  T21             ;; *** INCORRECT CC BITS ***
    844 013056  020401                          CMP     R4,R1           ; CHECK R1 TO CONTAIN PROPER DATA
    845 013060  001402                          BEQ     2$
    846 013062  104423  013004                  ERROR.  T21             ;; *** R1 WRONG AFTER MOV ***
    847 013066  000263                  2$:     SEVC                    ; SET V & C BITS
```

```
PC 13062 = R1 WRONG AFTER MOV.

848 013070  010000                      MOV     R0,R0
849 013072  104405                      CHECKCC+5
850 013074  104423  013004              ERROR,  T21             ;; *** INCORRECT CC BITS ***
851
852                                     ;*************************************************************************
                                        .SBTTL      T22 -- CMP AND BIS
                                        ;*************************************************************************
    013100  104420  000022      T22:    BEGINTEST, 22           ;; CHECK SWITCH REGISTER OPTIONS.
853 013104  000277              CMP0:   SCC
854 013106  012700  177777              MOV     #177777,R0      ; LOAD REGISTER
855 013112  050002                      BIS     R0,R2           ; CHECK THE BIS INSTRUCTION
856 013114  104411                      CHECKCC+11              ; CHECK FOR CC = 11
857 013116  104423  013100              ERROR,  T22             ;; *** INCORRECT CC BITS ***
858 013122  020002                      CMP     R0,R2           ; CHECK COMPARE
859 013124  001402                      BEQ     2$              ; CONTINUE IF OK
860 013126  104423  013100              ERROR,  T22             ;; *** BIS OR CMP INSTRUCTION FAILED ***
861 013132  022702  000077      2$:     CMP     #77,R2          ; CHECK IT AGAIN
862 013136  100002                      BPL     3$              ; CONTINUE IF OK
863 013140  104423  013100              ERROR,  T22             ;; *** CMP INSTRUCTION FAILED (WRONG CC) ***
864 013144  020227  000077      3$:     CMP     R2,#77          ; ONCE MORE
865 013150  100402                      BMI     4$              ; CONTINUE IF OK
866 013152  104423  013100              ERROR,  T22             ;; *** BIC OR CMP FAILED ***
867 013156                      4$:
868
869                                     ;*************************************************************************
                                        .SBTTL      T23 -- BIC AND BIT
                                        ;*************************************************************************
    013156  104420  000023      T23:    BEGINTEST, 23           ;; CHECK SWITCH REGISTER OPTIONS.
870 013162  012703  177777      BIC0:   MOV     #177777,R3      ; LOAD REGISTER
871 013166  012700  001006              MOV     #TEMP,R0        ; PLACE THE ADDRESS OF TEMP IN R0
872 013172  012710  125252              MOV     #125252,(R0)    ; SET (R0)
873 013176  000277                      SCC
874 013200  041003                      BIC     (R0),R3         ; CLEAR EVERY OTHER BIT
875 013202  104401                      CHECKCC+1               ; CHECK FOR CC = 1
876 013204  104423  013156              ERROR,  T23             ;; *** INCORRECT CC BITS ***
877 013210  031003                      BIT     (R0),R3         ; CHECK IT
878 013212  001402                      BEQ     1$              ; CONTINUE IF OK
879 013214  104423  013156              ERROR,  T23             ;; *** BIC OR BIT INSTRUCTION FAILED ***
880 013220  032703  052525      1$:     BIT     #52525,R3       ; CHECK IT
881 013224  104401                      CHECKCC+1               ; CHECK FOR CC = 1
882 013226  104423  013156              ERROR,  T23             ;; *** INCORRECT CC BITS ***
883 013232  052703  125252              BIS     #125252,R3      ; SET THE BITS THAT WERE CLEARED
884 013236  100402                      BMI     2$              ; CONTINUE IF OK
885 013240  104423  013156              ERROR,  T23             ;; *** BIT OR BIS INSTRUCTION FAILED ***
886 013244  042703  077777      2$:     BIC     #77777,R3       ; CLEAR ALL THE BITS EXCEPT FOR SIGN
887 013250  104411                      CHECKCC+11              ; CHECK FOR CC = 11
888 013252  104423  013156              ERROR,  T23             ;; *** INCORRECT CC BITS ***
889 013256  012700  177777              MOV     #177777,R0
890 013262  030003                      BIT     R0,R3           ; CHECK IT
891 013264  104411                      CHECKCC+11              ; CHECK FOR CC = 11
892 013266  104423  013156              ERROR,  T23             ;; *** INCORRECT CC BITS ***
893 013272  000263                      SEVC                    ; SET V & C BITS
894 013274  040000                      BIC     R0,R0
895 013276  104405                      CHECKCC+5               ; CHECK CC = 5
896 013300  104423  013156              ERROR,  T23             ;; *** INCORRECT CC BITS ***
897 013304  005700                      TST     R0              ; CHECK R0 TO CONTAIN 0
898 013306  001402                      BEQ     3$
```

PC 13310 = BIC FAILED.

```
                                        ERROR,  T23             ;; *** BIC FAILED ***
                                3$:
                                ;*************************************************************************
                                .SBTTL      T24 -- INC AND DEC
                                ;*************************************************************************
013314  104420  000024         T24:    BEGINTEST, 24           ;; CHECK SWITCH REGISTER OPTIONS.
013320  012704  077777         INC0:   MOV     #77777,R4       ; R4=77777
013324  000261                         SEC
013326  005204                         INC     R4              ; ADD ONES INTO REG. 4
013330  104413                         CHECKCC,13              ; CHECK FOR CC = 13
013332  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013336  012704  177776                 MOV     #177776,R4
013342  005204                         INC     R4
013344  104411                         CHECKCC,11              ; CHECK FOR CC = 11
013346  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013352  005204                         INC     R4
013354  104405                         CHECKCC,5               ; CHECK FOR CC = 5
013356  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013362  005204                         INC     R4
013364  104401                         CHECKCC,1               ; CHECK FOR CC = 1
013366  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013372  022704  000001                 CMP     #1,R4           ; CHECK IT
013376  001402                         BEQ     4$              ; FAILED
013400  104423  013314                 ERROR,  T24             ;; *** INC INSTRUCTION FAILED ***
013404  000261                 4$:     SEC
013406  005304                         DEC     R4              ; SUBTRACT ONES FROM REG. 4
013410  104405                         CHECKCC,5               ; CHECK FOR CC = 5
013412  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013416  005304                         DEC     R4
013420  104411                         CHECKCC,11              ; CHECK FOR CC = 11
013422  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013426  012704  100000                 MOV     #100000,R4
013432  005304                         DEC     R4              ;
013434  104403                         CHECKCC,3               ; CHECK FOR CC = 3
013436  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***
013442  005304                         DEC     R4              ;
013444  104401                         CHECKCC,1               ; CHECK FOR CC = 1
013446  104423  013314                 ERROR,  T24             ;; *** INCORRECT CC BITS ***

                                ;*************************************************************************
                                .SBTTL      T25 -- COM AND NEG
                                ;*************************************************************************
013452  104420  000025         T25:    BEGINTEST, 25           ;; CHECK SWITCH REGISTER OPTIONS.
013456  012703  125252         COM0:   MOV     #125252,R3      ; LOAD EVERY OTHER BIT
013462  000277                         SCC
013464  005103                         COM     R3              ; 1'S COMPLEMENT
013466  104401                         CHECKCC,1               ; CHECK FOR CC = 1
013470  104423  013452                 ERROR,  T25             ;; *** INCORRECT CC BITS ***
013474  022703  052525                 CMP     #52525,R3       ; CHECK IT
013500  001402                         BEQ     2$              ; CONTINUE IF OK
013502  104423  013452                 ERROR,  T25             ;; *** COM INSTRUCTION FAILED ***
013506  000277                 2$:     SCC
013510  005103                         COM     R3              ; COMPLEMENT BACK
013512  104411                         CHECKCC,11              ; CHECK FOR CC = 11
013514  104423  013452                 ERROR,  T25             ;; *** INCORRECT CC BITS ***
013520  022703  125252                 CMP     #125252,R3      ; CHECK IT
```

PC 13514 = INCORRECT CC BITS.

```
    950 013524  001402                      BEQ     3$              ; CONTINUE IF OK
    951 013526  104423  013452              ERROR,  T25             ;; *** COM INSTRUCTION FAILED ***
    952 013532  012703  177777      3$:     MOV     #177777,R3
    953 013536  000277                      SCC
    954 013540  005103                      COM     R3
    955 013542  104405                      CHECKCC+5               ; CHECK FOR CC = 5
    956 013544  104423  013452              ERROR,  T25             ;; *** INCORRECT CC BITS ***
    957
    958 013550  012700  000001      NEG0:   MOV     #1,R0           ; LOAD THE REGISTER
    959 013554  005400                      NEG     R0              ; 2'S COMPLEMENT
    960 013556  104411                      CHECKCC+11              ; CHECK FOR CC = 11
    961 013560  104423  013550              ERROR,  NEG0            ;; *** INCORRECT CC BITS ***
    962 013564  022700  177777              CMP     #177777,R0      ; CHECK IT
    963 013570  001402                      BEQ     2$              ; CONTINUE IF OK
    964 013572  104423  013550              ERROR,  NEG0            ;; *** NEG INSTRUCTION FAILED ***
    965 013576  012700  100000      2$:     MOV     #100000,R0
    966 013602  005400                      NEG     R0              ; 2'S COMPLEMENT
    967 013604  104413                      CHECKCC+13              ; CHECK FOR CC = 13
    968 013606  104423  013550              ERROR,  NEG0            ;; *** INCORRECT CC BITS ***
    969 013612  022700  100000              CMP     #100000,R0      ; CHECK IT
    970 013616  001402                      BEQ     3$              ; CONTINUE IF OK
    971 013620  104423  013550              ERROR,  NEG0            ;; *** NEG FAILED ***
    972 013624                      3$:
    973
    974                             ;*************************************************************************
                                    .SBTTL      T26 -- ROL AND ROR
                                    ;*************************************************************************
        013624  104420  000026      T26:    BEGINTEST, 26           ;; CHECK SWITCH REGISTER OPTIONS.
    975 013630  012701  020000      ROL0:   MOV     #20000,R1       ; LOAD REGISTER
    976 013634  000257                      CCC                     ; CLEAR FLAGS
    977 013636  006101                      ROL     R1              ; SHIFT
    978 013640  006101                      ROL     R1              ;
    979 013642  104412                      CHECKCC+12              ; CHECK FOR CC = 12
    980 013644  104423  013624              ERROR,  T26             ;; *** INCORRECT CC BITS ***
    981 013650  022701  100000              CMP     #100000,R1      ; CHECK IT
    982 013654  001402                      BEQ     1$              ; CONTINUE IF OK
    983 013656  104423  013624              ERROR,  T26             ;; *** ROL INSTRUCTION FAILED ***
    984 013662  006101              1$:     ROL     R1              ; SHIFT
    985 013664  104407                      CHECKCC+7               ; CHECK FOR CC = 7
    986 013666  104423  013624              ERROR,  T26             ;; *** INCORRECT CC BITS ***
    987 013672  006101                      ROL     R1              ; SHIFT
    988 013674  022701  000001              CMP     #1,R1           ; CHECK IT
    989 013700  001402                      BEQ     2$              ; CONTINUE IF OK
    990 013702  104423  013624              ERROR,  T26             ;; *** ROL FAILED ***
    991 013706                      2$:
    992 013706  012702  000004      RORO:   MOV     #4,R2           ; LOAD REGISTER
    993 013712  000257                      CCC                     ; CLEAR FLAGS
    994 013714  006002                      ROR     R2              ; SHIFT
    995 013716  006002                      ROR     R2              ;
    996 013720  022702  000001              CMP     #1,R2           ; CHECK IT
    997 013724  001402                      BEQ     1$              ; CONTINUE IF OK
    998 013726  104423  013706              ERROR,  RORO            ;; *** ROR INSTRUCTION FAILED ***
    999 013732  006002              1$:     ROR     R2              ; SHIFT
   1000 013734  104407                      CHECKCC+7               ; CHECK FOR CC = 7
   1001 013736  104423  013706              ERROR,  RORO            ;; *** INCORRECT CC BITS ***
   1002 013742  006002                      ROR     R2              ; SHIFT
   1003 013744  104412                      CHECKCC+12              ; CHECK FOR CC = 12
```

PC 13746 = INCORRECT CC BITS.

```
1004 013746  104423  013706                ERROR,  RORO            ;; *** INCORRECT CC BITS ***
1005 013752  022702  100000                CMP     #100000,R2      ; CHECK IT
1006 013756  001402                        BEQ     2$              ; CONTINUE IF OK
1007 013760  104423  013706                ERROR,  RORO            ;; *** ROR FAILED ***
1008 013764                          2$:
1009
1010                                 ;*******************************************************************************
                                     .SBTTL      T27 -- ASL AND ASR
                                     ;*******************************************************************************
     013764  104420  000027         T27:    BEGINTEST, 27           ;; CHECK SWITCH REGISTER OPTIONS.
1011 013770  012703  020000         ASL0:   MOV     #20000,R3       ; LOAD REGISTER
1012 013774  000257                        CCC                     ; CLEAR FLAGS
1013 013776  006303                        ASL     R3              ; SHIFT
1014 014000  006303                        ASL     R3              ;
1015 014002  104412                        CHECKCC+12              ; CHECK FOR CC = 12
1016 014004  104423  013764                ERROR,  T27             ;; *** INCORRECT CC BITS ***
1017 014010  022703  100000                CMP     #100000,R3      ; CHECK IT
1018 014014  001402                        BEQ     4$              ; CONTINUE IF OK
1019 014016  104423  013764                ERROR,  T27             ;; *** ASL INSTRUCTION FAILED ***
1020 014022  006303                  4$:   ASL     R3              ; SHIFT
1021 014024  104407                        CHECKCC+7               ; CHECK FOR CC = 7
1022 014026  104423  013764                ERROR,  T27             ;; *** INCORRECT CC BITS ***
1023 014032  006303                        ASL     R3              ; SHIFT
1024 014034  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1025 014036  104423  013764                ERROR,  T27             ;; *** INCORRECT CC BITS ***
1026
1027 014042  012704  000004         ASR0:   MOV     #4,R4           ; LOAD REGISTER
1028 014046  000257                        CCC                     ; CLEAR FLAGS
1029 014050  006204                        ASR     R4              ; SHIFT
1030 014052  006204                        ASR     R4              ;
1031 014054  022704  000001                CMP     #1,R4           ; CHECK IT
1032 014060  001402                        BEQ     2$              ; CONTINUE IF OK
1033 014062  104423  014042                ERROR,  ASR0            ;; *** ASR INSTRUCTION FAILED ***
1034 014066  006204                  2$:   ASR     R4              ; SHIFT
1035 014070  104407                        CHECKCC+7               ; CHECK FOR CC = 7
1036 014072  104423  014042                ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
1037 014076  006204                        ASR     R4              ; SHIFT
1038 014100  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1039 014102  104423  014042                ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
1040 014106  012703  100002                MOV     #100002,R3      ; LOAD REGISTER
1041 014112  006203                        ASR     R3              ; SHIFT
1042 014114  006203                        ASR     R3              ;
1043 014116  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1044 014120  104423  014042                ERROR,  ASR0            ;; *** INCORRECT CC BITS ***
1045 014124  022703  160000                CMP     #160000,R3      ; CHECK IT
1046 014130  001402                        BEQ     3$              ; CONTINUE IF OK
1047 014132  104423  014042                ERROR,  ASR0            ;; *** ASR FAILED ***
1048 014136                          3$:
1049
1050                                 ;*******************************************************************************
                                     .SBTTL      T30 -- ADC AND SBC
                                     ;*******************************************************************************
     014136  104420  000030         T30:    BEGINTEST, 30           ;; CHECK SWITCH REGISTER OPTIONS.
1051 014142  005000                 ADC0:   CLR     R0              ; CLEAR THE REGISTER
1052 014144  000257                        CCC                     ; CLEAR FLAGS
1053 014146  005500                        ADC     R0              ; ADD C BIT = 0
1054 014150  104404                        CHECKCC+4               ; CHECK FOR CC = 4
```

PC 14152 - INCORRECT CC BITS.

```
1055 014152  104423  014136                ERROR,  T30             ;;; *** INCORRECT CC BITS ***
1056 014156  000261                        SEC                     ; C=1
1057 014160  005500                        ADC     R0              ; ADD C BIT=1
1058 014162  000261                        SEC                     ; C=1
1059 014164  005500                        ADC     R0              ; AGAIN
1060 014166  104400                        CHECKCC+0               ; CHECK FOR CC = 0
1061 014170  104423  014136                ERROR,  T30             ;;; *** INCORRECT CC BITS ***
1062 014174  022700  000002                CMP     #2,R0           ; CHECK IT
1063 014200  001402                        BEQ     4$              ; CONTINUE IF OK
1064 014202  104423  014136                ERROR,  T30             ;;; *** ADC INSTRUCTION FAILED ***
1065 014206  012700  077777          4$:   MOV     #77777,R0       ; LOAD LARGEST POSITVE NUMBER
1066 014212  000261                        SEC                     ; C=1
1067 014214  005500                        ADC     R0              ;  DD C BIT=1
1068 014216  104412                        CHECKCC+12              ;  HECK FOR CC = 12
1069 014220  104423  014136                ERROR,  T30             ;  *** INCORRECT CC BITS ***
1070 014224  022700  100000                CMP     #100000,R0      ; HECK IT
1071 014230  001402                        BEQ     6$              ; FAILED
1072 014232  104423  014136                ERROR,  T30             ;;; *** ADC INSTRUCTION FAILED ***
1073 014236  012700  177777          6$:   MOV     #-1,R0          ; LOAD -1
1074 014242  000261                        SEC                     ; C=1
1075 014244  005500                        ADC     R0              ; ADD C BIT=1
1076 014246  104405                        CHECKCC+5               ; CHECK FOR CC = 5
1077 014250  104423  014136                ERROR,  T30             ;;; *** INCORRECT CC BITS ***
1078
1079 014254  012701  000003          SBC0: MOV     #3,R1           ; LOAD REGISTER
1080 014260  000257                        CCC                     ; CLEAR FLAGS
1081 014262  005601                        SBC     R1              ; SUBTRACT C BIT=0
1082 014264  104400                        CHECKCC+0               ; CHECK FOR CC = 0
1083 014266  104423  014254                ERROR,  SBC0            ;;; *** INCORRECT CC BITS ***
1084 014272  022701  000003                CMP     #3,R1           ; CHECK IT
1085 014276  001402                        BEQ     2$              ; CONTINUE IF OK
1086 014300  104423  014254                ERROR,  SBC0            ;;; *** SBC INSTRUCTION FAILED ***
1087 014304  000261                  2$:   SEC                     ; C=1
1088 014306  005601                        SBC     R1              ; SUBTRACT C BIT=1
1089 014310  000261                        SEC                     ; C=1
1090 014312  005601                        SBC     R1              ;
1091 014314  104400                        CHECKCC+0               ; CHECK FOR CC = 0
1092 014316  104423  014254                ERROR,  SBC0            ;;; *** INCORRECT CC BITS ***
1093 014322  022701  000001                CMP     #1,R1           ; CHECK IT
1094 014326  001402                        BEQ     3$              ; CONTINUE IF OK
1095 014330  104423  014254                ERROR,  SBC0            ;;; *** SBC INSTRUCTION FAILED ***
1096 014334  000261                  3$:   SEC                     ; C=1
1097 014336  005601                        SBC     R1              ; SUBTRACT C BIT=1
1098 014340  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1099 014342  104423  014254                ERROR,  SBC0            ;;; *** INCORRECT CC BITS ***
1100 014346  000261                        SEC                     ; C=1
1101 014350  005601                        SBC     R1              ; SUBTRACT C BIT = 1
1102 014352  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1103 014354  104423  014254                ERROR,  SBC0            ;;; *** INCORRECT CC BITS ***
1104 014360  022701  177777                CMP     #-1,R1          ; CHECK IT
1105 014364  001402                        BEQ     4$              ; CONTINUE IF F OK
1106 014366  104423  014254                ERROR,  SBC0            ;;; *** SBC INSTRUCTION FAILED ***
1107 014372  012701  100000          4$:   MOV     #100000,R1      ; LOAD R1
1108 014376  000261                        SEC                     ; C=1
1109 014400  005601                        SBC     R1              ; SUBTRACT C BIT = 1
1110 014402  104402                        CHECKCC+2               ; CHECK FOR CC = 2
1111 014404  104423  014254                ERROR,  SBC0            ;;; *** INCORRECT CC BITS ***
```

PC 14404 = INCORRECT CC BITS.

```
1112
1113                                    ;*******************************************************************************
                                        .SBTTL      T31 -- SXT, SWAB AND XOR
                                        ;*******************************************************************************
     014410  104420  000031            T31:    BEGINTEST, 31           ;; CHECK SWITCH REGISTER OPTIONS.
1114 014414  005002                    SXT0:   CLR     R2              ; CLEAR REGISTER
1115 014416  000277                            SCC
1116 014420  000254                            CLNZ
1117 014422  006702                            SXT     R2              ; SIGN EXTEND
1118 014424  104405                            CHECKCC,5               ; CHECK FOR CC = 5
1119 014426  104423  014410                    ERROR,  T31             ;; *** INCORRECT CC BITS ***
1120 014432  005702                            TST     R2              ; REG. 2 SHOULD STILL BE 0
1121 014434  001402                            BEQ     2$              ; CONTINUE IF OK
1122 014436  104423  014410                    ERROR,  T31             ;; *** SXT INSTRUCTION FAILED ***
1123 014442  000273                    2$:     SENVC                   ; SET N, V & C BITS
1124 014444  006702                            SXT     R2              ; SIGN EXTEND
1125 014446  104411                            CHECKCC,11              ; CHECK FOR CC = 11
1126 014450  104423  014410                    ERROR,  T31             ;; *** INCORRECT CC BITS ***
1127 014454  022702  177777                    CMP     #-1,R2          ; REG. 2 SHOULD NOW HAVE -1
1128 014460  001402                            BEQ     SWAB0           ; CONTINUE IF OK
1129 014462  104423  014410                    ERROR,  T31             ;; *** SXT FAILED ***
1130
1131 014466  012703  125125            SWAB0:  MOV     #125125,R3      ; LOAD BIT PATTERN INTO REGISTER
1132 014472  000277                            SCC
1133 014474  000250                            CLN
1134 014476  000303                            SWAB    R3              ; SWAP BYTES OF REGISTER
1135 014500  104410                            CHECKCC,10              ; CHECK FOR CC = 10
1136 014502  104423  014466                    ERROR,  SWAB0           ;; *** INCORRECT CC BITS ***
1137 014506  022703  052652                    CMP     #52652,R3       ; CHECK IT
1138 014512  001402                            BEQ     1$              ; CONTINUE IF OK
1139 014514  104423  014466                    ERROR,  SWAB0           ;; *** SWAB INSTRUCTION FAILED ***
1140 014520  012703  000377            1$:     MOV     #377,R3
1141 014524  000277                            SCC
1142 014526  000244                            CLZ
1143 014530  000303                            SWAB    R3
1144 014532  104404                            CHECKCC,4               ; CHECK FOR CC = 4
1145 014534  104423  014466                    ERROR,  SWAB0           ;; *** INCORRECT CC BITS ***
1146 014540  022703  177400                    CMP     #177400,R3
1147 014544  001402                            BEQ     XOR0
1148 014546  104423  014466                    ERROR,  SWAB0           ;; *** SWAB FAILED ***
1149
1150 014552  012704  177777            XOR0:   MOV     #-1,R4          ; LOAD REGISTERS
1151 014556  012703  177777                    MOV     #-1,R3          ;
1152 014562  000277                            SCC
1153 014564  074403                            XOR     R4,R3           ; SHOULD PRODUCE 0'S IN REG. 3
1154 014566  104405                            CHECKCC,5               ; CHECK FOR CC = 5
1155 014570  104423  014552                    ERROR,  XOR0            ;; *** INCORRECT CC BITS ***
1156 014574  012703  077777                    MOV     #77777,R3
1157 014600  010400                            MOV     R4,R0           ; PLACE A -1 IN R0
1158 014602  000263                            SEVC                    ; SET V & C BITS
1159 014604  000244                            CLZ
1160 014606  074003                            XOR     R0,R3
1161 014610  104411                            CHECKCC,11              ; CHECK FOR CC = 11
1162 014612  104423  014552                    ERROR,  XOR0            ;; *** INCORRECT CC BITS ***
1163 014616  012702  125252                    MOV     #125252,R2      ; LOAD REGISTERS
1164 014622  012704  052525                    MOV     #52525,R4       ;
1165 014626  000277                            SCC
```

PC 14612 = INCORRECT CC BITS.

```
1166 014630  074204                    XOR     R2,R4           ; SHOULD PRODUCE ALL 1'S IN REG. 4
1167 014632  104411                    CHECKCC*11              ; CHECK FOR CC = 11
1168 014634  104423  014552            ERROR,  XORO            ;; *** INCORRECT CC BITS ***
1169 014640  022704  177777            CMP     #-1,R4          ; CHECK IT
1170 014644  001402                    BEQ     1$              ; CONTINUE IF OK
1171 014646  104423  014552            ERROR,  XORO            ;; *** XOR FAILED ***
1172 014652                    1$:
1173
1174                                   ;*************************************************************************
                                       .SBTTL      T32 -- ADD AND SUB
                                       ;*************************************************************************
     014652  104420  000032    T32:    BEGINTEST, 32           ;; CHECK SWITCH REGISTER OPTIONS.
1175 014656  012701  021421    ADDO:   MOV     #21421,R1       ; LOAD REGISTERS
1176 014662  060101                    ADD     R1,R1           ; ADD
1177 014664  104400                    CHECKCC*0               ; CHECK FOR CC = 0
1178 014666  104423  014652            ERROR,  T32             ;; *** INCORRECT CC BITS ***
1179 014672  022701  043042            CMP     #43042,R1       ; CHECK IT
1180 014676  001402                    BEQ     1$              ; CONTINUE IF OK
1181 014700  104423  014652            ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
1182 014704  012700  156357    1$:     MOV     #-21421,R0      ; LOAD REGISTERS
1183 014710  060000                    ADD     R0,R0           ; ADD
1184 014712  104411                    CHECKCC*11              ; CHECK FOR CC = 11
1185 014714  104423  014652            ERROR,  T32             ;; *** INCORRECT CC BITS ***
1186 014720  022700  134736            CMP     #-43042,R0      ; CHECK IT
1187 014724  001402                    BEQ     2$              ; CONTINUE IF OK
1188 014726  104423  014652            ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
1189 014732  012702  100000    2$:     MOV     #100000,R2      ; LOAD REGISTERS
1190 014736  060202                    ADD     R2,R2           ; ADD SHOULD RESULT AS 0'S
1191 014740  104407                    CHECKCC*7               ; CHECK FOR CC = 7
1192 014742  104423  014652            ERROR,  T32             ;; *** INCORRECT CC BITS ***
1193 014746  012704  021421            MOV     #21421,R4       ; LOAD REGISTERS
1194 014752  012701  156357            MOV     #-21421,R1      ;
1195 014756  060401                    ADD     R4,R1           ; ADD SHOULD RESULT AS 0'S
1196 014760  001402                    BEQ     3$              ; CONTINUE IF OK
1197 014762  104423  014652            ERROR,  T32             ;; *** ADD INSTRUCTION FAILED ***
1198 014766  005404            3$:     NEG     R4              ; SWITCH SOURCE AND DESTINATION
1199 014770  012701  021421            MOV     #21421,R1       ;
1200 014774  060104                    ADD     R1,R4           ; SHOULD RESULT AS 0'S
1201 014776  001402                    BEQ     SUBO            ; CONTINUE IF OK
1202 015000  104423  014652            ERROR,  T32             ;; *** ADD FAILED ***
1203
1204 015004  012702  021421    SUBO:   MOV     #21421,R2       ; LOAD REGISTERS
1205 015010  012703  156357            MOV     #-21421,R3      ;
1206 015014  160203                    SUB     R2,R3           ; RESULT SHOULD=-43042
1207 015016  104410                    CHECKCC*10              ; CHECK FOR CC = 10
1208 015020  104423  015004            ERROR,  SUBO            ;; *** INCORRECT CC BITS ***
1209 015024  022703  134736            CMP     #-43042,R3      ; CHECK IT
1210 015030  001402                    BEQ     4$              ; CONTINUE IF OK
1211 015032  104423  015004            ERROR,  SUBO            ;; *** SUB INSTRUCTION FAILED ***
1212 015036  012703  021421    4$:     MOV     #21421,R3       ; LOAD REGISTER
1213 015042  010204                    MOV     R2,R4           ; NOW R4 = #21421
1214 015044  160403                    SUB     R4,R3           ; RESULT SHOULD=0
1215 015046  001402                    BEQ     6$
1216 015050  104423  015004            ERROR,  SUBO            ;; *** SUB NSTRUCTION FAILED ***
1217 015054  012703  177777    6$:     MOV     #-1,R3          ; LOAD REGISTERS
1218 015060  012702  077777            MOV     #77777,R2       ; LOAD REGISTERS
1219 015064  160302                    SUB     R3,R2           ; RESULT SHOULD BE 100000 AND OVERFLOW
```

```
 1220 015066  104413                            CHECKCC+13              ; CHECK FOR CC = 13
 1221 015070  104423  015004                    ERROR,  SUBO            ;; *** INCORRECT CC BITS ***
 1222 015074  022702  100000                    CMP     #100000,R2      ; CHECK IT
 1223 015100  001402                            BEQ     8$              ; CONTINUE IF OK
 1224 015102  104423  015004                    ERROR,  SUBO            ;; *** SUB INSTRUCTION FAILED ***
 1225 015106  012704  177777          8$:       MOV     #-1,R4
 1226 015112  160304                            SUB     R3,R4
 1227 015114  104404                            CHECKCC+4               ; CHECK FOR CC = 4
 1228 015116  104423  015004                    ERROR,  SUBO            ;; *** INCORRECT CC BITS ***
 1229
 1230                                ;*************************************************************************
                                     .SBTTL      T33 -- MTPS AND MFPS
                                     ;*************************************************************************
      015122  104420  000033        T33:      BEGINTEST, 33             ;; CHECK SWITCH REGISTER OPTIONS.
 1231 015126  012701  177777        TPSW:     MOV     #177777,R1
 1232 015132  005000                            CLR     R0
 1233 015134  106400                            MTPS    R0              ; SET PSW TO 0
 1234 015136  104400                            CHECKCC+0               ; CHECK FOR CC = 0
 1235 015140  104423  015122                    ERROR,  T33             ;; *** INCORRECT CC BITS ***
 1236 015144  106701                            MFPS    R1              ; MOVE PSW TO R1
 1237 015146  001402                            BEQ     2$              ; BR IF BIT 8 OF PSW WAS EXTENDED IN R1
 1238 015150  104423  015122                    ERROR,  T33             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
 1239 015154  104404                2$:       CHECKCC+4               ; CHECK FOR CC = 4
 1240 015156  104423  015122                    ERROR,  T33             ;; *** INCORRECT CC BITS ***
 1241 015162  012700  000337                    MOV     #337,R0
 1242 015166  106400                            MTPS    R0              ; EXPECT 317 SINCE MTPS DOESN'T SET T BIT
 1243 015170  104417                            CHECKCC+17              ; CHECK FOR CC = 17
 1244 015172  104423  015122                    ERROR,  T33             ;; *** INCORRECT CC BITS ***
 1245 015176  106701                            MFPS    R1              ; MOVE PSW TO R1
 1246 015200  104411                            CHECKCC+11              ; CHECK FOR CC = 11 (C BIT SHOULD NOT BE EFFECTED BY MFPS)
 1247 015202  104423  015122                    ERROR,  T33             ;; *** INCORRECT CC BITS ***
 1248 015206  022701  177717                    CMP     #177717,R1      ; CHECK TO SEE IF BIT 8 OF PSW WAS EXTENDED THRU R1
 1249 015212  001402                            BEQ     3$
 1250 015214  104423  015122                    ERROR,  T33             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
 1251 015220                        3$:
 1252                                ;
 1253                                .SBTTL
 1254                                .SBTTL  NON-ZERO ADDRESS MODES
 1255                                ;
 1256                                ;*************************************************************************
                                     .SBTTL      T34 -- TEST MODES 0 AND 1 USING MOVB AND MOV
                                     ;*************************************************************************
      015220  104420  000034        T34:      BEGINTEST, 34             ;; CHECK SWITCH REGISTER OPTIONS.
 1257 015224  112700  000252        MODE0:    MOVB    #252,R0         ; LOAD REGISTERS
 1258 015230  110001                            MOVB    R0,R1           ;
 1259 015232  110102                            MOVB    R1,R2           ;
 1260 015234  122702  000252                    CMPB    #252,R2         ; CHECK IT
 1261 015240  001402                            BEQ     1$              ; OK, CONTINUE
 1262 015242  104423  015220                    ERROR,  T34             ;; *** MOVB INSTRUCTION FAILED IN MODE 0 ***
 1263 015246  012700  125252        1$:       MOV     #125252,R0      ; LOAD REGISTERS
 1264 015252  010001                            MOV     R0,R1           ;
 1265 015254  010102                            MOV     R1,R2           ;
 1266 015256  022702  125252                    CMP     #125252,R2      ; CHECK IT
 1267 015262  001402                            BEQ     MODE1           ; OK, CONTINUE
 1268 015264  104423  015220                    ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 0 ***
 1269
 1270 015270  012700  001006        MODE1:    MOV     #TEMP,R0        ; LOAD ADDRESSES INTO REGS.
```

PC 15264 = MOV INSTRUCTION FAILED IN MODE 0.

```
1271 015274  012701  001010                    MOV     #TEMP1,R1       ;
1272 015300  012702  001012                    MOV     #TEMP2,R2       ;
1273 015304  005067  163502                    CLR     TEMP2           ; START CLEAN
1274 015310  112710  000125                    MOVB    #125,(R0)       ; LOAD THE LOCATIONS
1275 015314  111011                            MOVB    (R0),(R1)       ; TEMP => TEMP1
1276 015316  111112                            MOVB    (R1),(R2)       ; TEMP1 => TEMP2
1277 015320  122767  000125  163464            CMPB    #125,TEMP2      ; CHECK IT
1278 015326  001402                            BEQ     1$              ; OK, CONTINUE
1279 015330  104423  015220                    ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 1 ***
1280 015334  012710  052525            1$:     MOV     #52525,(R0)     ; LOAD THE LOCATIONS
1281 015340  011011                            MOV     (R0),(R1)       ; TEMP => TEMP1
1282 015342  011112                            MOV     (R1),(R2)       ; TEMP1 => TEMP2
1283 015344  022767  052525  163440            CMP     #52525,TEMP2    ; CHECK IT
1284 015352  001402                            BEQ     2$              ; OK, CONTINUE
1285 015354  104423  015220                    ERROR,  T34             ;; *** MOV INSTRUCTION FAILED IN MODE 1 ***
1286 015360                            2$:
1287
1288                                    ;*******************************************************************************
                                       .SBTTL       T35 -- TEST MODE 2 USING MOVB AND MOV
                                       ;*******************************************************************************
     015360  104420  000035            T35:    BEGINTEST, 35           ;; CHECK SWITCH REGISTER OPTIONS.
1289 015364  012700  001006            MODE2:  MOV     #TEMP,R0        ; LOAD ADDRESSES
1290 015370  012701  001010                    MOV     #TEMP1,R1       ;
1291 015374  012702  001012                    MOV     #TEMP2,R2
1292 015400  105022                            CLRB    (R2)+           ; START CLEAN
1293 015402  112710  000252                    MOVB    #252,(R0)       ; LOAD THE LOCATIONS
1294 015406  112021                            MOVB    (R0)+,(R1)+     ; TEMP => TEMP1
1295 015410  105201                            INCB    R1              ; MAKE IT EVEN
1296 015412  111167  163370                    MOVB    (R1),TEMP       ; MORE 0'S INTO TEMP
1297 015416  105200                            INCB    R0              ; MAKE IT EVEN
1298 015420  112021                            MOVB    (R0)+,(R1)+     ; TEMP1 => TEMP2
1299 015422  124227  000252                    CMPB    -(R2),#252      ; CHECK IT
1300 015426  001003                            BNE     1$              ; FAILED
1301 015430  105767  163352                    TSTB    TEMP            ; CHECK IT
1302 015434  001402                            BEQ     2$              ; OK, CONTINUE
1303 015436                            1$:
     015436  104423  015360                    ERROR,  T35             ;; *** INSTRUCTIONS FAILED IN MODE 2 ***
1304
1305 015442  005741                    2$:     TST     -(R1)
1306 015444  005022                            CLR     (R2)+           ; START CLEAN
1307 015446  012740  125252                    MOV     #125252,-(R0)   ; LOAD LOCATIONS
1308 015452  012020                            MOV     (R0)+,(R0)+     ; TEMP => TEMP1
1309 015454  011067  163326                    MOV     (R0),TEMP       ; 0 => TEMP
1310 015460  012121                            MOV     (R1)+,(R1)+     ; 125252 => TEMP2
1311 015462  024227  125252                    CMP     -(R2),#125252   ; CHECK IT
1312 015466  001003                            BNE     3$              ; FAILED
1313 015470  005767  163312                    TST     TEMP            ; CHECK IT
1314 015474  001402                            BEQ     4$              ; OK, CONTINUE
1315 015476                            3$:
     015476  104423  015360                    ERROR,  T35             ;; *** INSTRUCTIONS FAILED IN MODE 2 ***
1316 015502                            4$:
1317
1318                                    ;*******************************************************************************
                                       .SBTTL       T36 -- TEST MODE 3 USING MOVB AND MOV
                                       ;*******************************************************************************
     015502  104420  000036            T36:    BEGINTEST, 36           ;; CHECK SWITCH REGISTER OPTIONS.
1319 015506  012767  001006  163264    MODE3:  MOV     #TEMP,ADR       ; LOAD ADDRESSES
```

T36 -- TEST MODE 3 USING MOVB AND MOV

```
1320 015514  012767  001010  163260          MOV     #TEMP1,ADR1     ;
1321 015522  012767  001012  163254          MOV     #TEMP2,ADR2     ;
1322 015530  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1323 015534  012701  001002                  MOV     #ADR1,R1        ;
1324 015540  105067  163246                  CLRB    TEMP2           ; START CLEAN
1325 015544  112767  000125  163234          MOVB    #125,TEMP       ;
1326 015552  113031                          MOVB    @(R0)+,@(R1)+   ; TEMP => TEMP1
1327 015554  113167  163226                  MOVB    @(R1)+,TEMP     ; TEMP2 => TEMP
1328 015560  113030                          MOVB    @(R0)+,@(R0)+   ; TEMP1 => TEMP2
1329 015562  122767  000125  163222          CMPB    #125,TEMP2      ; CHECK IT
1330 015570  001003                          BNE     1$              ; FAILED
1331 015572  105767  163210                  TSTB    TEMP            ; CHECK IT
1332 015576  001402                          BEQ     2$              ; OK, CONTINUE
1333 015600                          1$:
     015600  104423  015502                  ERROR,  T36             ;; *** INSTRUCTIONS FAILED IN MODE 3 ***
1334
1335 015604  005067  163202          2$:     CLR     TEMP2           ; START CLEAN
1336 015610  012767  052525  163170          MOV     #52525,TEMP     ; LOAD LOCATIONS
1337 015616  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1338 015622  012701  001002                  MOV     #ADR1,R1        ;
1339 015626  013030                          MOV     @(R0)+,@(R0)+   ; TEMP => TEMP1
1340 015630  013067  163152                  MOV     @(R0)+,TEMP     ; TEMP2 => TEMP
1341 015634  013131                          MOV     @(R1)+,@(R1)+   ; TEMP1 => TEMP2
1342 015636  022767  052525  163146          CMP     #52525,TEMP2    ; CHECK IT
1343 015644  001003                          BNE     3$              ; FAILED
1344 015646  005767  163134                  TST     TEMP            ; CHECK IT
1345 015652  001402                          BEQ     4$              ; OK, CONTINUE
1346 015654                          3$:
     015654  104423  015502                  ERROR,  T36             ;; *** INSTRUCTIONS FAILED IN MODE 3 ***
1347 015660                          4$:
1348
1349                                 ;*******************************************************************************
                                     .SBTTL      T37 -- TEST MODE 4 USING MOVB AND MOV
                                     ;*******************************************************************************
     015660  104420  000037          T37:    BEGINTEST, 37           ;; CHECK SWITCH REGISTER OPTIONS.
1350 015664  105067  163116          MODE4:  CLRB    TEMP            ; START CLEAN
1351 015670  012700  001006                  MOV     #TEMP,R0        ; LOAD ADDRESSES
1352 015674  012701  001010                  MOV     #TEMP1,R1       ;
1353 015700  012702  001012                  MOV     #TEMP2,R2       ;
1354 015704  005202                          INC     R2              ; ADJUST THE POINTER
1355 015706  021267  163101                  CMP     (R2),TEMP2+1
1356 015712  001402                          BEQ     1$
1357 015714  104423  015660                  ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
1358 015720  112742  000252          1$:     MOVB    #252,-(R2)      ; LOAD TEMP2
1359 015724  005201                          INC     R1              ; ADJUST THE POINTERS
1360 015726  005202                          INC     R2              ;
1361 015730  114241                          MOVB    -(R2),-(R1)     ; TEMP2 => TEMP1
1362 015732  005200                          INC     R0              ; ADJUST THE POINTERS
1363 015734  005202                          INC     R2              ;
1364 015736  114042                          MOVB    -(R0),-(R2)     ; TEMP => TEMP2
1365 015740  105200                          INCB    R0              ; ADJUST THE POINTERS
1366 015742  021067  163041                  CMP     (R0),TEMP+1
1367 015746  001402                          BEQ     2$
1368 015750  104423  015660                  ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
1369 015754  105201                  2$:     INCB    R1              ;
1370 015756  114140                          MOVB    -(R1),-(R0)     ; TEMP1 => TEMP
1371 015760  122767  000252  163020          CMPB    #252,TEMP       ; CHECK IT
```

PC 15750 = INSTRUCTIONS FAILED IN MODE 4.

```
 1372 015766  001003                          BNE     3$              ; FAILED
 1373 015770  105767  163016                  TSTB    TEMP2           ; CHECK IT
 1374 015774  001402                          BEQ     4$              ; OK, CONTINUE
 1375 015776                          3$:
      015776  104423  015660                  ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
 1376
 1377 016002  005067  163000          4$:     CLR     TEMP            ; START CLEAN
 1378 016006  012700  001006                  MOV     #TEMP,R0        ; LOAD ADDRESSES
 1379 016012  012701  001010                  MOV     #TEMP1,R1       ;
 1380 016016  012702  001012                  MOV     #TEMP2,R2       ;
 1381 016022  005722                          TST     (R2)+           ; ADJUST THE POINTER
 1382 016024  021267  162764                  CMP     (R2),TEMP2+2
 1383 016030  001402                          BEQ     5$
 1384 016032  104423  015660                  ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
 1385 016036  012742  125252          5$:     MOV     #125252,-(R2)   ; LOAD TEMP2
 1386 016042  005721                          TST     (R1)+           ; ADJUST THE POINTERS
 1387 016044  005722                          TST     (R2)+           ;
 1388 016046  014241                          MOV     -(R2),-(R1)     ; TEMP2 => TEMP1
 1389 016050  005720                          TST     (R0)+           ; ADJUST POINTERS
 1390 016052  005722                          TST     (R2)+           ;
 1391 016054  014042                          MOV     -(R0),-(R2)     ; TEMP => TEMP2
 1392 016056  005720                          TST     (R0)+           ; ADJUST THE POINTERS
 1393 016060  005721                          TST     (R1)+           ;
 1394 016062  014140                          MOV     -(R1),-(R0)     ; TEMP1 => TEMP
 1395 016064  022767  125252  162714          CMP     #125252,TEMP    ; CHECK IT
 1396 016072  001003                          BNE     6$              ; FAILED
 1397 016074  005767  162712                  TST     TEMP2           ; CHECK IT
 1398 016100  001402                          BEQ     7$              ; OK, CONTINUE
 1399 016102                          6$:
      016102  104423  015660                  ERROR,  T37             ;; *** INSTRUCTIONS FAILED IN MODE 4 ***
 1400 016106                          7$:
 1401
 1402                                         ;*************************************************************************
                                              .SBTTL      T40 -- TEST MODE 5 USING MOVB AND MOV
                                              ;*************************************************************************
      016106  104420  000040          T40:    BEGINTEST, 40           ;; CHECK SWITCH REGISTER OPTIONS.
 1403 016112  105067  162670          MODE5:  CLRB    TEMP            ; START CLEAN
 1404 016116  012767  001006  162654          MOV     #TEMP,ADR       ; LOAD ADDRESSES
 1405 016124  012767  001010  162650          MOV     #TEMP1,ADR1     ;
 1406 016132  012767  001012  162644          MOV     #TEMP2,ADR2     ;
 1407 016140  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
 1408 016144  012701  001002                  MOV     #ADR1,R1        ;
 1409 016150  012702  001004                  MOV     #ADR2,R2        ;
 1410 016154  005722                          TST     (R2)+           ; ADJUST THE POINTER
 1411 016156  112752  000125                  MOVB    #125,@-(R2)     ; LOAD TEMP2
 1412 016162  022122                          CMP     (R1)+,(R2)+     ; ADJUST THE POINTERS
 1413 016164  115251                          MOVB    @-(R2),@-(R1)   ; TEMP2 => TEMP1
 1414 016166  022022                          CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
 1415 016170  115052                          MOVB    @-(R0),@-(R2)   ; TEMP => TEMP2
 1416 016172  022022                          CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
 1417 016174  125052                          CMPB    @-(R0),@-(R2)   ;CHECK IT
 1418 016176  001402                          BEQ     1$
 1419 016200  104423  016106                  ERROR,  T40             ;; *** INSTRUCTION FAILED IN MODE 5 ***
 1420 016204  022120                  1$:     CMP     (R1)+,(R0)+     ; ADJUST THE POINTERS
 1421 016206  115150                          MOVB    @-(R1),@-(R0)   ; TEMP1 => TEMP
 1422 016210  122767  000125  162570          CMPB    #125,TEMP       ; CHECK IT
 1423 016216  001003                          BNE     2$              ; FAILED
```

```
PC 16200 = INSTRUCTION FAILED IN MODE 5.

 1424 016220  105767  162566                    TSTB    TEMP2           ; CHECK IT
 1425 016224  001402                            BEQ     3$              ; OK, CONTINUE
 1426 016226                              2$:
      016226  104423  016106                    ERROR,  T40             ;; *** INSTRUCTIONS FAILED IN MODE 5 ***
 1427
 1428 016232  005067  162550              3$:   CLR     TEMP            ; START CLEAN
 1429 016236  012700  001000                    MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
 1430 016242  012701  001002                    MOV     #ADR1,R1        ;
 1431 016246  012702  001004                    MOV     #ADR2,R2        ;
 1432 016252  005722                            TST     (R2)+           ; ADJUST THE POINTER
 1433 016254  012752  052525                    MOV     #52525,@-(R2)   ; LOAD TEMP2
 1434 016260  022122                            CMP     (R1)+,(R2)+     ; ADJUST THE POINTERS
 1435 016262  015251                            MOV     @-(R2),@-(R1)   ; TEMP2 => TEMP1
 1436 016264  022022                            CMP     (R0)+,(R2)+     ; ADJUST THE POINTERS
 1437 016266  015052                            MOV     @-(R0),@-(R2)   ; TEMP => TEMP2
 1438 016270  022021                            CMP     (R0)+,(R1)+     ; ADJUST THE POINTERS
 1439 016272  015150                            MOV     @-(R1),@-(R0)   ; TEMP1 => TEMP
 1440 016274  022767  052525  162504            CMP     #52525,TEMP     ; CHECK IT
 1441 016302  001003                            BNE     4$              ; FAILED
 1442 016304  005767  162502                    TST     TEMP2           ; CHECK IT
 1443 016310  001402                            BEQ     5$              ; OK, CONTINUE
 1444 016312                              4$:
      016312  104423  016106                    ERROR,  T40             ;; *** INSTRUCTIONS FAILED IN MODE 5 ***
 1445 016316                              5$:
 1446
 1447                                    ;*******************************************************************************
                                         .SBTTL      T41 -- TEST MODE 6 USING MOVB AND MOV
                                         ;*******************************************************************************
      016316  104420  000041             T41:    BEGINTEST, 41           ;; CHECK SWITCH REGISTER OPTIONS.
 1448 016322  005067  162464             MODE6:  CLR     TEMP2           ; START CLEAN
 1449 016326  012700  001006                     MOV     #TEMP,R0        ; LOAD ADDRESSES
 1450 016332  012701  001010                     MOV     #TEMP1,R1       ;
 1451 016336  012702  001012                     MOV     #TEMP2,R2       ;
 1452 016342  112760  000252  000000            MOVB    #252,0(R0)      ; LOAD TEMP (LOW BYTE)
 1453 016350  112760  000252  000001            MOVB    #252,1(R0)      ; LOAD TEMP (HIGH BYTE)
 1454 016356  022767  125252  162422            CMP     #125252,TEMP    ; CHECK IT
 1455 016364  001012                            BNE     1$              ; FAILED
 1456 016366  116062  000001  000000            MOVB    1(R0),0(R2)     ; TEMP(H) => TEMP2(L)
 1457 016374  116160  000002  000005            MOVB    2(R1),5(R0)     ; TEMP2(L) => TEMP2(H)
 1458 016402  022767  125252  162402            CMP     #125252,TEMP2   ; CHECK IT
 1459 016410  001402                            BEQ     2$              ; OK, CONTINUE
 1460 016412                              1$:
      016412  104423  016316                    ERROR,  T41             ;; *** INSTRUCTIONS FAILED IN MODE 6 ***
 1461
 1462 016416  005067  162366              2$:   CLR     TEMP1           ; START CLEAN
 1463 016422  012760  052525  000000            MOV     #52525,0(R0)    ; LOAD TEMP
 1464 016430  016260  177774  000002            MOV     -4(R2),2(R0)    ; TEMP => TEMP1
 1465 016436  022767  052525  162344            CMP     #52525,TEMP1    ; CHECK IT
 1466 016444  001402                            BEQ     3$              ; OK, CONTINUE
 1467 016446  104423  016316                    ERROR,  T41             ;; *** INSTRUCTIONS FAILED IN MODE 6 ***
 1468 016452                              3$:
 1469
 1470                                    ;*******************************************************************************
                                         .SBTTL      T42 -- TEST MODE 7 USING MOVB AND MOV
                                         ;*******************************************************************************
      016452  104420  000042             T42:    BEGINTEST, 42           ;; CHECK SWITCH REGISTER OPTIONS.
 1471 016456  005067  162326             MODE7:  CLR     TEMP1           ; START CLEAN
```

T42 -- TEST MODE 7 USING MOVB AND MOV

```
1472 016462  012767  001006  162310          MOV     #TEMP,ADR       ; LOAD ADDRESSES
1473 016470  012767  001010  162304          MOV     #TEMP1,ADR1     ;
1474 016476  012767  001012  162300          MOV     #TEMP2,ADR2     ;
1475 016504  012700  001000                  MOV     #ADR,R0         ; LOAD ADDRESSES OF ADDRESSES
1476 016510  012701  001002                  MOV     #ADR1,R1        ;
1477 016514  012702  001004                  MOV     #ADR2,R2        ;
1478 016520  112770  000252  000000          MOVB    #252,@0(R0)     ; LOAD TEMP
1479 016526  117270  177774  000002          MOVB    @-4(R2),@2(R0)  ; TEMP => TEMP1
1480 016534  122767  000252  162246          CMPB    #252,TEMP1      ; CHECK IT
1481 016542  001402                          BEQ     1$              ; OK, CONTINUE
1482 016544  104423  016452                  ERROR,  T42             ;; *** MODE 7 IS FAILING ***
1483 016550  012770  125252  000000  1$:     MOV     #125252,@0(R0)  ; LOAD TEMP
1484 016556  017270  177774  000002          MOV     @-4(R2),@2(R0)  ; TEMP => TEMP1
1485 016564  022767  125252  162216          CMP     #125252,TEMP1   ; CHECK IT
1486 016572  001402                          BEQ     2$              ; OK, CONTINUE
1487 016574  104423  016452                  ERROR,  T42             ;; *** INSTRUCTIONS FAILED IN MODE 7 ***
1488
1489 016600                          2$:
1490                                 ;*****************************************************************************
                                     .SBTTL      T43 -- TSTB, CLRB, AND MOVB
                                     ;*****************************************************************************
     016600  104420  000043         T43:    BEGINTEST, 43           ;; CHECK SWITCH REGISTER OPTIONS.
1491 016604  012700  001006         TSTB1:  MOV     #TEMP,R0        ; LOAD ADDRESSES
1492 016610  012701  001010                 MOV     #TEMP1,R1       ;
1493 016614  000277                         SCC
1494 016616  105010                         CLRB    (R0)            ; CLEAR THE LOCATION
1495 016620  104404                         CHECKCC+4               ; CHECK FOR CC = 4
1496 016622  104423  016600                 ERROR,  T43             ;; *** INCORRECT CC BITS ***
1497 016626  105710                         TSTB    (R0)            ; CHECK IT
1498 016630  104404                         CHECKCC+4               ; CHECK FOR CC = 4
1499 016632  104423  016600                 ERROR,  T43             ;; *** INCORRECT CC BITS ***
1500 016636  112711  000377                 MOVB    #377,(R1)       ; LOAD THE LOCATION
1501 016642  104410                         CHECKCC+10              ; CHECK FOR CC = 10
1502 016644  104423  016600                 ERROR,  T43             ;; *** INCORRECT CC BITS ***
1503 016650  105711                         TSTB    (R1)            ; CHECK IT
1504 016652  104410                         CHECKCC+10              ; CHECK FOR CC = 10
1505 016654  104423  016600                 ERROR,  T43             ;; *** INCORRECT CC BITS ***
1506 016660  010002                         MOV     R0,R2           ; R2 IS NOW POINTING TO LOCATION TEMP
1507 016662  112762  000200  000000         MOVB    #200,0(R2)      ; PLACE #200 IN LOCATION TEMP
1508 016670  112241                         MOVB    (R2)+,-(R1)     ; MOVE #200 TO LOCATION TEMP+1
1509 016672  026127  177777  100200         CMP     -1(R1),#100200  ; CHECK THE DATA IN LOCATION TEMP
1510 016700  001402                         BEQ     4$
1511 016702  104423  016600                 ERROR,  T43             ;; *** MOVB INSTRUCTION FAILED ***
1512 016706  020102                 4$:     CMP     R1,R2           ; CHECK THE REGISTERS FOR PROPER VALUE
1513 016710  001402                         BEQ     5$
1514 016712  104423  016600                 ERROR,  T43             ;; *** MOVB INSTRUCTION FAILED ***
1515 016716                          5$:
1516
1517                                 ;*****************************************************************************
                                     .SBTTL      T44 -- CMPB AND BISB
                                     ;*****************************************************************************
     016716  104420  000044         T44:    BEGINTEST, 44           ;; CHECK SWITCH REGISTER OPTIONS.
1518 016722  012701  001012         CMPB1:  MOV     #TEMP2,R1
1519 016726  012702  001006                 MOV     #TEMP,R2        ; LOAD ADDRESS
1520 016732  012711  000077                 MOV     #77,(R1)        ; PLACE 77 IN LOCATION TEMP2
1521 016736  112704  000377                 MOVB    #377,R4         ; R4 SHOULD CONTAIN #177777
1522 016742  150412                         BISB    R4,(R2)         ; LOAD LOCATION
```

```
1523 016744  104410                        CHECKCC+10              ; CHECK FOR CC = 10
1524 016746  104423  016716                ERROR.  T44             ;; *** INCORRECT CC BITS ***
1525 016752  120412                        CMPB    R4,(R2)         ; CHECK COMPARE
1526 016754  001402                        BEQ     2$              ; CONTINUE IF OK
1527 016756  104423  016716                ERROR.  T44             ;; *** BISB OR CMPB INSTRUCTION FAILED ***
1528 016762  121112              2$:       CMPB    (R1),(R2)       ; CHECK IT AGAIN
1529 016764  100002                        BPL     3$              ; CONTINUE IF OK
1530 016766  104423  016716                ERROR.  T44             ;; *** CMPB INSTRUCTION FAILED (WRONG CC) ***
1531 016772  121211              3$:       CMPB    (R2),(R1)       ; ONCE MORE
1532 016774  100402                        BMI     4$              ; CONTINUE IF OK
1533 016776  104423  016716                ERROR.  T44             ;; *** INSTR FAILED ***
1534 017002                      4$:
1535
1536                             ;*************************************************************************
                                 .SBTTL      T45 -- BICB AND BITB
                                 ;*************************************************************************
     017002  104420  000045     T45:        BEGINTEST, 45           ;; CHECK SWITCH REGISTER OPTIONS.
1537 017006  012703  001006     BICB1:      MOV     #TEMP,R3        ; LOAD ADDRESS
1538 017012  112713  000377                 MOVB    #377,(R3)       ; LOAD LOCATION
1539 017016  012700  001010                 MOV     #TEMP1,R0       ;PLACE THE ADDRESS OF TEMP1 IN R0
1540 017022  010001                         MOV     R0,R1           ;AND R1
1541 017024  112721  000252                 MOVB    #252,(R1)+      ; PLACE #252 IN TEMP1
1542 017030  000277                         SCC
1543 017032  146013  000000                 BICB    0(R0),(R3)      ; CLEAR EVERY OTHER BIT
1544 017036  104401                         CHECKCC+1               ; CHECK FOR CC = 1
1545 017040  104423  017002                 ERROR.  T45             ;; *** INCORRECT CC BITS ***
1546 017044  136113  177777                 BITB    -1(R1),(R3)     ; CHECK IT
1547 017050  001402                         BEQ     4$              ; CONTINUE IF OK
1548 017052  104423  017002                 ERROR.  T45             ;; *** BICB OR BITB INSTRUCTION FAILED ***
1549 017056  132713  000125     4$:         BITB    #125,(R3)       ; CHECK IT
1550 017062  104401                         CHECKCC+1               ; CHECK FOR CC = 1
1551 017064  104423  017002                 ERROR.  T45             ;; *** INCORRECT CC BITS ***
1552 017070  154113                         BISB    -(R1),(R3)      ; SET THE BITS THAT WERE CLEARED
1553 017072  100402                         BMI     6$              ; CONTINUE IF OK
1554 017074  104423  017002                 ERROR.  T45             ;; *** BITB OR BISB INSTRUCTION FAILED ***
1555 017100  012746  000177     6$:         MOV     #177,-(SP)      ; STORE #177 ON THE STACK
1556 017104  142613                         BICB    (SP)+,(R3)      ; CLEAR ALL THE BITS EXCEPT SIGN BIT
1557 017106  104411                         CHECKCC+11              ; CHECK FOR CC = 11
1558 017110  104423  017002                 ERROR.  T45             ;; *** INCORRECT CC BITS ***
1559 017114  132713  000377                 BITB    #377,(R3)       ; CHECK IT
1560 017120  104411                         CHECKCC+11              ; CHECK FOR CC = 11
1561 017122  104423  017002                 ERROR.  T45             ;; *** INCORRECT CC BITS ***
1562 017126  010300                         MOV     R3,R0           ; PLACE THE ADDRESS OF TEMP IN R0
1563 017130  012710  001010                 MOV     #TEMP1,(R0)     ; PLACE THE ADDRESS OF TEMP1 IN TEMP
1564 017134  012730  000377                 MOV     #377,@(R0)+     ; WRITE A 377 IN LOCATION TEMP1
1565 017140  000263                         SEVC                    ; SET V & C BITS
1566 017142  145070  000000                 BICB    @-(R0),@(R0)    ; BIT CLEAR THE CONTENTS
1567                                                                 ; OF TEMP1 TO THE CONTENTS OF TEMP1
1568 017146  104405                         CHECKCC+5               ; CHECK FOR CC = 5
1569 017150  104423  017002                 ERROR.  T45             ;; *** INCORRECT CC BITS ***
1570 017154  022027  001010                 CMP     (R0)+,#TEMP1    ; MAKE SURE THAT (R0) IS POINTING TO LOCATION TEMP1
1571 017160  001402                         BEQ     8$
1572 017162  104423  017002                 ERROR.  T45             ;; *** BICB OR CMP INSTRUCTION FAILED IN THE SPECIFIC MODE *
**
1573 017166  005750              8$:        TST     @-(R0)          ; TEST LOCATION TEMP1
1574 017170  001402                         BEQ     10$
1575 017172  104423  017002                 ERROR.  T45             ;; *** BICB INSTRUCTION FAILED ***
1576 017176  000257              10$:       CCC
```

```
PC 17172 = BICB INSTRUCTION FAILED.

 1577 017200  141010                       BICB    (R0),(R0)       ; CLEAR THE LOCATION TEMP
 1578 017202  104404                       CHECKCC+4               ; CHECK FOR CC = 4
 1579 017204  104423  017002               ERROR,  T45             ;; *** INCORRECT CC BITS ***
 1580
 1581                                ;*************************************************************************
                                     .SBTTL      T46 -- INCB AND DECB
                                     ;*************************************************************************
      017210  104420  000046        T46:    BEGINTEST, 46           ;; CHECK SWITCH REGISTER OPTIONS.
 1582 017214  012704  001006        INCB1:  MOV     #TEMP,R4        ; LOAD ADDRESS
 1583 017220  112714  000177               MOVB    #177,(R4)       ; TEMP LOCATION=177
 1584 017224  000261                       SEC
 1585 017226  105214                       INCB    (R4)            ; ADD ONES INTO LOCATION
 1586 017230  104413                       CHECKCC+13              ; CHECK FOR CC = 13
 1587 017232  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1588 017236  012714  000376               MOV     #376,(R4)
 1589 017242  105224                       INCB    (R4)+           ;
 1590 017244  104411                       CHECKCC+11              ; CHECK FOR CC = 11
 1591 017246  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1592 017252  105744                       TSTB    -(R4)           ; DECREMENT R4 BY 1
 1593 017254  005746                       TST     -(SP)           ; AND SP BY 2
 1594 017256  010426                       MOV     R4,(SP)+        ; PLACE THE ADDRESS OF TEMP ON THE STACK
 1595 017260  000241                       CLC                     ; CLEAR C BIT
 1596 017262  105256                       INCB    @-(SP)          ; INCREMENT THE CONTENTS OF LOCATION TEMP
 1597 017264  104404                       CHECKCC+4               ; CHECK FOR CC = 4
 1598 017266  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1599 017272  123634                       CMPB    @(SP)+,@(R4)+   ; RESTORE STACK POINTER
 1600 017274  000261                       SEC                     ; SET C BIT
 1601 017276  105264  177777               INCB    -1(R4)
 1602 017302  104401                       CHECKCC+1               ; CHECK FOR CC = 1
 1603 017304  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1604 017310  124427  000001               CMPB    -(R4),#1        ; CHECK IT
 1605 017314  001402                       BEQ     2$              ; CONTINUE IF OK
 1606 017316  104423  017210               ERROR,  T46             ;; *** INCB INSTRUCTION FAILED ***
 1607 017322  000261                2$:    SEC
 1608 017324  105314                       DECB    (R4)            ; SUBTRACT ONES FROM LOCATION
 1609 017326  104405                       CHECKCC+5               ; CHECK FOR CC = 5
 1610 017330  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1611 017334  105324                       DECB    (R4)+
 1612 017336  104411                       CHECKCC+11              ; CHECK FOR CC = 11
 1613 017340  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1614 017344  112764  000200  177777       MOVB    #200,-1(R4)
 1615 017352  105344                       DECB    -(R4)           ;
 1616 017354  104403                       CHECKCC+3               ; CHECK CC = 3
 1617 017356  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1618 017362  105364  000000               DECB    0(R4)           ;
 1619 017366  104401                       CHECKCC+1               ; CHECK FOR CC = 1
 1620 017370  104423  017210               ERROR,  T46             ;; *** INCORRECT CC BITS ***
 1621 017374  126427  000000  000176       CMPB    0(R4),#176
 1622 017402  001402                       BEQ     3$
 1623 017404  104423  017210               ERROR,  T46             ;; *** DECB INSTRUCTION FAILED ***
 1624 017410                        3$:
 1625
 1626                                ;*************************************************************************
                                     .SBTTL      T47 -- COMB AND NEGB
                                     ;*************************************************************************
      017410  104420  000047        T47:    BEGINTEST, 47           ;; CHECK SWITCH REGISTER OPTIONS.
 1627 017414  012703  001006        COMB1:  MOV     #TEMP,R3        ; LOAD ADDRESS
```

```
1628 017420  012704  001010                MOV     #TEMP1,R4
1629 017424  012714  000252                MOV     #252,(R4)
1630 017430  112413                        MOVB    (R4)+,(R3)      ; LOAD EVERY OTHER BIT
1631 017432  000277                        SCC
1632 017434  105113                        COMB    (R3)            ; 1'S COMPLEMENT
1633 017436  104401                        CHECKCC+1               ; CHECK FOR CC = 1
1634 017440  104423  017410                ERROR,  T47             ;; *** INCORRECT CC BITS ***
1635 017444  122713  000125                CMPB    #125,(R3)       ; CHECK IT
1636 017450  001402                        BEQ     2$              ; CONTINUE IF OK
1637 017452  104423  017410                ERROR,  T47             ;; *** COMB INSTRUCTION FAILED ***
1638 017456  000277                2$:     SCC
1639 017460  105113                        COMB    (R3)            ; COMPLEMENT BACK
1640 017462  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1641 017464  104423  017410                ERROR,  T47             ;; *** INCORRECT CC BITS ***
1642 017470  010400                        MOV     R4,R0
1643 017472  126013  177777                CMPB    -1(R0),(R3)     ; CHECK IT
1644 017476  001402                        BEQ     3$              ; CONTINUE IF OK
1645 017500  104423  017410                ERROR,  T47             ;; *** COMB INSTRUCTION FAILED ***
1646 017504  112724  000377          3$:   MOVB    #377,(R4)+
1647 017510  114413                        MOVB    -(R4),(R3)      ; PLACE #377 IN (R3)
1648 017512  000277                        SCC
1649 017514  105113                        COMB    (R3)
1650 017516  104405                        CHECKCC+5               ; CHECK FOR CC = 5
1651 017520  104423  017410                ERROR,  T47             ;; *** INCORRECT CC BITS ***
1652
1653 017524  012700  001006          NEGB1:MOV     #TEMP,R0        ; LOAD ADDRESS
1654 017530  112710  000001                MOVB    #1,(R0)         ; LOAD THE LOCATION
1655 017534  105410                        NEGB    (R0)            ; 2'S COMPLEMENT
1656 017536  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1657 017540  104423  017524                ERROR,  NEGB1           ;; *** INCORRECT CC BITS ***
1658 017544  122710  000377                CMPB    #377,(R0)       ; CHECK IT
1659 017550  001402                        BEQ     2$              ; CONTINUE IF OK
1660 017552  104423  017524                ERROR,  NEGB1           ;; *** NEGB INSTRUCTION FAILED ***
1661 017556  012710  000200          2$:   MOV     #200,(R0)
1662 017562  105410                        NEGB    (R0)            ; 2'S COMPLEMENT
1663 017564  104413                        CHECKCC+13              ; CHECK FOR CC = 13
1664 017566  104423  017524                ERROR,  NEGB1           ;; *** INCORRECT CC BITS ***
1665 017572  122710  000200                CMPB    #200,(R0)       ; CHECK IT
1666 017576  001402                        BEQ     3$              ; CONTINUE IF OK
1667 017600  104423  017524                ERROR,  NEGB1           ;; *** NEGB FAILED. ***
1668 017604                          3$:
1669
1670                                ;*******************************************************************************
                                    .SBTTL      T50 -- ROLB AND RORB
                                    ;*******************************************************************************
     017604  104420  000050        T50:    BEGINTEST, 50           ;; CHECK SWITCH REGISTER OPTIONS.
1671 017610  012701  001010        ROLB1:  MOV     #TEMP1,R1       ; LOAD ADDRESS
1672 017614  112711  000040                MOVB    #40,(R1)        ; LOAD LOCATION
1673 017620  000257                        CCC                     ; CLEAR FLAGS
1674 017622  106111                        ROLB    (R1)            ; SHIFT
1675 017624  106111                        ROLB    (R1)            ;
1676 017626  104412                        CHECKCC+12              ; CHECK FOR CC = 12
1677 017630  104423  017604                ERROR,  T50             ;; *** INCORRECT CC BITS ***
1678 017634  122711  000200                CMPB    #200,(R1)       ; CHECK IT
1679 017640  001402                        BEQ     1$              ; CONTINUE IF OK
1680 017642  104423  017604                ERROR,  T50             ;; *** ROLB INSTRUCTION FAILED ***
1681 017646  106111                1$:     ROLB    (R1)            ; SHIFT
```

PC 17642 = ROLB INSTRUCTION FAILED.

```
 1682 017650  104407                    CHECKCC+7               ; CHECK FOR CC = 7
 1683 017652  104423  017604            ERROR,  T50             ;; *** INCORRECT CC BITS ***
 1684 017656  106111                    ROLB    (R1)            ; SHIFT
 1685 017660  122711  000001            CMPB    #1,(R1)         ; CHECK IT
 1686 017664  001402                    BEQ     RORB1           ; CONTINUE IF OK
 1687 017666  104423  017604            ERROR,  T50             ;; *** ROLB FAILED ***
 1688
 1689 017672  012702  001010    RORB1:  MOV     #TEMP1,R2       ; LOAD ADDRESS
 1690 017676  112712  000004            MOVB    #4,(R2)         ; LOAD LOCATION
 1691 017702  000257                    CCC                     ; CLEAR FLAGS
 1692 017704  106012                    RORB    (R2)            ; SHIFT
 1693 017706  106012                    RORB    (R2)            ;
 1694 017710  122712  000001            CMPB    #1,(R2)         ; CHECK IT
 1695 017714  001402                    BEQ     1$              ; CONTINUE IF OK
 1696 017716  104423  017672            ERROR,  RORB1           ;; *** RORB INSTRUCTION FAILED ***
 1697 017722  106012            1$:     RORB    (R2)            ; SHIFT
 1698 017724  104407                    CHECKCC+7               ; CHECK FOR CC = 7
 1699 017726  104423  017672            ERROR,  RORB1           ;; *** INCORRECT CC BITS ***
 1700 017732  106012                    RORB    (R2)            ; SHIFT
 1701 017734  104412                    CHECKCC+12              ; CHECK FOR CC = 12
 1702 017736  104423  017672            ERROR,  RORB1           ;; *** INCORRECT CC BITS ***
 1703 017742  122712  000200            CMPB    #200,(R2)       ; CHECK IT
 1704 017746  001402                    BEQ     2$              ; CONTINUE IF OK
 1705 017750  104423  017672            ERROR,  RORB1           ;; *** RORB FAILED ***
 1706 017754                    2$:
 1707
 1708                           ;*************************************************************************
                                .SBTTL      T51 -- ASLB AND ASRB
                                ;*************************************************************************
      017754  104420  000051   T51:    BEGINTEST, 51           ;; CHECK SWITCH REGISTER OPTIONS.
 1709 017760  012703  001010   ASLB1:  MOV     #TEMP1,R3       ; LOAD ADDRESS
 1710 017764  112713  000040            MOVB    #40,(R3)        ; LOAD LOCATION
 1711 017770  000257                    CCC                     ; CLEAR FLAGS
 1712 017772  106313                    ASLB    (R3)            ; SHIFT
 1713 017774  106313                    ASLB    (R3)            ;
 1714 017776  104412                    CHECKCC+12              ; CHECK FOR CC = 12
 1715 020000  104423  017754            ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1716 020004  122713  000200            CMPB    #200,(R3)       ; CHECK IT
 1717 020010  001402                    BEQ     4$              ; CONTINUE IF OK
 1718 020012  104423  017754            ERROR,  T51             ;; *** ASLB INSTRUCTION FAILED ***
 1719 020016  106313            4$:     ASLB    (R3)            ; SHIFT
 1720 020020  104407                    CHECKCC+7               ; CHECK FOR CC = 7
 1721 020022  104423  017754            ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1722 020026  106313                    ASLB    (R3)            ; SHIFT
 1723 020030  104404                    CHECKCC+4               ; CHECK FOR CC = 4
 1724 020032  104423  017754            ERROR,  T51             ;; *** INCORRECT CC BITS ***
 1725
 1726 020036  012704  001010   ASRB1:  MOV     #TEMP1,R4       ; LOAD ADDRESSES
 1727 020042  012703  001012            MOV     #TEMP2,R3       ;
 1728 020046  112714  000004            MOVB    #4,(R4)         ; LOAD LOCATION
 1729 020052  000257                    CCC                     ; CLEAR FLAGS
 1730 020054  106214                    ASRB    (R4)            ; SHIFT
 1731 020056  106214                    ASRB    (R4)            ;
 1732 020060  122714  000001            CMPB    #1,(R4)         ; CHECK IT
 1733 020064  001402                    BEQ     2$              ; CONTINUE IF OK
 1734 020066  104423  020036            ERROR,  ASRB1           ;; *** ASRB INSTRUCTION FAILED ***
 1735 020072  106214            2$:     ASRB    (R4)            ; SHIFT
```

PC 20066 = ASRB INSTRUCTION FAILED.

```
1736 020074  104407                        CHECKCC+7               ; CHECK FOR CC = 7
1737 020076  104423  020036                ERROR.  ASRB1           ;; *** INCORRECT CC BITS ***
1738 020102  106214                        ASRB    (R4)            ; SHIFT
1739 020104  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1740 020106  104423  020036                ERROR.  ASRB1           ;; *** INCORRECT CC BITS ***
1741 020112  112713  000202                MOVB    #202,(R3)       ; LOAD LOCATION
1742 020116  106213                        ASRB    (R3)            ; SHIFT
1743 020120  106213                        ASRB    (R3)            ;
1744 020122  104411                        CHECKCC+11              ; CHECK FOR CC = 11
1745 020124  104423  020036                ERROR.  ASRB1           ;; *** INCORRECT CC BITS ***
1746 020130  122713  000340                CMPB    #340,(R3)       ; CHECK IT
1747 020134  001402                        BEQ     3$              ; CONTINUE IF OK
1748 020136  104423  020036                ERROR.  ASRB1           ;; *** ASRB FAILED ***
1749 020142                          3$:
1750
1751                                 ;*******************************************************************************
                                     .SBTTL      TS2 -- ADCB AND SBCB
                                     ;*******************************************************************************
     020142  104420  000052         TS2:    BEGINTEST, 52           ;; CHECK SWITCH REGISTER OPTIONS.
1752 020146  012700  001012         ADCB1:  MOV     #TEMP2,R0       ; LOAD ADDRESS
1753 020152  105010                        CLRB    (R0)            ; CLEAR THE LOCATION
1754 020154  000257                        CCC                     ; CLEAR FLAGS
1755 020156  105510                        ADCB    (R0)            ; ADD C BIT = 0
1756 020160  104404                        CHECKCC+4               ; CHECK FOR CC = 4
1757 020162  104423  020142                ERROR.  TS2             ;; *** INCORRECT CC BITS ***
1758 020166  000261                        SEC                     ; C=1
1759 020170  105510                        ADCB    (R0)            ; ADD C BIT=1
1760 020172  000261                        SEC                     ; C=1
1761 020174  105510                        ADCB    (R0)            ; AGAIN
1762 020176  104400                        CHECKCC+0               ; CHECK FOR CC = 0
1763 020200  104423  020142                ERROR.  TS2             ;; *** INCORRECT CC BITS ***
1764 020204  122710  000002                CMPB    #2,(R0)         ; CHECK IT
1765 020210  001402                        BEQ     4$              ; CONTINUE IF OK
1766 020212  104423  020142                ERROR.  TS2             ;; *** ADCB INSTRUCTION FAILED ***
1767 020216  112710  000177          4$:   MOVB    #177,(R0)       ; LOAD LARGEST POSITVE BYTE
1768 020222  000261                        SEC                     ; C=1
1769 020224  105510                        ADCB    (R0)            ; ADD C BIT=1
1770 020226  104412                        CHECKCC+12              ; CHECK FOR CC = 12
1771 020230  104423  020142                ERROR.  TS2             ;; *** INCORRECT CC BITS ***
1772 020234  122710  000200                CMPB    #200,(R0)       ; CHECK IT
1773 020240  001402                        BEQ     6$              ; CONTINUE IF OK
1774 020242  104423  020142                ERROR.  TS2             ;; *** ADCB INSTRUCTION FAILED ***
1775 020246  112710  000377          6$:   MOVB    #377,(R0)       ; LOAD -1
1776 020252  000261                        SEC                     ; C=1
1777 020254  105510                        ADCB    (R0)            ; ADD C BIT=1
1778 020256  104405                        CHECKCC+5               ; CHECK FOR CC = 5
1779 020260  104423  020142                ERROR.  TS2             ;; *** INCORRECT CC BITS ***
1780
1781 020264  012701  001012         SBCB1:  MOV     #TEMP2,R1       ; LOAD ADDRESS
1782 020270  112711  000003                MOVB    #3,(R1)         ; LOAD LOCATION
1783 020274  000257                        CCC                     ; CLEAR FLAGS
1784 020276  105611                        SBCB    (R1)            ; SUBTRACT C BIT=0
1785 020300  104400                        CHECKCC+0               ; CHECK FOR CC = 0
1786 020302  104423  020264                ERROR.  SBCB1           ;; *** INCORRECT CC BITS ***
1787 020306  122711  000003                CMPB    #3,(R1)         ; CHECK IT
1788 020312  001402                        BEQ     2$              ; CONTINUE IF OK
1789 020314  104423  020264                ERROR.  SBCB1           ;; *** SBCB INSTRUCTION FAILED ***
```

PC 20314 = SBCB INSTRUCTION FAILED.

```
 1790 020320  000261                2$:     SEC                     ; C=1
 1791 020322  105611                        SBCB    (R1)            ; SUBTRACT C BIT=1
 1792 020324  000261                        SEC                     ; C=1
 1793 020326  105611                        SBCB    (R1)            ;
 1794 020330  104400                        CHECKCC+0               ; CHECK FOR CC = 0
 1795 020332  104423  020264                ERROR,  SBCB1           ;; *** INCORRECT CC BITS ***
 1796 020336  122711  000001                CMPB    #1,(R1)         ; CHECK IT
 1797 020342  001402                        BEQ     3$              ; CONTINUE IF OK
 1798 020344  104423  020264                ERROR,  SBCB1           ;; *** SBCB INSTRUCTION FAILED ***
 1799 020350  000261                3$:     SEC                     ; C=1
 1800 020352  105611                        SBCB    (R1)            ; SUBTRACT C BIT=1
 1801 020354  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 1802 020356  104423  020264                ERROR,  SBCB1           ;; *** INCORRECT CC BITS ***
 1803 020362  000261                        SEC                     ; C=1
 1804 020364  105611                        SBCB    (R1)            ; SUBTRACT C BIT = 1
 1805 020366  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 1806 020370  104423  020264                ERROR,  SBCB1           ;; *** INCORRECT CC BITS ***
 1807 020374  122711  000377                CMPB    #377,(R1)       ; CHECK IT
 1808 020400  001402                        BEQ     4$              ; CONTINUE IF OK
 1809 020402  104423  020264                ERROR,  SBCB1           ;; *** SBCB INSTRUCTION FAILED ***
 1810 020406  112711  000200        4$:     MOVB    #200,(R1)       ; LOAD R1
 1811 020412  000261                        SEC                     ; C=1
 1812 020414  105611                        SBCB    (R1)            ; SUBTRACT C BIT = 1
 1813 020416  104402                        CHECKCC+2               ; CHECK FOR CC = 2
 1814 020420  104423  020264                ERROR,  SBCB1           ;; *** INCORRECT CC BITS ***
 1815
 1816                                ;*************************************************************************
                                     .SBTTL      TS3 -- TST, CLR, AND MOV
                                     ;*************************************************************************
      020424  104420  000053        TS3:    BEGINTEST, 53           ;; CHECK SWITCH REGISTER OPTIONS.
 1817 020430  012701  001006        MOV1:   MOV     #TEMP,R1        ; LOAD ADDRESSES
 1818 020434  012700  001010                MOV     #TEMP1,R0       ;
 1819 020440  000277                        SCC
 1820 020442  005010                        CLR     (R0)            ; CLEAR THE LOCATION
 1821 020444  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 1822 020446  104423  020424                ERROR,  TS3             ;; *** INCORRECT CC BITS ***
 1823 020452  005720                        TST     (R0)+           ; CHECK IT
 1824 020454  104404                        CHECKCC+4               ; CHECK FOR CC = 4
 1825 020456  104423  020424                ERROR,  TS3             ;; *** INCORRECT CC BITS ***
Z 1826 020462  010040                      MOV     R0,-(R0)        ; POSSIBLE "Z" ASSEMBLY ERROR O.K.      ;JSD REV B
 1827 020464  012730  177777                MOV     #177777,@(R0)+
 1828 020470  016011  177776                MOV     -2(R0),(R1)     ; LOAD THE LOCATION.
 1829 020474  104410                        CHECKCC+10              ; CHECK FOR CC = 10
 1830 020476  104423  020424                ERROR,  TS3             ;; *** INCORRECT CC BITS ***
 1831 020502  005711                        TST     (R1)            ; CHECK IT
 1832 020504  104410                        CHECKCC+10              ; CHECK FOR CC = 10
 1833 020506  104423  020424                ERROR,  TS3             ;; *** INCORRECT CC BITS ***
 1834
 1835                                ;*************************************************************************
                                     .SBTTL      TS4 -- CMP AND BIS
                                     ;*************************************************************************
      020512  104420  000054        TS4:    BEGINTEST, 54           ;; CHECK SWITCH REGISTER OPTIONS.
 1836 020516  012702  001010        CMP1:   MOV     #TEMP1,R2       ; LOAD ADDRESS
 1837 020522  012700  001006                MOV     #TEMP,R0        ; PLACE THE ADDRESS OF TEMP IN R0
 1838 020526  000257                        CCC                     ; CLEAR CC BITS.
 1839 020530  012720  177777                MOV     #177777,(R0)+   ; PLACE #177777 IN TEMP AND INC R0 BY 2
 1840 020534  054012                        BIS     -(R0),(R2)      ; LOAD LOCATION
```

```
 1841 020536  104410                          CHECKCC,10              ; CHECK FOR CC = 10
 1842 020540  104423  020512                  ERROR.  TS4             ;; *** INCORRECT CC BITS ***
 1843 020544  022227  177777                  CMP     (R2)+,#177777   ; CHECK COMPARE
 1844 020550  001402                          BEQ     2$              ; CONTINUE IF OK
 1845 020552  104423  020512                  ERROR.  TS4             ;; *** CMP OR BIS INSTRUCTION FAILED ***
 1846 020556  020227  001012          2$:     CMP     R2,#TEMP1+2     ; CHECK R2 TO CONTAIN ADDRESS OF TEMP1+2
 1847 020562  001402                          BEQ     3$
 1848 020564  104423  020512                  ERROR.  TS4             ;; *** NO AUTO INCREMENT ***
 1849 020570  022742  000077          3$:     CMP     #77,-(R2)       ; CHECK IT AGAIN
 1850 020574  104401                          CHECKCC,1               ; CHECK FOR CC = 1
 1851 020576  104423  020512                  ERROR.  TS4             ;; *** INCORRECT CC BITS ***
 1852 020602  022722  077777                  CMP     #77777,(R2)+
 1853 020606  104413                          CHECKCC,13              ; CHECK FOR CC = 13
 1854 020610  104423  020512                  ERROR.  TS4             ;; *** INCORRECT CC BITS ***
 1855 020614  024227  077777                  CMP     -(R2),#77777    ; ONCE MORE
 1856 020620  104410                          CHECKCC,10              ; CHECK FOR CC = 10
 1857 020622  104423  020512                  ERROR.  TS4             ;; *** INCORRECT CC BITS ***
 1858 020626  012767  052525  160156          MOV     #52525,TEMP2    ; SET EVERY OTHER BIT IN TEMP2
 1859 020634  012767  001012  160146          MOV     #TEMP2,TEMP1    ; PLACE THE ADDRESS OF TEMP2 IN TEMP1
 1860 020642  012704  001000                  MOV     #ADR,R4
 1861 020646  012714  001002                  MOV     #ADR1,(R4)      ; PLACE THE ADDRESS OF ADR1 IN ADR POINTED BY R4
 1862 020652  012734  125252                  MOV     #125252,@(R4)+  ; PLACE THE #125252 IN LOCATION ADR1
 1863 020656  057432  177776                  BIS     @-2(R4),@(R2)+  ; SET EVERY OTHER BIT AT LOCATION TEMP2
 1864                                                                 ; AND INCREMENT R2 BY 2
 1865 020662  010200                          MOV     R2,R0           ; PLACE ADDRESS OF TEMP2 IN R0
 1866 020664  025027  177777                  CMP     @-(R0),#177777  ; TEMP2 SHOULD CONTAIN ALL 1'S
 1867 020670  001402                          BEQ     4$
 1868 020672  104423  020512                  ERROR.  TS4             ;; *** CMP OR BIS INSTUCTIONS FAILED IN MODES OTHER THAN 0 *
**
 1869 020676  020227  001012          4$:     CMP     R2,#TEMP1+2     ; R2 SHOULD CONTAIN THE ADDRESS OF TEMP2
 1870 020702  001402                          BEQ     5$
 1871 020704  104423  020512                  ERROR.  TS4             ;; *** MODE 5 IS FAILING ***
 1872 020710  005040                  5$:     CLR     -(R0)           ; PLACE A 0 IN TEMP
 1873 020712  010067  160074                  MOV     R0,TEMP2        ; PLACE ADDRESS OF TEMP IN TEMP2
 1874 020716  022020                          CMP     (R0)+,(R0)+     ; BUMP R0 BY 4
 1875 020720  055070  000002                  BIS     @-(R0),@2(R0)   ; PLACE THE CONTENTS OF TEMP2 AT TEMP
 1876 020724  022767  001006  160054          CMP     #TEMP,TEMP      ; TEMP SHOULD CONTAIN ITS OWN ADDRESS
 1877 020732  001402                          BEQ     6$
 1878 020734  104423  020512                  ERROR.  TS4             ;; *** CMP OR BIS INSTRUCTIONS FAILED ***
 1879 020740                          6$:
 1880
 1881                                 ;*******************************************************************************
                                      .SBTTL      TS5 -- BIC AND BIT
                                      ;*******************************************************************************
      020740  104420  000055          TS5:    BEGINTEST, 55           ;; CHECK SWITCH REGISTER OPTIONS.
 1882 020744  012703  001006          BIC1:   MOV     #TEMP,R3        ; LOAD ADDRESS
 1883 020750  012713  177777                  MOV     #177777,(R3)    ; LOAD LOCATION
 1884 020754  012704  001000                  MOV     #ADR,R4         ; PLACE THE ADDRESS OF ADR IN R4
 1885 020760  012714  001002                  MOV     #ADR1,(R4)      ; PLACE THE ADDRESS OF ADR1 IN ADR
 1886 020764  011334                          MOV     (R3),@(R4)+     ; LOAD LOCATION ADR1 WITH #177777
 1887 020766  012700  001010                  MOV     #TEMP1,R0       ; PLACE THE ADDRESS OF TEMP1 IN R0
 1888 020772  012710  125252                  MOV     #125252,(R0)    ; SET EVERY OTHER BIT AT LOCATION TEMP1
 1889 020776  000277                          SCC
 1890 021000  042013                          BIC     (R0)+,(R3)      ; CLEAR EVERY OTHER BIT
 1891 021002  104401                          CHECKCC,1               ; CHECK FOR CC = 1
 1892 021004  104423  020740                  ERROR.  TS5             ;; *** INCORRECT CC BITS ***
 1893 021010  034013                          BIT     -(R0),(R3)      ; CHECK IT
 1894 021012  001402                          BEQ     1$              ; CONTINUE IF OK
```

PC 21014 = BIC OR BIT INSTRUCTION FAILED.

```
1895 021014  104423  020740                ERROR,  T55             ;; *** BIC OR BIT INSTRUCTION FAILED ***
1896 021020  032713  052525          1$:   BIT     #52525,(R3)     ; CHECK IT
1897 021024  104401                       CHECKCC+1               ; CHECK FOR CC = 1
1898 021026  104423  020740                ERROR,  T55             ;; *** INCORRECT CC BITS ***
1899 021032  056013  000000                BIS     0(R0),(R3)      ; SET THE BITS THAT WERE CLEARED
1900 021036  100402                       BMI     2$              ; CONTINUE IF OK
1901 021040  104423  020740                ERROR,  T55             ;; *** BIT OR BIS INSTRUCTION FAILED ***
1902 021044  012720  077777          2$:   MOV     #77777,(R0)+    ; SET ALL THE BITS AT TEMP1 EXCEPT SIGN
1903 021050  010002                       MOV     R0,R2
1904 021052  046213  177776                BIC     -2(R2),(R3)     ; TRY CLEARING THE OTHER BITS
1905 021056  104411                       CHECKCC+11              ; CHECK FOR CC = 11
1906 021060  104423  020740                ERROR,  T55             ;; *** INCORRECT CC BITS ***
1907 021064  020027  001012                CMP     R0,#TEMP1+2     ; R0 SHOULD CONTAIN THE ADDRESS OF TEMP1+2
1908 021070  001402                       BEQ     3$
1909 021072  104423  020740                ERROR,  T55             ;; *** R0 WRONG ***
1910 021076  010010          3$:   MOV     R0,(R0)         ; PLACE ADDRESS OF TEMP2 IN TEMP2.
1911 021100  000263                       SEVC                    ; SET V AND C BITS.
1912 021102  044000                       BIC     -(R0),R0        ; SHOULD CLEAR R0.
1913 021104  104405                       CHECKCC+5               ; CHECK FOR CC = 5
1914 021106  104423  020740                ERROR,  T55             ;; *** INCORRECT CC BITS ***
1915 021112  037413  177776                BIT     @-2(R4),(R3)    ; CHECK IT
1916 021116  104411                       CHECKCC+11              ; CHECK FOR CC = 11
1917 021120  104423  020740                ERROR,  T55             ;; *** INCORRECT CC BITS ***
1918 021124  012746  125252                MOV     #125252,-(SP)   ; SET EVERY OTHER BIT ON THE STACK
1919 021130  017423  177776                MOV     @-2(R4),(R3)+   ; SET ALL THE BITS AT LOCATION TEMP
1920 021134  046643  000000                BIC     0(SP),-(R3)     ; CLEAR EVERY OTHER BIT AT LOCATION TEMP
1921 021140  022327  052525                CMP     (R3)+,#52525    ; TEMP SHOULD CONTAIN # 52525
1922 021144  001402                       BEQ     4$
1923 021146  104423  020740                ERROR,  T55             ;; *** BIC FAILED IN MODE 6 ***
1924 021152  012700  001014          4$:   MOV     #TEMP2+2,R0     ; PLACE THE ADDRESS OF TEMP2+2 IN R0
1925 021156  010340                       MOV     R3,-(R0)        ; PLACE THE ADDRESS OF TEMP1 IN TEMP2
1926 021160  014330                       MOV     -(R3),@(R0)+    ; MOVE # 52525 IN LOCATION TEMP1
1927 021162  000263                       SEVC                    ; SET V & C BITS
1928 021164  033026                       BIT     @-(R0),(SP)+    ; BIT TEST TEMP1 WITH STACK AND RESTORE STACK POINTER
1929 021166  104405                       CHECKCC+5               ; CHECK FOR CC = 5
1930 021170  104423  020740                ERROR,  T55             ;; *** INCORRECT CC BITS ***
1931 021174  020627  003776                CMP     SP,#STACK       ; MAKE SURE THAT THE SP IS OK
1932 021200  001402                       BEQ     5$
1933 021202  104423  020740                ERROR,  T55             ;; *** STACK POINTER FOULED UP ***
1934 021206                          5$:
1935
1936                                ;*************************************************************************
                                    .SBTTL      T56 -- INC AND DEC
                                    ;*************************************************************************
     021206  104420  000056        T56:    BEGINTEST, 56           ;; CHECK SWITCH REGISTER OPTIONS.
1937 021212  012704  001010        INC1:   MOV     #TEMP1,R4       ; LOAD ADDRESS
1938 021216  012714  077777                MOV     #77777,(R4)     ; TEMP1 = 77777
1939 021222  000261                       SEC
1940 021224  005214                       INC     (R4)            ; ADD ONES INTO LOCATION
1941 021226  104413                       CHECKCC+13              ; CHECK FOR CC = 13
1942 021230  104423  021206                ERROR,  T56             ;; *** INCORRECT CC BITS ***
1943 021234  012714  177776                MOV     #177776,(R4)
1944 021240  012700  001006                MOV     #TEMP,R0        ; R0 IS POINTING TO LOCATION TEMP
1945 021244  005214                       INC     (R4)            ;
1946 021246  104411                       CHECKCC+11              ; CHECK CC = 11.
1947 021250  104423  021206                ERROR,  T56             ;; *** INCORRECT CC BITS ***
1948 021254  005214                       INC     (R4)            ;
```

PC 21250 = INCORRECT CC BITS.

```
 1949 021256  104405                        CHECKCC+5               ; CHECK FOR CC = 5
 1950 021260  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1951 021264  005214                        INC     (R4)
 1952 021266  104401                        CHECKCC+1               ; CHECK FOR CC = 1
 1953 021270  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1954 021274  026427  000000  000001        CMP     0(R4),#1        ; CHECK IT
 1955 021302  001402                        BEQ     4$              ; CONTINUE IF OK
 1956 021304  104423  021206                ERROR.  T56             ;; *** INC INSTRUCTION FAILED ***
 1957 021310  000261                4$:     SEC
 1958 021312  005314                        DEC     (R4)            ; SUBTRACT ONES FROM LOCATION
 1959 021314  104405                        CHECKCC+5               ; CHECK FOR CC = 5
 1960 021316  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1961 021322  005314                        DEC     (R4)
 1962 021324  104411                        CHECKCC+11              ; CHECK CC = 11.
 1963 021326  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1964 021332  012714  100000                MOV     #100000,(R4)
 1965 021336  005314                        DEC     (R4)            ;
 1966 021340  104403                        CHECKCC+3               ; CHECK FOR CC = 3
 1967 021342  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1968 021346  005314                        DEC     (R4)            ;
 1969 021350  104401                        CHECKCC+1               ; CHECK FOR CC = 1
 1970 021352  104423  021206                ERROR.  T56             ;; *** INCORRECT CC BITS ***
 1971
 1972                               ;*************************************************************************
                                    .SBTTL      T57 -- COM AND NEG
                                    ;*************************************************************************
      021356  104420  000057        T57:    BEGINTEST, 57           ;; CHECK SWITCH REGISTER OPTIONS.
 1973 021362  012703  001010        COM1:   MOV     #TEMP1,R3       ; LOAD ADDRESS
 1974 021366  012713  125252                MOV     #125252,(R3)    ; LOAD EVERY OTHER BIT
 1975 021372  000277                        SCC
 1976 021374  005163  000000                COM     0(R3)           ; 1'S COMPLEMENT
 1977 021400  104401                        CHECKCC+1               ; CHECK FOR CC = 1
 1978 021402  104423  021356                ERROR.  T57             ;; *** INCORRECT CC BITS ***
 1979 021406  022713  052525                CMP     #52525,(R3)     ; CHECK IT
 1980 021412  001402                        BEQ     2$              ; CONTINUE IF OK
 1981 021414  104423  021356                ERROR.  T57             ;; *** COM INSTRUCTION FAILED ***
 1982 021420  000277                2$:     SCC
 1983 021422  005123                        COM     (R3)+           ; COMPLEMENT BACK
 1984 021424  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 1985 021426  104423  021356                ERROR.  T57             ;; *** INCORRECT CC BITS ***
 1986 021432  022743  125252                CMP     #125252,-(R3)   ; CHECK IT
 1987 021436  001402                        BEQ     3$              ; CONTINUE IF OK
 1988 021440  104423  021356                ERROR.  T57             ;; *** COM INSTRUCTION FAILED ***
 1989 021444  010300                3$:     MOV     R3,R0           ; R0 IS NOW POINTING TO LOCATION TEMP1
 1990 021446  012710  177777                MOV     #177777,(R0)
 1991 021452  000277                        SCC
 1992 021454  005110                        COM     (R0)
 1993 021456  104405                        CHECKCC+5               ; CHECK FOR CC = 5
 1994 021460  104423  021356                ERROR.  T57             ;; *** INCORRECT CC BITS ***
 1995
 1996 021464  012704  001010        NEG1:   MOV     #TEMP1,R4       ; LOAD ADDRESS
 1997 021470  012724  000001                MOV     #1,(R4)+                ; LOAD THE LOCATION
 1998 021474  010402                        MOV     R4,R2
 1999 021476  012762  100000  000000        MOV     #100000,0(R2)
 2000 021504  005444                        NEG     -(R4)           ; 2'S COMPLEMENT
 2001 021506  104411                        CHECKCC+11              ; CHECK FOR CC = 11
 2002 021510  104423  021464                ERROR.  NEG1            ;; *** INCORRECT CC BITS ***
```

PC 21510 - INCORRECT CC BITS.

```
2003 021514  022724  177777                CMP     #177777,(R4)+   ; CHECK IT
2004 021520  001402                        BEQ     2$              ; CONTINUE IF OK
2005 021522  104423  021464                ERROR.  NEG1            ;; *** NEG INSTRUCTION FAILED ***
2006 021526  016444  000000          2$:   MOV     0(R4),-(R4)     ; TEMP1 HOLDS LARGEST NEGATIVE NUMBER
2007 021532  005414                        NEG     (R4)            ; 2'S COMPLEMENT
2008 021534  104413                        CHECKCC,13              ; CHECK FOR CC = 13
2009 021536  104423  021464                ERROR.  NEG1            ;; *** INCORRECT CC BITS ***
2010 021542  026214  000000                CMP     0(R2),(R4)      ; CHECK IT
2011 021546  001402                        BEQ     3$              ; CONTINUE IF OK
2012 021550  104423  021464                ERROR.  NEG1            ;; *** NEG FAILED. ***
2013 021554                          3$:
2014
2015                                ;*****************************************************************************
                                    .SBTTL      T60 -- ROL AND ROR
                                    ;*****************************************************************************
     021554  104420  000060        T60:    BEGINTEST, 60           ;; CHECK SWITCH REGISTER OPTIONS.
2016 021560  012701  001012        ROL1:   MOV     #TEMP2,R1       ; LOAD ADDRESS
2017 021564  012711  020000                MOV     #20000,(R1)     ; LOAD LOCATION
2018 021570  000257                        CCC                     ; CLEAR FLAGS
2019 021572  006121                        ROL     (R1)+           ; SHIFT
2020 021574  006141                        ROL     -(R1)           ;
2021 021576  104412                        CHECKCC,12              ; CHECK FOR CC = 12
2022 021600  104423  021554                ERROR.  T60             ;; *** INCORRECT CC BITS ***
2023 021604  022711  100000                CMP     #100000,(R1)    ; CHECK IT
2024 021610  001402                        BEQ     1$              ; CONTINUE IF OK
2025 021612  104423  021554                ERROR.  T60             ;; *** ROL INSTRUCTION FAILED ***
2026 021616  006161  000000          1$:   ROL     0(R1)           ; SHIFT
2027 021622  104407                        CHECKCC,7               ; CHECK FOR CC = 7
2028 021624  104423  021554                ERROR.  T60             ;; *** INCORRECT CC BITS ***
2029 021630  010102                        MOV     R1,R2           ; R2 IS NOW POINTING TO LOCATION TEMP2
2030 021632  006112                        ROL     (R2)            ; SHIFT
2031 021634  022711  000001                CMP     #1,(R1)         ; CHECK IT
2032 021640  001402                        BEQ     ROR1            ; CONTINUE IF OK
2033 021642  104423  021554                ERROR.  T60             ;; *** ROL FAILED. ***
2034
2035 021646  012702  001012        ROR1:   MOV     #TEMP2,R2       ; LOAD ADDRESS
2036 021652  012712  000004                MOV     #4,(R2)         ; LOAD LOCATION
2037 021656  000257                        CCC                     ; CLEAR FLAGS
2038 021660  006012                        ROR     (R2)            ; SHIFT
2039 021662  006012                        ROR     (R2)            ;
2040 021664  022712  000001                CMP     #1,(R2)         ; CHECK IT
2041 021670  001402                        BEQ     1$              ; CONTINUE IF OK
2042 021672  104423  021646                ERROR.  ROR1            ;; *** ROR INSTRUCTION FAILED ***
2043 021676  006012                  1$:   ROR     (R2)            ; SHIFT
2044 021700  104407                        CHECKCC,7               ; CHECK FOR CC = 7
2045 021702  104423  021646                ERROR.  ROR1            ;; *** INCORRECT CC BITS ***
2046 021706  006012                        ROR     (R2)            ; SHIFT
2047 021710  104412                        CHECKCC,12              ; CHECK FOR CC = 12
2048 021712  104423  021646                ERROR.  ROR1            ;; *** INCORRECT CC BITS ***
2049 021716  022712  100000                CMP     #100000,(R2)    ; CHECK IT
2050 021722  001402                        BEQ     2$              ; CONTINUE IF OK
2051 021724  104423  021646                ERROR.  ROR1            ;; *** ROR FAILED. ***
2052 021730                          2$:
2053
2054                                ;*****************************************************************************
                                    .SBTTL      T61 -- ASL AND ASR
                                    ;*****************************************************************************
```

T61 -- ASL AND ASR

```
        021730  104420  000061          T61:    BEGINTEST, 61           ;; CHECK SWITCH REGISTER OPTIONS.
   2055 021734  012703  001012          ASL1:   MOV     #TEMP2,R3       ; LOAD ADDRESS
   2056 021740  012713  020000                  MOV     #20000,(R3)     ; LOAD LOCATION
   2057 021744  000257                          CCC                     ; CLEAR FLAGS
   2058 021746  006313                          ASL     (R3)            ; SHIFT
   2059 021750  006313                          ASL     (R3)            ;
   2060 021752  104412                          CHECKCC#12              ; CHECK FOR CC = 12
   2061 021754  104423  021730                  ERROR,  T61             ;; *** INCORRECT CC BITS ***
   2062 021760  022713  100000                  CMP     #100000,(R3)    ; CHECK IT
   2063 021764  001402                          BEQ     4$              ; CONTINUE IF OK
   2064 021766  104423  021730                  ERROR,  T61             ;; *** ASL INSTRUCTION FAILED ***
   2065 021772  006313                  4$:     ASL     (R3)            ; SHIFT
   2066 021774  104407                          CHECKCC#7               ; CHECK FOR CC = 7
   2067 021776  104423  021730                  ERROR,  T61             ;; *** INCORRECT CC BITS ***
   2068 022002  006313                          ASL     (R3)            ; SHIFT
   2069 022004  104404                          CHECKCC#4               ; CHECK FOR CC = 4
   2070 022006  104423  021730                  ERROR,  T61             ;; *** INCORRECT CC BITS ***
   2071
   2072 022012  012704  001012          ASR1:   MOV     #TEMP2,R4       ; LOAD ADDRESSES
   2073 022016  012703  001006                  MOV     #TEMP,R3        ;
   2074 022022  012714  000004                  MOV     #4,(R4)         ; LOAD LOCATION
   2075 022026  000257                          CCC                     ; CLEAR FLAGS
   2076 022030  006214                          ASR     (R4)            ; SHIFT
   2077 022032  006214                          ASR     (R4)            ;
   2078 022034  022714  000001                  CMP     #1,(R4)         ; CHECK IT
   2079 022040  001402                          BEQ     2$              ; CONTINUE IF OK
   2080 022042  104423  022012                  ERROR,  ASR1            ;; *** ASR INSTRUCTION FAILED ***
   2081 022046  006214                  2$:     ASR     (R4)            ; SHIFT
   2082 022050  104407                          CHECKCC#7               ; CHECK FOR CC = 7
   2083 022052  104423  022012                  ERROR,  ASR1            ;; *** INCORRECT CC BITS ***
   2084 022056  006214                          ASR     (R4)            ; SHIFT
   2085 022060  104404                          CHECKCC#4               ; CHECK FOR CC = 4
   2086 022062  104423  022012                  ERROR,  ASR1            ;; *** INCORRECT CC BITS ***
   2087 022066  012713  100002                  MOV     #100002,(R3)    ; LOAD LOCATION
   2088 022072  006213                          ASR     (R3)            ; SHIFT
   2089 022074  006213                          ASR     (R3)            ;
   2090 022076  104411                          CHECKCC#11              ; CHECK FOR CC = 11
   2091 022100  104423  022012                  ERROR,  ASR1            ;; *** INCORRECT CC BITS ***
   2092 022104  022713  160000                  CMP     #160000,(R3)    ; CHECK IT
   2093 022110  001402                          BEQ     3$              ; CONTINUE IF OK
   2094 022112  104423  022012                  ERROR,  ASR1            ;; *** ASR FAILED ***
   2095 022116                          3$:
   2096
   2097                                 ;*************************************************************************
                                        .SBTTL      T62 -- ADC AND SBC
                                        ;*************************************************************************
        022116  104420  000062          T62:    BEGINTEST, 62           ;; CHECK SWITCH REGISTER OPTIONS.
   2098 022122  012700  001006          ADC1:   MOV     #TEMP,R0        ; LOAD ADDRESS
   2099 022126  005010                          CLR     (R0)            ; CLEAR THE LOCATION
   2100 022130  000257                          CCC                     ; CLEAR FLAGS
   2101 022132  005510                          ADC     (R0)            ; ADD C BIT = 0
   2102 022134  104404                          CHECKCC#4               ; CHECK FOR CC = 4
   2103 022136  104423  022116                  ERROR,  T62             ;; *** INCORRECT CC BITS ***
   2104 022142  000261                          SEC                     ; C=1
   2105 022144  005510                          ADC     (R0)            ; ADD C BIT=1
   2106 022146  000261                          SEC                     ; C=1
   2107 022150  005510                          ADC     (R0)            ; AGAIN
```

PC 22136 - INCORRECT CC BITS.

```
2108 022152  104400                    CHECKCC,0               ; CHECK FOR CC = 0
2109 022154  104423  022116            ERROR,  T62             ;; *** INCORRECT CC BITS ***
2110 022160  022710  000002            CMP     #2,(R0)         ; CHECK IT
2111 022164  001402                    BEQ     4$              ; CONTINUE IF OK
2112 022166  104423  022116            ERROR,  T62             ;; *** ADC INSTRUCTION FAILED ***
2113 022172  012710  077777      4$:   MOV     #77777,(R0)     ; LOAD LARGEST POSITVE NUMBER
2114 022176  000261                    SEC                     ; C=1
2115 022200  005510                    ADC     (R0)            ; ADD C BIT=1
2116 022202  104412                    CHECKCC,12              ; CHECK FOR CC = 12
2117 022204  104423  022116            ERROR,  T62             ;; *** INCORRECT CC BITS ***
2118 022210  022710  100000            CMP     #100000,(R0)    ; CHECK IT
2119 022214  001402                    BEQ     6$              ; CONTINUE IF OK
2120 022216  104423  022116            ERROR,  T62             ;; *** ADC INSTRUCTION FAILED ***
2121 022222  012710  177777      6$:   MOV     #-1,(R0)        ; LOAD -1
2122 022226  000261                    SEC                     ; C=1
2123 022230  005510                    ADC     (R0)            ; ADD C BIT=1
2124 022232  104405                    CHECKCC,5               ; CHECK FOR CC = 5
2125 022234  104423  022116            ERROR,  T62             ;; *** INCORRECT CC BITS ***
2126
2127 022240  012701  001006      SBC1: MOV     #TEMP,R1        ; LOAD ADDRESS
2128 022244  012711  000003            MOV     #3,(R1)         ; LOAD LOCATION
2129 022250  000257                    CCC                     ; CLEAR FLAGS
2130 022252  005611                    SBC     (R1)            ; SUBTRACT C BIT=0
2131 022254  104400                    CHECKCC,0               ; CHECK FOR CC = 0
2132 022256  104423  022240            ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
2133 022262  022711  000003            CMP     #3,(R1)         ; CHECK IT
2134 022266  001402                    BEQ     2$              ; CONTINUE IF OK
2135 022270  104423  022240            ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
2136 022274  000261              2$:   SEC                     ; C=1
2137 022276  005611                    SBC     (R1)            ; SUBTRACT C BIT=1
2138 022300  000261                    SEC                     ; C=1
2139 022302  005611                    SBC     (R1)            ;
2140 022304  104400                    CHECKCC,0               ; CHECK FOR CC = 0
2141 022306  104423  022240            ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
2142 022312  022711  000001            CMP     #1,(R1)         ; CHECK IT
2143 022316  001402                    BEQ     3$
2144 022320  104423  022240            ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
2145 022324  000261              3$:   SEC                     ; C=1
2146 022326  005611                    SBC     (R1)            ; SUBTRACT C BIT=1
2147 022330  104404                    CHECKCC,4               ; CHECK FOR CC = 4
2148 022332  104423  022240            ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
2149 022336  000261                    SEC                     ; C=1
2150 022340  005611                    SBC     (R1)            ; SUBTRACT C BIT = 1
2151 022342  104411                    CHECKCC,11              ; CHECK FOR CC = 11
2152 022344  104423  022240            ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
2153 022350  022711  177777            CMP     #-1,(R1)        ; CHECK IT
2154 022354  001402                    BEQ     4$              ; CONTINUE IF OK
2155 022356  104423  022240            ERROR,  SBC1            ;; *** SBC INSTRUCTION FAILED ***
2156 022362  012711  100000      4$:   MOV     #100000,(R1)    ; LOAD R1
2157 022366  000261                    SEC                     ; C=1
2158 022370  005611                    SBC     (R1)            ; SUBTRACT C BIT = 1
2159 022372  104402                    CHECKCC,2               ; CHECK FOR CC = 2
2160 022374  104423  022240            ERROR,  SBC1            ;; *** INCORRECT CC BITS ***
2161
2162                                   ;*************************************************************************
                                       .SBTTL      T63 -- SXT, SWAB, AND XOR
                                       ;*************************************************************************
```

```
        022400  104420  000063          T63:    BEGINTEST, 63           ;; CHECK SWITCH REGISTER OPTIONS.
   2163 022404  012702  001010          SXT1:   MOV     #TEMP1,R2       ; LOAD ADDRESS
   2164 022410  005012                          CLR     (R2)            ; CLEAR LOCATIONS
   2165 022412  000277                          SCC
   2166 022414  000254                          CLNZ
   2167 022416  006712                          SXT     (R2)            ; SIGN EXTEND
   2168 022420  104405                          CHECKCC,5               ; CHECK FOR CC = 5
   2169 022422  104423  022400                  ERROR,  T63             ;; *** INCORRECT CC BITS ***
   2170 022426  005712                          TST     (R2)            ; LOCATION SHOULD STILL BE 0
   2171 022430  001402                          BEQ     2$              ; CONTINUE IF OK
   2172 022432  104423  022400                  ERROR,  T63             ;; *** SXT INSTRUCTION FAILED ***
   2173 022436  000273                  2$:     SENVC                   ; SET N, V & C BITS
   2174 022440  006712                          SXT     (R2)            ; SIGN EXTEND
   2175 022442  104411                          CHECKCC,11              ; CHECK FOR CC = 11
   2176 022444  104423  022400                  ERROR,  T63             ;; *** INCORRECT CC BITS ***
   2177 022450  022712  177777                  CMP     #-1,(R2)        ; LOCATION SHOULD NEW HAVE -1
   2178 022454  001402                          BEQ     SWAB1           ; CONTINUE IF OK
   2179 022456  104423  022400                  ERROR,  T63             ;; *** SXT FAILED ***
   2180
   2181 022462  012703  001012          SWAB1:  MOV     #TEMP2,R3       ; LOAD ADDRESS
   2182 022466  012713  125125                  MOV     #125125,(R3)    ; LOAD BIT PATTERN INTO LOCATION
   2183 022472  000277                          SCC
   2184 022474  000250                          CLN
   2185 022476  000313                          SWAB    (R3)            ; SWAP BYTES OF LOCATIONS
   2186 022500  104410                          CHECKCC,10              ; CHECK FOR CC = 10
   2187 022502  104423  022462                  ERROR,  SWAB1           ;; *** INCORRECT CC BITS ***
   2188 022506  022713  052652                  CMP     #52652,(R3)     ; CHECK IT
   2189 022512  001402                          BEQ     1$              ; CONTINUE IF OK
   2190 022514  104423  022462                  ERROR,  SWAB1           ;; *** SWAB INSTRUCTION FAILED ***
   2191 022520  012713  000377          1$:     MOV     #377,(R3)
   2192 022524  000277                          SCC
   2193 022526  000244                          CLZ
   2194 022530  000363  000000                  SWAB    0(R3)
   2195 022534  104404                          CHECKCC,4               ; CHECK FOR CC = 4
   2196 022536  104423  022462                  ERROR,  SWAB1           ;; *** INCORRECT CC BITS ***
   2197 022542  022713  177400                  CMP     #177400,(R3)
   2198 022546  001402                          BEQ     XOR1
   2199 022550  104423  022462                  ERROR,  SWAB1           ;; *** SWAB FAILED. ***
   2200
   2201 022554  012704  177777          XOR1:   MOV     #-1,R4          ; LOAD LOCATIONS
   2202 022560  012767  177777  156222          MOV     #-1,TEMP1       ;
   2203 022566  000277                          SCC
   2204 022570  074467  156214                  XOR     R4,TEMP1        ; SHOULD PRODUCE 0'S IN TEMP1
   2205 022574  104405                          CHECKCC,5               ; CHECK FOR CC = 5
   2206 022576  104423  022554                  ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
   2207 022602  012767  077777  156200          MOV     #77777,TEMP1
   2208 022610  012700  001010                  MOV     #TEMP1,R0       ; PLACE THE ADDRESS OF TEMP1 IN R0
   2209 022614  000263                          SEVC                    ; SET V & C BITS
   2210 022616  000244                          CLZ
   2211 022620  074410                          XOR     R4,(R0)
   2212 022622  104411                          CHECKCC,11              ; CHECK FOR CC = 11
   2213 022624  104423  022554                  ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
   2214 022630  012701  125252                  MOV     #125252,R1      ; LOAD LOCATIONS
   2215 022634  012720  052525                  MOV     #52525,(R0)+    ;
   2216 022640  000277                          SCC
   2217 022642  074140                          XOR     R1,-(R0)        ; SHOULD PRODUCE ALL 1'S IN TEMP1
   2218 022644  104411                          CHECKCC,11              ; CHECK FOR CC = 11
```

PC 22646 = INCORRECT CC BITS.

```
   2219 022646  104423  022554                 ERROR,  XOR1            ;; *** INCORRECT CC BITS ***
   2220 022652  022767  177777  156130         CMP     #-1,TEMP1       ; CHECK IT
   2221 022660  001402                         BEQ     1$              ; CONTINUE IF OK
   2222 022662  104423  022554                 ERROR,  XOR1            ;; *** XOR FAILED. ***
   2223 022666                         1$:
   2224
   2225                                ;*****************************************************************************
                                       .SBTTL      T64 -- ADD, SUB, AND SOB
                                       ;*****************************************************************************
        022666  104420  000064         T64:    BEGINTEST, 64           ;; CHECK SWITCH REGISTER OPTIONS.
   2226 022672  012700  001012         ADD1:   MOV     #TEMP2,R0       ; LOAD ADDRESSES
   2227 022676  012701  001006                 MOV     #TEMP,R1        ;
   2228 022702  012767  021421  156102         MOV     #21421,TEMP2    ; LOAD LOCATIONS
   2229 022710  011011                         MOV     (R0),(R1)       ;
   2230 022712  061011                         ADD     (R0),(R1)       ; ADD
   2231 022714  104400                         CHECKCC+0               ; CHECK FOR CC = 0
   2232 022716  104423  022666                 ERROR,  T64             ;; *** INCORRECT CC BITS ***
   2233 022722  022767  043042  156056         CMP     #43042,TEMP     ; CHECK IT
   2234 022730  001402                         BEQ     1$              ; CONTINUE IF OK
   2235 022732  104423  022666                 ERROR,  T64             ;; *** ADD INSTRUCTION FAILED ***
   2236 022736  005010                 1$:     CLR     (R0)            ; CLEAR LOCATION TEMP2
Z  2237 022740  060020                         ADD     R0,(R0)+        ; PLACE THE ADDRESS OF TEMP2 IN TEMP2
   2238                                                                 ; POSSIBLE "Z" ASSEMBLY ERROR O.K.      ;JSD REV B
   2239 022742  024027  001014                 CMP     -(R0),#TEMP2+2  ; CHECK IT.
   2240 022746  001402                         BEQ     2$
   2241 022750  104423  022666                 ERROR,  T64             ;; *** ADD INSTRUCTION FAILED IN MODE 2 ***
   2242 022754  012767  156357  156030 2$:     MOV     #-21421,TEMP2   ; LOAD LOCATIONS
   2243 022762  012011                         MOV     (R0)+,(R1)      ;
   2244 022764  064011                         ADD     -(R0),(R1)      ; ADD
   2245 022766  104411                         CHECKCC+11              ; CHECK FOR CC = 11
   2246 022770  104423  022666                 ERROR,  T64             ;; *** INCORRECT CC BITS ***
   2247 022774  022767  134736  156004         CMP     #-43042,TEMP    ; CHECK IT
   2248 023002  001402                         BEQ     3$              ;CONTINUE IF OK
   2249 023004  104423  022666                 ERROR,  T64             ;; *** ADD INSTRUCTION FAILED ***
   2250 023010  012767  100000  155774 3$:     MOV     #100000,TEMP2   ; LOAD LOCATIONS
   2251 023016  011061  000000                 MOV     (R0),0(R1)      ;
   2252 023022  066011  000000                 ADD     0(R0),(R1)      ; ADD SHOULD RESULT AS 0'S
   2253 023026  104407                         CHECKCC+7               ; CHECK FOR CC=7
   2254 023030  104423  022666                 ERROR,  T64             ;; *** INCORRECT CC BITS ***
   2255 023034  012767  021421  155746         MOV     #21421,TEMP1    ; LOAD LOCATION TEMP1
   2256 023042  012760  001010  000000         MOV     #TEMP1,0(R0)    ; PLACE THE ADDRESS OF TEMP1 IN TEMP2
   2257 023050  012711  156357                 MOV     #-21421,(R1)    ; LOAD LOCATION TEMP
   2258 023054  010004                         MOV     R0,R4           ; MAKE R4 POINT TO LOCATION TEMP2
   2259 023056  067411  000000                 ADD     @0(R4),(R1)     ; ADD SHOULD RESULT AS 0'S
   2260 023062  104405                         CHECKCC+5               ; CHECK FOR CC=5
   2261 023064  104423  022666                 ERROR,  T64             ;; *** INCORRECT CC BITS ***
   2262 023070  005430                         NEG     @(R0)+          ; NEGATE THE CONTENTS OF TEMP1
   2263 023072  012746  021421                 MOV     #21421,-(SP)    ; PLACE # 21421 ON THE STACK
   2264 023076  065066  000000                 ADD     @-(R0),0(SP)    ; ADD, SHOULD=0'S
   2265 023102  104405                         CHECKCC+5               ; CHECK FOR CC=5
   2266 023104  104423  022666                 ERROR,  T64             ;; *** INCORRECT CC BITS ***
   2267 023110  005726                         TST     (SP)+           ; CHECK THE STACK TO CONTAIN 0, ALSO
   2268                                                                 ; RESTORE THE STACK POINTER
   2269 023112  001402                         BEQ     4$
   2270 023114  104423  022666                 ERROR,  T64             ;; *** ADD INSTRUCTION FAILED IN MODE 5 ***
   2271 023120  012767  137777  155664 4$:     MOV     #137777,TEMP2
   2272 023126  062767  137777  155656         ADD     #137777,TEMP2
```

PC 23114 = ADD INSTRUCTION FAILED IN MODE 5.

```
2273 023134  104403                          CHECKCC+3                ; CHECK CC=3
2274 023136  104423  022666                  ERROR.  T64              ;; *** INCORRECT CC BITS ***
2275 023142  022767  077776  155642          CMP     #77776,TEMP2
2276 023150  001402                          BEQ     SUB1
2277 023152  104423  022666                  ERROR.  T64              ;; *** ADD FAILED ***
2278
2279 023156  012702  001006          SUB1:   MOV     #TEMP,R2         ; LOAD ADDRESSES
2280 023162  012703  001010                  MOV     #TEMP1,R3        ;
2281 023166  012767  021421  155612          MOV     #21421,TEMP      ; LOAD LOCATIONS
2282 023174  012767  156357  155606          MOV     #-21421,TEMP1    ;
2283 023202  161213                          SUB     (R2),(R3)        ; RESULT SHOULD=-43042
2284 023204  104410                          CHECKCC+10               ; CHECK FOR CC = 10
2285 023206  104423  023156                  ERROR.  SUB1             ;; *** INCORRECT CC BITS ***
2286 023212  022767  134736  155570          CMP     #-43042,TEMP1    ; CHECK IT
2287 023220  001402                          BEQ     1$               ;CONTINUE IF OK
2288 023222  104423  023156                  ERROR.  SUB1             ;; *** SUB INSTRUCTION FAILED ***
2289 023226  012767  021421  155554  1$:     MOV     #21421,TEMP1     ; LOAD LOCATION
2290 023234  161213                          SUB     (R2),(R3)        ; RESULT SHOULD=0
2291 023236  001402                          BEQ     2$
2292 023240  104423  023156                  ERROR.  SUB1             ;; *** SUB INSTRUCTION FAILED ***
2293 023244  012767  177777  155536  2$:     MOV     #-1,TEMP1        ; LOAD LOCATIONS
2294 023252  012767  077777  155526          MOV     #77777,TEMP      ; LOAD LOCATIONS
2295 023260  161312                          SUB     (R3),(R2)        ; RESULT SHOULD GIVE 100000 AND OVERFLOW
2296 023262  104413                          CHECKCC+13               ; CHECK FOR CC = 13
2297 023264  104423  023156                  ERROR.  SUB1             ;; *** INCORRECT CC BITS ***
2298 023270  022767  100000  155510          CMP     #100000,TEMP     ; CHECK IT
2299 023276  001402                          BEQ     3$               ; CONTINUE IF OK
2300 023300  104423  023156                  ERROR.  SUB1             ;; *** SUB INSTRUCTION FAILED ***
2301 023304  012712  177777          3$:     MOV     #-1,(R2)
2302 023310  161312                          SUB     (R3),(R2)
2303 023312  104404                          CHECKCC+4                ; CHECK FOR CC = 4
2304 023314  104423  023156                  ERROR.  SUB1             ;; *** INCORRECT CC BITS ***
2305 023320  012767  077777  155460          MOV     #77777,TEMP
2306 023326  162767  077777  155452          SUB     #77777,TEMP
2307 023334  104404                          CHECKCC+4                ; CHECK FOR CC=4
2308 023336  104423  023156                  ERROR.  SUB1             ;; *** INCORRECT CC BITS ***
2309 023342  005767  155440                  TST     TEMP
2310 023346  001402                          BEQ     SOB1             ; TEMP SHOULD BE =0
2311 023350  104423  023156                  ERROR.  SUB1             ;; *** SUB INSTRUCTION FAILED ***
2312
2313 023354  012700  000012          SOB1:   MOV     #10.,R0          ; LOAD REGISTERS
2314 023360  005001                          CLR     R1               ;
2315 023362  005201                  1$:     INC     R1               ; KEEP COUNT
2316 023364  020127  000012                  CMP     R1,#10.
2317 023370  003402                          BLE     2$
2318 023372  104423  023354                  ERROR.  SOB1             ;; *** SOB INSTRUCTION FAILED ***
2319 023376  000277                  2$:     SCC
2320 023400  077010                          SOB     R0,1$            ; SUB. 1 FROM REG. 0. GO BACK TO 1$
2321 023402  104417                          CHECKCC+17               ; CHECK FOR CC = 17
2322 023404  104423  023354                  ERROR.  SOB1             ;; *** INCORRECT CC BITS ***
2323 023410  005700                          TST     R0               ; REG. 0 = 0 ?
2324 023412  001402                          BEQ     3$               ; NO, FAILED
2325 023414  104423  023354                  ERROR.  SOB1             ;; *** SOB INSTRUCTION FAILED ***
2326 023420  022701  000012          3$:     CMP     #10.,R1          ; DID IT GO THRU 10 TIMES ?
2327 023424  001402                          BEQ     4$               ; CONTINUE IF OK
2328 023426  104423  023354                  ERROR.  SOB1             ;; *** SOB INSTRUCTION FAILED ***
2329 023432  012704  000010          4$:     MOV     #10,R4           ; PLACE #10 IN R4
```

PC 23426 = SOB INSTRUCTION FAILED.

```
2330 023436  077401                  5$:     SOB     R4,5$           ; STAY HERE UNTILL R4 = 0
2331 023440  005704                          TST     R4
2332 023442  001402                          BEQ     6$              ; CONTINUE IF OK
2333 023444  104423  023354                  ERROR,  SOB1            ;; *** SOB FAILED ***
2334 023450                          6$:
2335
2336                                 ;*************************************************************************
                                     .SBTTL      T65 -- MTPS AND MFPS
                                     ;*************************************************************************
     023450  104420  000065          T65:    BEGINTEST, 65           ;; CHECK SWITCH REGISTER OPTIONS.
2337 023454  012700  001006          TPSW1:  MOV     #TEMP,R0        ; PUT THE ADDRESS OF TEMP IN R0
2338 023460  012701  001010                  MOV     #TEMP1,R1       ; PUT THE ADDRESS OF TEMP1 IN R1
2339 023464  012711  177777                  MOV     #177777,(R1)    ; TEMP1 = 177777
2340 023470  005010                          CLR     (R0)            ; TEMP  = 0
2341 023472  106410                          MTPS    (R0)            ; PSW = 0
2342 023474  104400                          CHECKCC+0               ; CHECK FOR CC = 0
2343 023476  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2344 023502  106711                          MFPS    (R1)            ; MOVE PSW TO TEMP1
2345 023504  104404                          CHECKCC+4               ; CHECK FOR CC = 4
2346 023506  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2347 023512  022711  177400                  CMP     #177400,(R1)    ; CHECK TEMP1 TO MAKE SURE THAT  ONLY
2348                                                                 ; THE LOWER BYTE WAS AFFECTED BY MFPS
2349 023516  001402                          BEQ     1$
2350 023520  104423  023450                  ERROR,  T65             ;; *** MTPS OR MFPS INSTRUCTION FAILED ***
2351 023524  005011                  1$:     CLR     (R1)
2352 023526  106427  000337                  MTPS    #337            ; SET PSW = 317 SINCE T BIT CAN NOT BE SET BY MTPS
2353 023532  104417                          CHECKCC+17              ; CHECK FOR CC = 17
2354 023534  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2355 023540  106767  155244                  MFPS    TEMP1           ; MOVE PSW TO TEMP1
2356 023544  104411                          CHECKCC+11              ; CHECK FOR CC = 11 (C BIT SHOULD NOT BE EFFECTED BY MFPS)
2357 023546  104423  023450                  ERROR,  T65             ;; *** INCORRECT CC BITS ***
2358 023552  022767  000317  155230          CMP     #317,TEMP1
2359 023560  001402                          BEQ     2$
2360 023562  104423  023450                  ERROR,  T65             ;; *** MFPS INSTRUCTION FAILED IN MODE 6 ***
2361 023566                          2$:
2362
2363                                 ;*************************************************************************
                                     .SBTTL      T66 -- BYTE INSTRUCTIONS AFFECTING FULL WORDS
                                     ;*************************************************************************
     023566  104420  000066          T66:    BEGINTEST, 66           ;; CHECK SWITCH REGISTER OPTIONS.
2364 023572  005000                  BTWRD:  CLR     R0
2365 023574  000277                          SCC
2366 023576  112700  000200                  MOVB    #200,R0         ; SET THE HIGHEST BIT OF THE
2367                                                                 ; LOWER BYTE
2368 023602  104411                          CHECKCC+11              ; CHECK FOR CC=11
2369 023604  104423  023566                  ERROR,  T66             ;; *** INCORRECT CC BITS ***
2370 023610  022700  177600                  CMP     #177600,R0      ; CHECK FOR SIGN EXTENSION IN R0
2371 023614  001402                          BEQ     1$
2372 023616  104423  023566                  ERROR,  T66             ;; *** SIGN WAS NOT EXTENDED IN R0 ***
2373 023622  000277                  1$:     SCC
2374 023624  012700  177777                  MOV     #177777,R0
2375 023630  112700  000000                  MOVB    #0,R0           ;CLEAR THE LOWER BYTE OF R0.
2376 023634  104405                          CHECKCC+5               ; CHECK FOR CC=5
2377 023636  104423  023566                  ERROR,  T66             ;; *** INCORRECT CC BITS ***
2378 023642  005700                          TST     R0              ; CHECK R0 FOR SIGN EXTENTION
2379 023644  001402                          BEQ     2$
2380 023646  104423  023566                  ERROR,  T66             ;; *** SIGN WAS NOT EXTENDED IN R0. ***
```

PC 23646 = SIGN WAS NOT EXTENDED IN R0..

```
2381 023652  012704  001012          2$:     MOV     #TEMP2,R4       ; R4 IS POINTING TO TEMP2
2382 023656  012714  000377                  MOV     #377,(R4)       ; PLACE #377 IN LOCATION TEMP2
2383 023662  012706  003774                  MOV     #STACK-2,R6
2384 023666  116426  000000                  MOVB    0(R4),(R6)+     ; PUSH # 377 ON STACK
2385 023672  022706  003776                  CMP     #STACK,R6
2386 023676  001402                          BEQ     3$
2387 023700  104423  023566                  ERROR.  T66             ;; *** R6 DID NOT GET INCREMENTED ***
2388 023704  124627  000377          3$:     CMPB    -(R6),#377      ; CHECK LOCATION STACK-2 TO
2389                                                                  ; CONTAIN PROPER DATA
2390 023710  001402                          BEQ     4$
2391 023712  104423  023566                  ERROR.  T66             ;; *** BYTE INSTRUCTION IS FAILING WITH R6 ***
2392 023716  022706  003774          4$:     CMP     #STACK-2,R6     ; CHECK THAT R6 WAS DECREMENTED
2393                                                                  ; BY 2 BY A BYTE INSTRUCTION
2394 023722  001402                          BEQ     5$
2395 023724  104423  023566                  ERROR.  T66             ;; *** R6 WAS NOT DECREMENTED ***
2396 023730  016467  000000  155050  5$:     MOV     0(R4),TEMP      ; SET THE LOWER BYTE OF LOCATION TEMP
2397 023736  005726                          TST     (R6)+           ; RESTORE STACK POINTER
2398 023740  000277                          SCC
2399 023742  114667  155041                  MOVB    -(SP),TEMP+1    ; SET THE HIGHER BYTE OF LOCATION TEMP ,
2400 023746  104411                          CHECKCC+11              ; CHECK FOR CC=11
2401 023750  104423  023566                  ERROR.  T66             ;; *** INCORRECT CC BITS ***
2402 023754  022767  177777  155024          CMP     #177777,TEMP    ; CHECK TEMP FOR THE CORRECT VALUE
2403 023762  001402                          BEQ     6$
2404 023764  104423  023566                  ERROR.  T66             ;; *** TEMP FOULED UP ***
2405 023770  005067  155012          6$:     CLR     TEMP
2406 023774  000241                          CLC
2407 023776  105167  155005                  COMB    TEMP+1          ; WRITE 1'S IN THE HIGHER BYTE OF TEMP
2408 024002  104411                          CHECKCC+11              ; CHECK FOR CC=11
2409 024004  104423  023566                  ERROR.  T66             ;; *** INCORRECT CC BITS ***
2410 024010  022767  177400  154770          CMP     #177400,TEMP
2411 024016  001402                          BEQ     7$
2412 024020  104423  023566                  ERROR.  T66             ;; *** TEMP FOULED UP ***
2413
2414 024024  004767  011026          7$:     CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEG...
2415                                                                  ;...RETURN AND FALL THRU IF NOT.
2416                                  .SBTTL
```

```
                                        .SBTTL  SECTION 2 -- T11 TRAPS AND INTERRUPTS.

024030  012767  036350  012270  S2:     MOV     #SEG2,SEGN      ; SEGMENT 2 TEXT.

                                        ;*************************************************************************
                                        .SBTTL      T67 -- TEST AUTO INCREMENT/DECREMENT OF STACK POINTER (R6)
                                        ;*************************************************************************
024036  104420  000067         T67:    BEGINTEST, 67           ;; CHECK SWITCH REGISTER OPTIONS.
024042  012700  001026         R6TST:  MOV     #K4,R0          ; USE K4 AS PSUEDO STACK...
024046  010006                  1$:     MOV     R0,R6           ; INIT SP.
024050  112667  154732                  MOVB    (R6)+,TEMP      ;SIX SHOULD INCREMENT BY TWO
024054  020627  001030                  CMP     R6,#K4+2
024060  001402                          BEQ     1$
024062  104423  024036                  ERROR,  T67             ;; *** R6 DID NOT AUTO INCREMENT BY TWO ***
024066  010006                  1$:     MOV     R0,R6
024070  114667  154712                  MOVB    -(R6),TEMP      ;SHOULD DECREMENT BY TWO
024074  020627  001024                  CMP     R6,#K4-2
024100  001402                          BEQ     2$
024102  104423  024036                  ERROR,  T67             ;; *** R6 DID NOT AUTO DECREMENT BY 2 ***
024106  010006                  2$:     MOV     R0,R6
024110  112626                          MOVB    (R6)+,(R6)+     ;DOUBLE AUTO INCREMENT OF R6
024112  020627  001032                  CMP     R6,#K4+4
024116  001402                          BEQ     3$
024120  104423  024036                  ERROR,  T67             ;; *** WRONG AUTO INCREMENT OF R6 ***
024124  010006                  3$:     MOV     R0,R6
024126  005001                          CLR     R1
024130  122621                          CMPB    (R6)+,(R1)+     ;TEST INCREMENT OF R6 AND R1.
024132  020627  001030                  CMP     R6,#K4+2
024136  001402                          BEQ     4$
024140  104423  024036                  ERROR,  T67             ;; *** WRONG INCREMENT OF R6 ***
024144  020127  000001          4$:     CMP     R1,#1
024150  001402                          BEQ     5$
024152  104423  024036                  ERROR,  T67             ;; *** WRONG INCREMENT OF R1 ***
024156  010006                  5$:     MOV     R0,R6
024160  005001                          CLR     R1
024162  122126                          CMPB    (R1)+,(R6)+     ; TEST INCREMENT OF R1 AND R6.
024164  020127  000001                  CMP     R1,#1
024170  001402                          BEQ     6$
024172  104423  024036                  ERROR,  T67             ;; *** WRONG INCREMENT OF R1 ***
024176  020627  001030          6$:     CMP     R6,#K4+2
024202  001402                          BEQ     7$
024204  104423  024036                  ERROR,  T67             ;; *** WRONG INCREMENT OF R6 ***
024210                          7$:

                                        ;*************************************************************************
                                        .SBTTL      T70 -- TEST BYTE TRANSFERS USING STACK POINTER (R6)
                                        ;*************************************************************************
024210  104420  000070         T70:    BEGINTEST, 70           ;; CHECK SWITCH REGISTER OPTIONS.
024214  012767  123456  154606          MOV     #123456,K5
024222  012767  050505  154570          MOV     #050505,K1
024230  012701  001020                  MOV     #K1,R1          ;R1=(050505)K1
024234  012706  001030                  MOV     #K5,R6          ;R6=(123456)K5
024240  112621                          MOVB    (R6)+,(R1)+     ;LOW .BYE OF R6 0 R1
024242  022767  050456  154550          CMP     #050456,K1
024250  001402                          BEQ     1$
024252  104423  024210                  ERROR,  T70             ;; *** FALSE TRANSFER OF .BYTE ***
```

PC 24252 = FALSE TRANSFER OF .BYTE.

```
2469 024256  012767  123456  154544  1$:     MOV     #123456,K5
2470 024264  012767  050505  154526          MOV     #050505,K1
2471 024272  012701  001020                  MOV     #K1,R1          ;R1(050505)K1
2472 024276  012706  001032                  MOV     #K6,R6          ;R6(123456)K5
2473 024302  114621                          MOVB    -(R6),(R1)+     ;LOW .BYTE OF R6 TO R1 (DECREMENT)
2474 024304  026727  154510  050456          CMP     K1,#050456
2475 024312  001402                          BEQ     2$
2476 024314  104423  024210                  ERROR.  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2477
2478 024320  012767  123456  154472  2$:     MOV     #123456,K1
2479 024326  012767  050505  154474          MOV     #050505,K5
2480 024334  012701  001020                  MOV     #K1,R1          ;(123456)
2481 024340  012706  001030                  MOV     #K5,R6          ;(050505)
2482 024344  112126                          MOVB    (R1)+,(R6)+     ;LOW OF R1 TO LOW OF R6
2483 024346  022767  050456  154454          CMP     #050456,K5
2484 024354  001402                          BEQ     3$
2485 024356  104423  024210                  ERROR.  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2486
2487 024362  012767  123456  154430  3$:     MOV     #123456,K1
2488 024370  012767  050505  154432          MOV     #050505,K5
2489 024376  012701  001021                  MOV     #K1+1,R1        ;123456
2490 024402  012706  001030                  MOV     #K5,R6          ;050505
2491 024406  112126                          MOVB    (R1)+,(R6)+     ;HIGH OF R1 TO LOW OF R6
2492 024410  026727  154414  050647          CMP     K5,#050647
2493 024416  001402                          BEQ     4$
2494 024420  104423  024210                  ERROR.  T70             ;; *** FALSE R6 .BYTE TRANSFER ***
2495
2496 024424  012767  123456  154366  4$:     MOV     #123456,K1
2497 024432  012767  050505  154370          MOV     #050505,K5
2498 024440  012701  001021                  MOV     #K1+1,R1        ;R1-123456-ODD ADDRESS
2499 024444  012706  001030                  MOV     #K5,R6          ;R6-050505--.EVEN ADDRESS
2500 024450  112621                          MOVB    (R6)+,(R1)+     ;LOW OF R6 TO HIGH OF R1
2501 024452  022767  042456  154340          CMP     #042456,K1
2502 024460  001402                          BEQ     5$
2503 024462  104423  024210                  ERROR.  T70             ;; *** FAILED LOW OF 6 TO HIGH OF 1 ***
2504 024466                          5$:
2505
2506                                 ;*************************************************************************
                                     .SBTTL      T71 -- TEST BYTE COMPARES ON SEQUENTIAL ODD/EVEN ADDRESSES
                                     ;*************************************************************************
     024466  104420  000071        T71:    BEGINTEST, 71           ;; CHECK SWITCH REGISTER OPTIONS.
2507 024472  005067  154320                  CLR     K0              ; K0 <= 000000
2508 024476  012767  052525  154314          MOV     #052525,K1      ; K1 <= 052525
2509 024504  012767  052400  154310          MOV     #052400,K2      ; K2 <= 052400
2510 024512  126767  154302  154301          CMPB    K1,K1+1
2511 024520  001402                          BEQ     1$
2512 024522  104423  024466                  ERROR.  T71             ;; *** LOW TO HIGH IN SAME WORD FAILS ***
2513 024526  126767  154267  154264  1$:     CMPB    K1+1,K1
2514 024534  001402                          BEQ     2$
2515 024536  104423  024466                  ERROR.  T71             ;; *** HIGH TO LOW IN SAME WORD FAILS ***
2516 024542  126767  154255  154250  2$:     CMPB    K2+1,K1
2517 024550  001402                          BEQ     3$
2518 024552  104423  024466                  ERROR.  T71             ;; *** HIGH TO LOW IN DIFFERENT WORDS FAILS ***
2519 024556  126767  154240  154232  3$:     CMPB    K2,K0
2520 024564  001402                          BEQ     4$
2521 024566  104423  024466                  ERROR.  T71             ;; *** EVEN TO EVEN FAILED ***
2522 024572  126767  154223  154223  4$:     CMPB    K1+1,K2+1
```

PC 24566 = EVEN TO EVEN FAILED.

```
2523 024600  001402                          BEQ     5$
2524 024602  104423  024466                  ERROR,  T71             ;; *** ODD TO ODD FAILED ***
2525 024606  126767  154210  154207  5$:     CMPB    K2,K2+1
2526 024614  001002                          BNE     6$
2527 024616  104423  024466                  ERROR,  T71             ;; *** LOW TO HIGH IN SAME WORD FAILED ***
2528 024622  126767  154175  154173  6$:     CMPB    K2+1,K2+1
2529 024630  001402                          BEQ     7$
2530 024632  104423  024466                  ERROR,  T71             ;; *** HIGH TO HIGH IN SAME WORD FAILED ***
2531 024636  126767  154160  154155  7$:     CMPB    K2,K1+1
2532 024644  001002                          BNE     8$
2533 024646  104423  024466                  ERROR,  T71             ;; *** EVEN TO ODD FAILED ***
2534 024652                          8$:
2535
2536                                 ;*************************************************************************
                                     .SBTTL      T72 -- TEST THAT BUS TIME-OUT TRAPS TO THE "RESTART" ADDRESS
                                     ;*************************************************************************
     024652  104420  000072        T72:    BEGINTEST, 72           ;; CHECK SWITCH REGISTER OPTIONS.
2537                                 ;
2538                                 ; T11 TRAPS TO ODT "RESTART" ON TIME-OUT.  THE CODE THERE WILL
2539                                 ; EMULATE A TRAP THRU THE TRADITIONAL BUS ERROR VECTOR (4).
2540                                 ;
2541 024656  012706  003776                  MOV     #STACK,SP       ; SET STACK POINTER.
2542 024662  012767  024712  153114          MOV     #3$,ERRVEC      ; SET TRAP VECTOR.
2543 024670  106427  000000                  MTPS    #PRIO
2544 024674  005737  160000          1$:     TST     @#160000        ; NEXM -- TIME-OUT AND TRAP.
2545 024700  000257                          CCC                     ; SHOULD EXECUTE THIS...
2546 024702  000240                  2$:     NOP                     ;...THEN TRAP AND STACK THIS PC.
2547 024704  104423  024652                  ERROR,  T72             ;; *** BUS TIME-OUT DIDN'T TRAP AT ALL ***
2548 024710  000414                          BR      5$
2549 024712  011600                  3$:     MOV     (SP),R0         ; GET STACKED PC...
2550 024714  016601  000002                  MOV     2(SP),R1        ;...AND PSW.
2551 024720  020027  024702                  CMP     R0,#2$
2552 024724  001402                          BEQ     4$              ; BR IF PC IS RIGHT.
2553 024726  104423  024652                  ERROR,  T72             ;; *** STACKED PC INCORRECT ON TIME-OUT TRAP ***
2554
2555 024732  005701                  4$:     TST     R1
2556 024734  001402                          BEQ     5$              ; BR IF PSW IS RIGHT.
2557 024736  104423  024652                  ERROR,  T72             ;; *** STACKED PSW INCORRECT ON TIME-OUT TRAP ***
2558
2559 024742  012767  000006  153034  5$:     MOV     #ERRVEC+2,ERRVEC ; RESET TRAP CATCHER.
2560
2561                                 ;*************************************************************************
                                     .SBTTL      T73 -- TEST THAT AN "ILLEGAL" INSTRUCTION TRAPS TO 4
                                     ;*************************************************************************
     024750  104420  000073        T73:    BEGINTEST, 73           ;; CHECK SWITCH REGISTER OPTIONS.
2562                                 ;
2563                                 ; DO BOTH "JMP R" AND "JSR R,R"
2564                                 ; ON ERROR, R0 POINTS TO THE OFFENDER.
2565                                 ;
2566 024754  012700  025012                  MOV     #1$,R0          ; 1ST PASS USES A "JMP R"
2567 024760  000402                          SKP2
2568 024762  012700  025016          8$:     MOV     #2$,R0          ; 2ND PASS USES A "JSR R,R"
2569 024766  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
2570 024772  012767  025030  153004          MOV     #3$,ERRVEC      ; SET TRAP PC...
2571 025000  005067  153002                  CLR     ERRVEC+2        ;...AND PSW.
2572 025004  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2573 025010  000110                          JMP     (R0)            ; DISPATCH TO ONE OR THE OTHER.
```

T73 -- TEST THAT AN "ILLEGAL" INSTRUCTION TRAPS TO 4

```
2574 025012  000101                  1$:    JMP     R1              ; ILLEGAL JMP -- SHOULD TRAP.
2575 025014  000401                         SKP1
2576 025016  004101                  2$:    JSR     R1,R1           ; ILLEGAL JSR -- SHOULD TRAP.
2577 025020  000240                         NOP
2578 025022  104423  024750                 ERROR,  T73             ;; *** ILLEGAL JMP/JSR DIDN'T TRAP ***
2579 025026  000427                         BR      7$
2580
2581 025030  106701                  3$:    MFPS    R1              ; OK, GET CURRENT (NEW) PSW.
2582 025032  001402                         BEQ     4$              ; BR IF IT'S RIGHT.
2583 025034  104423  024750                 ERROR,  T73             ;; *** PSW INCORRECT AFTER ILLEGAL INSTR TRAP TO 4 ***
2584 025040  020627  003772          4$:    CMP     SP,#STACK-4
2585 025044  001402                         BEQ     5$              ; BR IF STACK IS RIGHT.
2586 025046  104423  024750                 ERROR,  T73             ;; *** STACK POINTER INCORRECT ***
2587 025052  021627  025014          5$:    CMP     (SP),#1$+2
2588 025056  001405                         BEQ     6$              ; BR IF STACKED PC IS RIGHT (JMP).
2589 025060  021627  025020                 CMP     (SP),#2$+2
2590 025064  001402                         BEQ     6$              ; BR IF STACKED PC IS RIGHT (JSR).
2591 025066  104423  024750                 ERROR,  T73             ;; *** INCORRECT PC SAVED ON STACK ***
2592 025072  026627  000002  000317  6$:    CMP     2(SP),#PRI6!17
2593 025100  001402                         BEQ     7$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2594 025102  104423  024750                 ERROR,  T73             ;; *** INCORRECT PSW SAVED ON STACK ***
2595
2596 025106  020027  025012          7$:    CMP     R0,#1$          ; IS THIS THE 1ST PASS ??
2597 025112  001723                         BEQ     8$              ; LOOP ONCE IF SO...
2598 025114  012767  000006  152662         MOV     #ERRVEC+2,ERRVEC ;...OTHERWISE, RESET VECTOR...
2599
2600                                 ;*************************************************************************
                                     .SBTTL      T74 -- TEST THAT A RESERVED (UNDEFINED) INSTRUCTION TRAPS TO 10
                                     ;*************************************************************************
     025122  104420  000074          T74:   BEGINTEST, 74           ;; CHECK SWITCH REGISTER OPTIONS.
2601 025126  012706  003776                 MOV     #STACK,SP       ;STACK POINTER SETUP
2602 025132  012767  025162  152650         MOV     #2$,RESVEC      ; SET TRAP PC...
2603 025140  005067  152646                 CLR     RESVEC+2        ;...AND PSW.
2604 025144  106427  000317                 MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2605 025150  000010                         RESRVD                  ; RESERVED INSTRUCTION -- SHOULD TRAP
2606 025152  000240                  1$:    NOP
2607 025154  104423  025122                 ERROR,  T74             ;; *** RESERVED OPCODE DIDN'T TRAP ***
2608 025160  000424                         BR      6$
2609
2610 025162  106700                  2$:    MFPS    R0
2611 025164  001402                         BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2612 025166  104423  025122                 ERROR,  T74             ;; *** NEW PSW INCORRECT AFTER TRAP TO 10 ***
2613 025172  020627  003772          3$:    CMP     SP,#STACK-4
2614 025176  001402                         BEQ     4$              ; BR IF SP PUSHED DOWN RIGHT.
2615 025200  104423  025122                 ERROR,  T74             ;; *** STACK POINTER INCORRECT ***
2616 025204  021627  025152          4$:    CMP     (SP),#1$
2617 025210  001402                         BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2618 025212  104423  025122                 ERROR,  T74             ;; *** INCORRECT PC SAVED ON STACK ***
2619 025216  026627  000002  000317  5$:    CMP     2(SP),#PRI6!17
2620 025224  001402                         BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2621 025226  104423  025122                 ERROR,  T74             ;; *** INCORRECT PSW SAVED ON STACK ***
2622
2623 025232  012767  000012  152550  6$:    MOV     #RESVEC+2,RESVEC ; RESET VECTOR.
2624
2625                                 ;*************************************************************************
                                     .SBTTL      T75 -- TEST THAT A "BPT" INSTRUCTION TRAPS TO 14
                                     ;*************************************************************************
```

```
                025240  104420  000075          T75:    BEGINTEST, 75           ;; CHECK SWITCH REGISTER OPTIONS.
2626            025244  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
2627            025250  012767  025300  152536          MOV     #2$,BPTVEC      ; SET TRAP PC...
2628            025256  005067  152534                  CLR     BPTVEC+2        ;...AND PSW.
2629            025262  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2630            025266  000003                          BPT                     ; BPT -- SHOULD TRAP.
2631            025270  000240                  1$:     NOP
2632            025272  104423  025240                  ERROR,  T75             ;; *** BPT DIDN'T TRAP ***
2633            025276  000424                          BR      6$
2634
2635            025300  106700                  2$:     MFPS    R0
2636            025302  001402                          BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2637            025304  104423  025240                  ERROR,  T75             ;; *** PSW INCORRECT AFTER BPT TRAP TO 14 ***
2638            025310  020627  003772          3$:     CMP     SP,#STACK-4
2639            025314  001402                          BEQ     4$              ; BR IF STACK IS RIGHT.
2640            025316  104423  025240                  ERROR,  T75             ;; *** STACK POINTER INCORRECT ***
2641            025322  021627  025270          4$:     CMP     (SP),#1$
2642            025326  001402                          BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2643            025330  104423  025240                  ERROR,  T75             ;; *** INCORRECT PC SAVED ON STACK ***
2644            025334  026627  000002  000317  5$:     CMP     2(SP),#PRI6!17
2645            025342  001402                          BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2646            025344  104423  025240                  ERROR,  T75             ;; *** INCORRECT PSW SAVED ON STACK ***
2647
2648            025350  012767  000016  152436  6$:     MOV     #BPTVEC+2,BPTVEC ; RESET VECTOR.
2649
2650                                                    ;*******************************************************************
                                                        .SBTTL      T76 -- TEST THAT AN "IOT" INSTRUCTION TRAPS TO 20
                                                        ;*******************************************************************
                025356  104420  000076          T76:    BEGINTEST, 76           ;; CHECK SWITCH REGISTER OPTIONS.
2651            025362  012706  003776                  MOV     #STACK,SP       ;STACK POINTER SETUP
2652            025366  012767  025416  152424          MOV     #2$,IOTVEC      ; SET TRAP PC...
2653            025374  005067  152422                  CLR     IOTVEC+2        ;...AND PSW.
2654            025400  106427  000317                  MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2655            025404  000004                          IOT                     ; IOT -- SHOULD TRAP.
2656            025406  000240                  1$:     NOP
2657            025410  104423  025356                  ERROR,  T76             ;; *** IOT DID NOT TRAP ***
2658            025414  000424                          BR      6$
2659
2660            025416  106700                  2$:     MFPS    R0
2661            025420  001402                          BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2662            025422  104423  025356                  ERROR,  T76             ;; *** PSW INCORRECT AFTER IOT TRAP TO 20 ***
2663            025426  020627  003772          3$:     CMP     SP,#STACK-4
2664            025432  001402                          BEQ     4$              ; BR IS STACK IS RIGHT.
2665            025434  104423  025356                  ERROR,  T76             ;; *** STACK POINTER INCORRECT ***
2666            025440  021627  025406          4$:     CMP     (SP),#1$
2667            025444  001402                          BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2668            025446  104423  025356                  ERROR,  T76             ;; *** INCORRECT PC SAVED ON STACK ***
2669            025452  026627  000002  000317  5$:     CMP     2(SP),#PRI6!17
2670            025460  001402                          BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2671            025462  104423  025356                  ERROR,  T76             ;; *** INCORRECT PSW SAVED ON STACK ***
2672
2673            025466  012767  000022  152324  6$:     MOV     #IOTVEC+2,IOTVEC ; RESET VECTOR.
2674
2675                                                    ;*******************************************************************
                                                        .SBTTL      T77 -- TEST THAT "POWER-FAIL" TRAPS TO 24 (REQUIRES Q-BUS EXERCISER)
                                                        ;*******************************************************************
                025474  104420  000077          T77:    BEGINTEST, 77           ;; CHECK SWITCH REGISTER OPTIONS.
```

```
 2676 025500  012706  003776                    MOV     #STACK,SP       ;STACK POINTER SETUP
 2677 025504  004767  007700                    CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
 2678 025510  001456                            BEQ     7$              ; BYPASS TEST IF NOT.
 2679 025512  010004                            MOV     R0,R4           ; YES, MOVE POINTER TO R4 AND FALL THRU.
 2680                                 ;
 2681                                 ; QBX CSR1<13> IS USED TO EMULATE A POWER DOWN/UP SEQUENCE.
 2682                                 ; CAUTION:
 2683                                 ; WE ONLY WANT TO RECOGNIZE THE "DOWN" HALF OF THE POWER SEQUENCE.
 2684                                 ; ABORT THE PWR-CYCLE AT THAT POINT SO WE DON'T CRASH TO ODT.
 2685                                 ;
 2686 025514  012767  025564  152302            MOV     #2$,PWRVEC      ; SET POWER TRAP PC...
 2687 025522  005067  152300                    CLR     PWRVEC+2        ;...AND PSW.
 2688 025526  106427  000300                    MTPS    #PRI6           ; SET CURRENT (OLD) PSW.
 2689 025532  012700  140000                    MOV     #140000,R0      ; IF LOOPING, WE HAVE TO WAIT LONG...
 2690 025536  077001                            SOB     R0,.            ;...ENOUGH (~~250MSEC) TO INSURE THE...
 2691                                                                    ;...PREVIOUS BPOK CYCLE WAS FINISHED...
 2692                                                                    ;...(IN QBX) BEFORE TRYING ANOTHER.
 2693 025540  012700  000005                    MOV     #5,R0           ; OK, SET A 25 USEC TIMER.
 2694 025544  052714  020000                    BIS     #20000,(R4)     ; SET PWR-CYCLE REQUEST...
 2695 025550  000257                            CCC                     ;...SHOULD SEE IT AFTER THIS...
 2696 025552  077001                 1$:        SOB     R0,.            ;...AND STACK THIS PC.
 2697 025554  005014                            CLR     (R4)            ; TRAP NOT RECEIVED, CLEAR THE REQUEST.
 2698 025556  104423  025474                    ERROR.  T77             ;; *** POWER-FAIL DID NOT TRAP ***
 2699 025562  000426                            BR      6$
 2700
 2701 025564  106700                 2$:        MFPS    R0              ; OK, GET NEW PSW...
 2702 025566  005014                            CLR     (R4)            ;...AND CLEAR PWR-CYCLE REQUEST.
 2703 025570  005700                            TST     R0
 2704 025572  001402                            BEQ     3$              ; BR IF NEW PSW IS RIGHT.
 2705 025574  104423  025474                    ERROR.  T77             ;; *** PSW INCORRECT AFTER PWR-FAIL TRAP TO 24 ***
 2706 025600  020627  003772          3$:       CMP     SP,#STACK-4
 2707 025604  001402                            BEQ     4$              ; BR IS STACK IS RIGHT.
 2708 025606  104423  025474                    ERROR.  T77             ;; *** STACK POINTER INCORRECT ***
 2709 025612  021627  025552          4$:       CMP     (SP),#1$
 2710 025616  001402                            BEQ     5$              ; BR IF STACKED PC IS RIGHT.
 2711 025620  104423  025474                    ERROR.  T77             ;; *** INCORRECT PC SAVED ON STACK ***
 2712 025624  026627  000002  000300  5$:        CMP     2(SP),#PRI6
 2713 025632  001402                            BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
 2714 025634  104423  025474                    ERROR.  T77             ;; *** INCORRECT PSW SAVED ON STACK ***
 2715
 2716 025640  012767  000026  152156  6$:       MOV     #PWRVEC+2,PWRVEC ; RESTORE VECTOR.
 2717 025646                        7$:
 2718
 2719                                 ;*************************************************************************
                                      .SBTTL      T100 -- TEST THAT AN "EMT" INSTRUCTION TRAPS TO 30
                                      ;*************************************************************************
      025646  104420  000100         T100:   BEGINTEST, 100          ;; CHECK SWITCH REGISTER OPTIONS.
 2720                                 ;
 2721                                 ; NOW EMT'S HAVE TO BE OK OR WE'D NEVER HAVE GOTTEN STARTED IN THE FIRST
 2722                                 ; PLACE.  WE'VE ALREADY VERIFIED ALL VARIATIONS OF THE EMT OPCODE,
 2723                                 ; SO WE'LL JUST TEST FOR CORRECT STACK MANIPULATION HERE.
 2724                                 ;
 2725 025652  012706  003776                    MOV     #STACK,SP       ;STACK POINTER SETUP
 2726 025656  016701  152146                    MOV     EMTVEC,R1       ; SAVE EMT VECTOR IN R1.
 2727 025662  012767  025706  152140            MOV     #2$,EMTVEC      ; SET TRAP PC...
 2728 025670  005067  152136                    CLR     EMTVEC+2        ;...AND PSW.
 2729 025674  106427  000317                    MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
```

T100 -- TEST THAT AN "EMT" INSTRUCTION TRAPS TO 30

```
2730 025700  104000                        EMT                     ; EMT -- SHOULD TRAP...
2731 025702  000240              1$:       NOP                     ;...AND STACK THIS PC.
2732 025704  000240                        NOP                     ; IF IT DOESN'T, JUST FALL THRU ANYWAY.
2733
2734 025706  106700              2$:       MFPS    R0              ; GET NEW PSW...
2735 025710  010167  152114                MOV     R1,EMTVEC       ;...AND RESTORE VECTOR.
2736 025714  005700                        TST     R0
2737 025716  001402                        BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2738 025720  104423  025646                ERROR,  T100            ;; *** PSW INCORRECT AFTER EMT TRAP TO 30 ***
2739 025724  020627  003772      3$:       CMP     SP,#STACK-4
2740 025730  001402                        BEQ     4$              ; BR IF SP IS RIGHT.
2741 025732  104423  025646                ERROR,  T100            ;; *** STACK POINTER INCORRECT ***
2742 025736  021627  025702      4$:       CMP     (SP),#1$
2743 025742  001402                        BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2744 025744  104423  025646                ERROR,  T100            ;; *** INCORRECT PC SAVED ON STACK ***
2745 025750  026627  000002  000317  5$:   CMP     2(SP),#PRI6!17
2746 025756  001402                        BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2747 025760  104423  025646                ERROR,  T100            ;; *** INCORRECT PSW SAVED ON STACK ***
2748 025764              6$:
2749
2750                                       ;*************************************************************************
                                           .SBTTL      T101 -- TEST THAT A "TRAP" INSTRUCTION TRAPS TO 34
                                           ;*************************************************************************
     025764  104420  000101              T101:   BEGINTEST, 101          ;; CHECK SWITCH REGISTER OPTIONS.
2751                                       ;
2752                                       ; WE ALREADY KNOW THAT ALL VARIATIONS OF THE TRAP OPCODE WORK.
2753                                       ; SO WE'LL JUST CHECK THAT THE STACK IS HANDLED PROPERLY.
2754                                       ;
2755 025770  012706  003776                MOV     #STACK,SP       ;STACK POINTER SETUP
2756 025774  016701  152034                MOV     TRPVEC,R1       ; SAVE TRAP VECTOR IN R1.
2757 026000  012767  026024  152026        MOV     #2$,TRPVEC      ; SET TRAP PC...
2758 026006  005067  152024                CLR     TRPVEC+2        ;...AND PSW.
2759 026012  106427  000317                MTPS    #PRI6!17        ; SET CURRENT (OLD) PSW.
2760 026016  104400                        TRAP+0                  ; TRAP -- SHOULD TRAP...
2761 026020  000240              1$:       NOP                     ;...AND STACK THIS PC.
2762 026022  000240                        NOP                     ; IF IT DOESN'T, JUST FALL THRU ANYWAY.
2763
2764 026024  106700              2$:       MFPS    R0              ; GET NEW PSW...
2765 026026  010167  152002                MOV     R1,TRPVEC       ;...AND RESTORE VECTOR.
2766 026032  005700                        TST     R0
2767 026034  001402                        BEQ     3$              ; BR IF NEW PSW IS RIGHT.
2768 026036  104423  025764                ERROR,  T101            ;; *** PSW INCORRECT AFTER TRAP TO 34 ***
2769 026042  020627  003772      3$:       CMP     SP,#STACK-4
2770 026046  001402                        BEQ     4$              ; BR IF STACK IS RIGHT.
2771 026050  104423  025764                ERROR,  T101            ;; *** STACK POINTER INCORRECT ***
2772 026054  021627  026020      4$:       CMP     (SP),#1$
2773 026060  001402                        BEQ     5$              ; BR IF STACKED PC IS RIGHT.
2774 026062  104423  025764                ERROR,  T101            ;; *** INCORRECT PC SAVED ON STACK ***
2775 026066  026627  000002  000317  5$:   CMP     2(SP),#PRI6!17
2776 026074  001402                        BEQ     6$              ; BR IF STACKED (OLD) PSW IS RIGHT.
2777 026076  104423  025764                ERROR,  T101            ;; *** INCORRECT PSW SAVED ON STACK ***
2778 026102              6$:
2779
2780                                       ;*************************************************************************
                                           .SBTTL      T102 -- TEST THAT THE "T" BIT CAUSES A TRACE TRAP
                                           ;*************************************************************************
     026102  104420  000102              T102:   BEGINTEST, 102          ;; CHECK SWITCH REGISTER OPTIONS.
```

T102 -- TEST THAT THE "T" BIT CAUSES A TRACE TRAP

```
2781                                            ;
2782                                            ; FIRST, SET "T" VIA RTI -- SHOULD TRAP TO 14 BEFORE NEXT INSTRUCTION.
2783                                            ;
2784 026106  012706  003776                     MOV     #STACK,SP
2785 026112  012767  026150  151674             MOV     #3$,TRTVEC      ; SET TRAP PC...
2786 026120  005067  151672                     CLR     TRTVEC+2        ;...AND NEW PSW.
2787 026124  012746  000020                     MOV     #20,-(SP)       ; PUSH OLD PSW WITH T BIT...
2788 026130  012746  026136                     MOV     #.+6,-(SP)      ;...AND PC.
2789 026134  000002                            RTI                     ; SHOULD SET T BIT...
2790 026136  000240                     1$:    NOP                     ;...AND TRAP BEFORE THIS EXECUTES.
2791 026140  000240                     2$:    NOP
2792 026142  104423  026102                     ERROR.  T102            ;; *** TRACE BIT DID NOT TRAP ***
2793 026146  000417                            BR      6$
2794
2795 026150  106700                     3$:    MFPS    R0              ; TRAPPED OK, T SHOULD BE CLEAR NOW.
2796 026152  001402                            BEQ     4$              ; BR IF SO.
2797 026154  104423  026102                     ERROR.  T102            ;; *** NEW PSW INCORRECT AFTER TRACE TRAP TO 14 ***
2798 026160  021627  026136              4$:    CMP     (SP),#1$        ; IT SHOULD HAVE TRAPPED BEFORE...
2799 026164  001402                            BEQ     5$              ;...THE INSTRUCTION AT 1$.
2800 026166  104423  026102                     ERROR.  T102            ;; *** TRACE TRAP HAPPENED AT WRONG TIME ***
2801 026172  026627  000002  000020       5$:    CMP     2(SP),#20       ; AND T BIT SHOULD HAVE BEEN STACKED...
2802 026200  001402                            BEQ     6$              ;...ALONG WITH THE OLD PSW.
2803 026202  104423  026102                     ERROR.  T102            ;; *** T BIT WASN'T SAVED IN STACKED PSW ***
2804                                            ;
2805                                            ; NOW SET "T" VIA RTT -- SHOULD ALLOW NEXT INSTRUCTION, THEN TRAP.
2806                                            ;
2807 026206  012767  026240  151600       6$:    MOV     #8$,TRTVEC      ; CHANGE VECTOR.
2808 026214  012746  000020                     MOV     #20,-(SP)       ; PUSH OLD PSW WITH T BIT...
2809 026220  012746  026226                     MOV     #.+6,-(SP)      ;...AND PC.
2810 026224  000006                            RTT                     ; SHOULD SET T BIT...
2811 026226  000240                            NOP                     ;...EXECUTE THIS...
2812 026230  000240                     7$:    NOP                     ;...AND TRAP BEFORE THIS.
2813 026232  104423  026102                     ERROR.  T102            ;; *** TRACE TRAP DIDN'T HAPPEN ***
2814 026236  000417                            BR      11$
2815
2816 026240  106700                     8$:    MFPS    R0              ; OK, T SHOULD BE CLEAR NOW.
2817 026242  001402                            BEQ     9$
2818 026244  104423  026102                     ERROR.  T102            ;; *** NEW PSW INCORRECT AFTER TRACE TRAP TO 14 ***
2819 026250  021627  026230              9$:    CMP     (SP),#7$        ; AND IT SHOULD HAVE ALLOWED...
2820 026254  001402                            BEQ     10$             ;...THE 1ST NOP TO EXECUTE.
2821 026256  104423  026102                     ERROR.  T102            ;; *** RTT DIDN'T ALLOW NEXT INSTRUCTION TO EXECUTE ***
2822 026262  026627  000002  000020      10$:    CMP     2(SP),#20       ; AND THE T BIT SHOULD BE ON STACK.
2823 026270  001402                            BEQ     11$
2824 026272  104423  026102                     ERROR.  T102            ;; *** T BIT WASN'T SAVED IN STACKED PSW ***
2825 026276                             11$:
2826
2827                                            ;*****************************************************************************
                                                .SBTTL      T103 -- TEST THAT TRACE TRAP ON A TRAP IS INHIBITED
                                                ;*****************************************************************************
     026276  104420  000103                T103:   BEGINTEST, 103          ;; CHECK SWITCH REGISTER OPTIONS.
2828 026302  012706  003776                     MOV     #STACK,SP
2829 026306  012767  026360  151500             MOV     #1$,TRTVEC      ; SET TRACE VECTOR...
2830 026314  005067  151476                     CLR     TRTVEC+2        ;...AND PSW.
2831 026320  012767  026370  151472             MOV     #2$,IOTVEC      ; SET IOT VECTOR...
2832 026326  005067  151470                     CLR     IOTVEC+2        ;...AND PSW.
2833 026332  012746  000020                     MOV     #20,-(SP)       ; PUSH PSW WITH T BIT...
2834 026336  012746  026344                     MOV     #.+6,-(SP)      ;...AND PC.
```

T103 -- TEST THAT TRACE TRAP ON A TRAP IS INHIBITED

```
                                        RTT                     ; SET T BIT...
                                        IOT                     ;...AND XCT ANOTHER TRAP...
                                        NOP                     ; THE IOT SHOULD WIN...
                                        NOP                     ;...AND INHIBIT THE TRACE.
                        026276          ERROR,  T103            ;; *** NEITHER TRAP OCCURRED ??? ***
                                        BR      2$
                                1$:     NOP                     ; IOT SHOULD HAVE CLEARED T...
                                        NOP                     ;...AND INHIBITED THE TRACE.
                        026276          ERROR,  T103            ;; *** TRACE TRAP ON TRAP WASN'T INHIBITED ***
                        151422  2$:     MOV     #IOTVEC+2,IOTVEC ; RESET IOT VECTOR.

                                ;*******************************************************************************
                                .SBTTL      T104 -- TEST THAT RESET HAS NO EFFECT ON A TRACE TRAP
                                ;*******************************************************************************
                                T104:   BEGINTEST, 104          ;; CHECK SWITCH REGISTER OPTIONS.
                                        MOV     #STACK,SP       ;SET STACK
                        151400          MOV     #1$,TRTVEC      ; SET TRACE VECTOR.
                                        CLR     TRTVEC+2
                                        MOV     #20,-(SP)       ; PUSH PSW WITH T BIT...
                                        MOV     #.+6,-(SP)      ;...AND PC.
                                        RTT                     ; SET "T" BIT...
                                        RESET                   ;...AND XCT BUS RESET.
                                        NOP                     ; SHOULD HAVE NO EFFECT...
                                        NOP                     ;... I.E. TRAP SHOULD STILL OCCUR...
                                        NOP                     ;...AFTER THE RESET INSTRUCTION.
                        026376          ERROR,  T104            ;; *** BUS-RESET DISABLED OR INHIBITED TRACE TRAP ***

                                1$:     MTPS    #PRIO           ; WE'RE ALL DONE.
                        151334          MOV     #TRTVEC+2,TRTVEC ; RESET TRACE VECTOR.

                                ;*******************************************************************************
                                .SBTTL      T105 -- TEST THAT ODD ADDRESS TRAPS ARE NOT IMPLEMENTED
                                ;*******************************************************************************
                                T105:   BEGINTEST, 105          ;; CHECK SWITCH REGISTER OPTIONS.
                                        MOV     #STACK,SP       ; SET STACK.
                        151306          MOV     #2$,ERRVEC      ; IN CASE ODD ADDRESS TRAPS.
                                        JMP     1$+1
                                        NOP                     ; IT DIDN'T DO ANYTHING AT ALL !!
                        026460          ERROR,  T105            ;; *** JUMP TO AN ODD ADDRESS DID NOTHING ***
                                1$:     MOV     #3$,PC          ; ADJUST ODD PC AND EXIT.
                                2$:
                        026460          ERROR,  T105            ;; *** JMP TO AN ODD ADDRESS TIMED OUT ***
                        151256  3$:     MOV     #ERRVEC+2,ERRVEC ; RESET THE TRAP CATCHER.

                                ;*******************************************************************************
                                .SBTTL      T106 -- TEST THE CPU TYPE INSTRUCTION (MFPT)
                                ;*******************************************************************************
                                T106:   BEGINTEST, 106          ;; CHECK SWITCH REGISTER OPTIONS.
                                        MOV     #STACK,SP       ; SET STACK.
                        151244          MOV     #2$,RESVEC      ; SET TRAP CATCHER.
                                        CLR     R0
                                        CCC                     ; CLEAR CC BITS.
                                        MFPT                    ; MOVE FROM PROC TYPE (TO R0).
                                        MFPS    R1
                                        BIT     #17,R1
```

T106 -- TEST THE CPU TYPE INSTRUCTION (MFPT)

```
2882 026560  001402                          BEQ     1$
2883 026562  104423  026526                  ERROR,  T106            ;; *** CC BITS CHANGED ON MFPT INSTRUCTION ***
2884 026566  020027  000004          1$:     CMP     R0,#KXT11
2885 026572  001405                          BEQ     3$
2886 026574  104423  026526                  ERROR,  T106            ;; *** RETURNED VALUE INCORRECT ON MFPT ***
2887 026600  000402                          BR      3$
2888 026602                          2$:
     026602  104423  026526                  ERROR,  T106            ;; *** MFPT TRAPPED TO 10 ***
2889 026606  012767  000012  151174  3$:     MOV     @RESVEC+2,RESVEC
2890
2891                                 ;*************************************************************************
                                     .SBTTL      T107 -- TEST EXTERNAL Q-BUS INTERRUPT (REQUIRES Q-BUS EXERCISER)
                                     ;*************************************************************************
     026614  104420  000107          T107:   BEGINTEST, 107          ;; CHECK SWITCH REGISTER OPTIONS.
2892 026620  012706  003776                  MOV     #STACK,SP
2893 026624  004767  006560                  CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
2894 026630  001444                          BEQ     4$              ; IF NOT, BYPASS TEST.
2895 026632  010004                          MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
2896                                 ;
2897                                 ; USE QBX TO FAKE A Q-BUS INTERRUPT (NO DMA).
2898                                 ;
2899 026634  012703  000110                  MOV     #QBXVEC,R3      ; GET VECTOR POINTER.
2900 026640  005014                          CLR     (R4)            ; CLEAR CSR1 (NO DMA REQUIRED).
2901 026642  012713  026706                  MOV     #1$,(R3)        ; SET Q-BUS VECTOR...
2902 026646  012763  000200  000002          MOV     #PRI4,2(R3)     ;...AND PRIORITY.
2903 026654  106427  000200                  MTPS    #PRI4           ; RAISE CPU TO Q-BUS LEVEL.
2904 026660  012764  000001  000002          MOV     #1,2(R4)        ; SET QBX GO (CLEAR DONE)...
2905 026666  052764  000020  000002          BIS     #20,2(R4)       ;...ENABLE INT 4 ON DONE...
2906 026674  042764  000001  000002          BIC     #1,2(R4)        ;...AND CLEAR GO (SET DONE).
2907 026702  000240                          240                     ; INTERRUPT SHOULD BE HELD OFF.
2908 026704  000403                          BR      2$
2909 026706                          1$:
     026706  104423  026614                  ERROR,  T107            ;; *** Q-BUS INTERRUPTS AT WRONG LEVEL ***
2910 026712  000407                          BR      3$
2911 026714  012713  026732          2$:     MOV     #3$,(R3)        ; CHANGE THE VECTOR.
2912 026720  106427  000000                  MTPS    #PRI0           ; LOWER CPU...
2913 026724  000240                          240                     ;...INTERRUTPT SHOULD COME IN.
2914 026726  104423  026614                  ERROR,  T107            ;; *** Q-BUS INTERRUPT NOT RECEIVED ***
2915 026732  005014                  3$:     CLR     (R4)            ; THAT'S ALL THERE IS TO IT.
2916 026734  012723  000112                  MOV     #QBXVEC+2,(R3)+ ; RESET Q-BUS VECTOR.
2917 026740  005013                          CLR     (R3)
2918 026742                          4$:
2919
2920                                 ;*************************************************************************
                                     .SBTTL      T110 -- TEST THE "BEVNT" INTERRUPT (REQUIRES Q-BUS EXERCISER)
                                     ;*************************************************************************
     026742  104420  000110          T110:   BEGINTEST, 110          ;; CHECK SWITCH REGISTER OPTIONS.
2921 026746  012706  003776                  MOV     #STACK,SP       ; SET STACK POINTER.
2922 026752  004767  006432                  CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
2923 026756  001433                          BEQ     4$              ; IF NOT, BYPASS TEST.
2924 026760  010004                          MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
2925                                 ;
2926                                 ; QBX CSR1<11> EMULATES A "BEVNT" INTERRUPT.
2927                                 ;
2928 026762  106427  000300                  MTPS    #PRI6           ; SET PRI 06
2929 026766  012767  027006  151104          MOV     #1$,BEVNT       ; SET BEVNT VECTOR.
2930 026774  005000                          CLR     R0
```

T110 -- TEST THE "BEVNT" INTERRUPT (REQUIRES Q-BUS EXERCISER)

```
 2931 026776  012714  004000                    MOV     #4000,(R4)      ; SET "BEVNT" REQUEST.
 2932 027002  077001                            SOB     R0,.            ; DELAY ABOUT 250 MSEC, INTERRUPT...
 2933 027004  000403                            BR      2$              ;...SHOULD BE MASKED AT THIS LEVEL.
 2934 027006                              1$:
      027006  104423  026742                    ERROR,  T110            ;; *** BEVNT INTERRUPT LEVEL INCORRECT ***
 2935 027012  000410                            BR      3$
 2936 027014  012767  027034  151056  2$:      MOV     #3$,BEVNT       ; OK, CHANGE VECTOR...
 2937 027022  106427  000240                    MTPS    #PRI5           ;...AND LOWER CPU TO PRI 5.
 2938 027026  077001                            SOB     R0,.            ; DELAY AGAIN, INT SHOULD COME IN.
 2939 027030  104423  026742                    ERROR,  T110            ;; *** BEVNT INTERRUPT NOT RECEIVED ***
 2940 027034  012767  000102  151036  3$:      MOV     #BEVNT+2,BEVNT  ; RESET VECTOR.
 2941 027042  106427  000000                    MTPS    #PRI0           ; LOWER CPU TO 0.
 2942
 2943 027046  004767  006004          4$:      CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
 2944                                                                  ;...RETURN AND FALL THRU IF NOT.
 2945                                    .SBTTL
```

SECTION 3 -- LOCAL RAM TESTS.

```
                                        .SBTTL  SECTION 3 -- LOCAL RAM TESTS.
                                
027052  012767  036373  007246  S3:     MOV     #SEG3,SEGN      ; SEGMENT 3 TEXT.
                                ;       MOV     #RAMADR,TEMP    ; SAVE RAM ADDRESS...                   ;JSD REV B
                                ;       MOV     #RAMSIZ,TEMP1   ;...AND SIZE.                           ;JSD REV B
027060  000401                          BR      .+4                                                     ;JSD REV B
027062  000000                  FPLUS:  .WORD                   ; FLAG (0=FALCON, 1=FALCON+)            ;JSD REV B
027064  005067  177772                  CLR     FPLUS           ; ASSUME FALCON; CLEAR FLAG             ;JSD REV B
027070  012767  160010  151710          MOV     #RAMAD1,TEMP    ; SET FALCON RAM ADDR                   ;JSD REV B
027076  012767  003764  151704          MOV     #RAMSZ1,TEMP1   ;...AND SIZE.                           ;JSD REV B
027104  012767  027200  150672          MOV     #11$,ERRVEC     ; SET TIMEOUT VECTOR                    ;JSD REV B
027112  012700  177600                  MOV     #177600,R0      ; START OF FALCON+ ODT SCRATCH          ;JSD REV B
027116  012701  000100                  MOV     #64.,R1         ; NUMBER OF WORDS                       ;JSD REV B
027122  005720                  10$:    TST     (R0)+           ; TEST LOCATION (IF PRESENT)            ;JSD REV B
027124  077102                          SOB     R1,10$          ;    (FALCON WILL TRAP)                 ;JSD REV B
027126  012767  000001  177726          MOV     #1,FPLUS        ; ELSE, SET FLAG FOR FALCON+            ;JSD REV B
027134  012767  100000  151644          MOV     #RAMAD2,TEMP    ; SET FALCON+ RAM ADDR                  ;JSD REV B
027142  012767  024000  151640          MOV     #RAMSZ2,TEMP1   ;...AND SIZE.                           ;JSD REV B
027150  005077  151632                  CLR     @TEMP           ; CHECK IF SOCKET B IS PRESENT          ;JSD REV B
027154  027727  151626  177777          CMP     @TEMP,#-1       ; IF CLEARED LOC IS NOT ALL 1'S         ;JSD REV B
027162  001006                          BNE     11$             ; THEN THERE IS RAM SPACE HERE          ;JSD REV B
027164  062767  040000  151614          ADD     #40000,TEMP     ; ELSE SKIP OVER SOCKET B               ;JSD REV B
027172  162767  020000  151610          SUB     #20000,TEMP1    ; REDUCE RAM SIZE (# WORDS)             ;JSD REV B
027200                          11$:                                                                    ;JSD REV B

                                ;*****************************************************************************
                                .SBTTL      T111 -- LOCAL RAM ADDRESS TEST
                                ;*****************************************************************************
027200  104420  000111          T111:   BEGINTEST, 111          ;; CHECK SWITCH REGISTER OPTIONS.
                                ;
                                ; WRITE EACH RAM LOCATION WITH ITS OWN ADDRESS, READ BACK AND VERIFY.
                                ; REPEAT A SECOND TIME USING BYTE-SWAPPED ADDRESS AS DATA.
                                ;
027204  012706  003776                  MOV     #STACK,SP
027210  012767  027312  150566          MOV     #7$,ERRVEC      ; SET TIME-OUT VECTOR.
027216  016700  151564                  MOV     TEMP,R0         ; GET RAM ADDRESS...
027222  016701  151562                  MOV     TEMP1,R1        ;...AND SIZE.
027226  010146                          MOV     R1,-(SP)        ; SAVE A COPY.
                                
027230  010010                  1$:     MOV     R0,(R0)         ; LOAD ADDRESSES (FROM BOTTOM UP).
027232  005720                          TST     (R0)+           ; BUMP
027234  077103                          SOB     R1,1$           ; LOOP 'TIL DONE.
                                
027236  011602                          MOV     (SP),R2         ; RECOVER SIZE K.
027240  014001                  2$:     MOV     -(R0),R1        ; READ 'EM BACK (FROM TOP DOWN).
027242  020100                          CMP     R1,R0           ; DATA = ADDRESS ??
027244  001402                          BEQ     3$              ; BR IF SO.
027246  104423  027200                  ERROR.  T111            ;; *** RAM DATA INCORRECT AT ADDRESS IN R0 ***
027252  077206                  3$:     SOB     R2,2$           ; LOOP 'TIL DONE.
                                
027254  011602                          MOV     (SP),R2         ; RECOVER SIZE K.
027256  010001                  4$:     MOV     R0,R1           ; ADDRESS => R1...
027260  000301                          SWAB    R1              ;...SWAP BYTES...
027262  010120                          MOV     R1,(R0)+        ;...AND LOAD SWAPPED ADDRESSES.
027264  077204                          SOB     R2,4$
                                
027266  011603                          MOV     (SP),R3         ; RECOVER SIZE K.
```

PC 27246 = RAM DATA INCORRECT AT ADDRESS IN R0.

```
3001 027270  014001                  5$:     MOV     -(R0),R1        ; GET DATA.
3002 027272  010002                          MOV     R0,R2
3003 027274  000302                          SWAB    R2
3004 027276  020102                          CMP     R1,R2           ; DATA = SWAPPED ADDRESS ??
3005 027300  001402                          BEQ     6$              ; BR IF SO.
3006 027302  104423  027200                  ERROR.  T111            ;; *** RAM DATA (SWAPPED) INCORRECT AT ADDRESS IN R0 ***
3007 027306  077310                  6$:     SOB     R3,5$
3008 027310  000402                          BR      8$              ; DONE, EXIT.
3009
3010 027312                          7$:
     027312  104423  027200                  ERROR.  T111            ;; *** BUS TIME-OUT -- RAM ADDRESS IN R0 ***
3011 027316                          8$:
3012
3013                                 ;*************************************************************************
                                     .SBTTL      T112 -- LOCAL RAM DATA TEST
                                     ;*************************************************************************
     027316  104420  000112        T112:   BEGINTEST, 112          ;; CHECK SWITCH REGISTER OPTIONS.
3014                                 ;
3015                                 ; FILL RAM WITH A BACKGROUND PATTERN OF 0'S.
3016                                 ; FLOAT AND VERIFY A 1 BIT THRU EACH WORD, AND WHEN DONE
3017                                 ; RESTORE THAT WORD TO THE BACKGROUND.
3018                                 ; WHEN ALL WORDS DONE, VERIFY THAT THE ENTIRE RAM STILL CONTAINS
3019                                 ; THE CORRECT BACKGROUND DATA.
3020                                 ;
3021                                 ; REPEAT A SECOND TIME WITH BACKGROUND 1, AND A FLOATING 0.
3022                                 ;
3023 027322  012706  003776                  MOV     #STACK,SP
3024 027326  012767  027502  150450          MOV     #10$,ERRVEC     ; SET TIME-OUT VECTOR.
3025 027334  005046                          CLR     -(SP)           ; BACKGROUND = 0'S FIRST...
3026 027336  000402                          SKP2
3027 027340  012746  177777          1$:     MOV     #-1,-(SP)       ;...THEN 1'S.
3028 027344  016700  151436                  MOV     TEMP,R0         ; SET RAM ADDRESS...
3029 027350  016701  151434                  MOV     TEMP1,R1        ;...AND SIZE.
3030 027354  011620                  2$:     MOV     (SP),(R0)+      ; FILL RAM WITH BACKGROUND DATA.
3031 027356  077102                          SOB     R1,2$
3032
3033 027360  016700  151422                  MOV     TEMP,R0         ; RESET ADDRESS...
3034 027364  016701  151420                  MOV     TEMP1,R1        ;...AND WORD COUNT...
3035 027370  011004                  3$:     MOV     (R0),R4         ;...AND EXERCISE 1ST/NEXT LOCATION.
3036 027372  020416                          CMP     R4,(SP)
3037 027374  001402                          BEQ     4$              ; BR IF BACKGROUND IS RIGHT.
3038 027376  104423  027370                  ERROR.  3$              ;; *** INITIAL BACKGROUND INCORRECT AT ADDRESS IN R0 ***
3039 027402  012702  000001          4$:     MOV     #1,R2
3040 027406  011603                          MOV     (SP),R3         ; COPY CURRENT BACKGROUND TO R3.
3041 027410  074203                          XOR     R2,R3           ; SET OR CLEAR BIT 0.
3042 027412  012702  000020                  MOV     #16.,R2         ; SET BIT COUNTER.
3043 027416  000402                          SKP2
3044 027420  006303                  5$:     ASL     R3              ; ROTATE DATA...
3045 027422  005503                           ADC     R3              ;...BIT 15 => BIT 0.
3046 027424  010310                  6$:     MOV     R3,(R0)         ; WRITE FLOATING DATA...
3047 027426  011004                          MOV     (R0),R4         ;...AND READ IT BACK.
3048 027430  020304                          CMP     R3,R4
3049 027432  001402                          BEQ     7$              ; BR IF ITS RIGHT.
3050 027434  104423  027424                  ERROR.  6$              ;; *** FLOATING DATA INCORRECT AT ADDRESS IN R0 ***
3051 027440  077211                  7$:     SOB     R2,5$           ; LOOP FOR ALL BITS.
3052 027442  011620                          MOV     (SP),(R0)+      ; THEN RESTORE, BUMP ADDRESS...
3053 027444  077127                          SOB     R1,3$           ;...AND LOOP FOR ALL WORDS.
```

```
PC 27434 = FLOATING DATA INCORRECT AT ADDRESS IN R0.

 3054
 3055 027446  016700  151334                MOV     TEMP,R0         ; NOW, RESET POINTERS...
 3056 027452  016701  151332                MOV     TEMP1,R1        ;...AND VERIFY THAT RESTORED...
 3057 027456  011004                8$:     MOV     (R0),R4         ;...BACKGROUND IS STILL CORRECT.
 3058 027460  020416                        CMP     R4,(SP)
 3059 027462  001402                        BEQ     9$              ; BR IF SO.
 3060 027464  104423  027456                ERROR.  8$              ;; *** BACKGROUND CHANGED AT ADDRESS IN R0 ***
 3061 027470  005720                9$:     TST     (R0)+           ; BUMP...
 3062 027472  077107                        SOB     R1,8$           ;...AND LOOP 'TIL DONE.
 3063
 3064 027474  005726                        TST     (SP)+           ; FINALLY, IF BACKGROUND IS 0...
 3065 027476  001720                        BEQ     1$              ;...LOOP ONCE MORE...
 3066 027500  000402                        BR      11$             ;...OTHERWISE, WE'RE ALL DONE.
 3067
 3068 027502                        10$:
      027502  104423  027316                ERROR.  T112            ;; *** BUS TIME-OUT -- RAM ADDRESS IN R0 ***
 3069 027506                        11$:
 3070
 3071                               ;*****************************************************************************
                                    .SBTTL      T113 -- OPTIONAL DMA TEST (REQUIRES Q-BUS EXERCISER)
                                    ;*****************************************************************************
      027506  104420  000113      T113:   BEGINTEST, 113          ;; CHECK SWITCH REGISTER OPTIONS.
 3072 027512  012706  003776                MOV     #STACK,SP
 3073 027516  004767  005666                CALL    QBXIN           ; Q-BUS EXERCISOR AVAILABLE ??
 3074 027522  001504                        BEQ     10$             ; IF NOT, BYPASS TEST.
 3075 027524  010004                        MOV     R0,R4           ; YES, CSR1 POINTER => R4 AND FALL THRU.
 3076                               ;
 3077                               ; TRANSFER EACH RAM WORD TO THE QBX AND BACK.
 3078                               ; VERIFY THAT RETURNED DATA MATCHES THAT WHICH WAS SENT.
 3079                               ;
 3080 027526  012767  027730  150250        MOV     #9$,ERRVEC      ; SET TRAP CATCHER.
 3081 027534  016701  151246                MOV     TEMP,R1         ; RAM ADDRESS POINTER...
 3082 027540  016702  151244                MOV     TEMP1,R2        ;...AND WORD COUNT.
 3083
 3084 027544  010111                2$:     MOV     R1,(R1)         ; SET 1ST/NEXT RAM WORD.
 3085 027546  012714  140407                MOV     #140407,(R4)    ; DATI (MEM => QBX) IN "HOG" MODE.
 3086 027552  010164  000004                MOV     R1,4(R4)        ; FROM RAM (R1) TO DATA REGISTER...
 3087 027556  012764  177774  000006        MOV     #-4,6(R4)       ;...4 TIMES.
 3088 027564  012764  000401  000002        MOV     #401,2(R4)      ; GO, INHIBIT BA+.
 3089 027572  005000                        CLR     R0
 3090 027574  105764  000002        3$:     TSTB    2(R4)
 3091 027600  100401                        BMI     33$             ; PROCEED WHEN DONE...
 3092 027602  077004                        SOB     R0,3$
 3093 027604  032764  040000  000002  33$:  BIT     #40000,2(R4)    ;...AND CHECK FOR NXM ERROR.
 3094 027612  001402                        BEQ     4$              ; BR IF OK.
 3095 027614  104423  027544                ERROR.  2$              ;; *** NXM ERROR IN QBX ON DATI (MEM TO QBX) ***
 3096 027620  026401  000010        4$:     CMP     10(R4),R1       ; DATA REGISTER CORRECT ??
 3097 027624  001402                        BEQ     5$
 3098 027626  104423  027544                ERROR.  2$              ;; *** DATA INCORRECT IN QBX ON DATI (MEM TO QBX) ***
 3099
 3100 027632  012714  000601        5$:     MOV     #000601,(R4)    ; DATO (QBX => MEM) IN "HOG" MODE.
 3101 027636  012764  001012  000004        MOV     #TEMP2,4(R4)    ; FROM DATA REGISTER TO TEMP2...
 3102 027644  012764  177774  000006        MOV     #-4,6(R4)       ;...4 TIMES.
 3103 027652  012764  000401  000002        MOV     #401,2(R4)      ; GO, INHIBIT BA+.
 3104 027660  005000                        CLR     R0
 3105 027662  105764  000002        6$:     TSTB    2(R4)
 3106 027666  100401                        BMI     66$             ; PROCEED WHEN DONE...
```

PC 27626 - DATA INCORRECT IN QBX ON DATI (MEM TO QBX).

```
3107 027670  077004                          SOB     R0,6$
3108 027672  032764  040000  000002  66$:    BIT     @#40000,2(R4)   ;...AND CHECK FOR NXM.
3109 027700  001402                          BEQ     7$              ; BR IF OK.
3110 027702  104423  027632                  ERROR.  5$              ;; *** NXM ERROR IN QBX ON DATO (QBX TO MEM) ***
3111 027706  026701  151100          7$:     CMP     TEMP2,R1        ; RETURNED DATA CORRECT ??
3112 027712  001402                          BEQ     8$
3113 027714  104423  027632                  ERROR.  5$              ;; *** DATA INCORRECT IN TEMP2 ON DATO (QBX TO MEM) ***
3114
3115 027720  005721                  8$:     TST     (R1)+           ; SET NEXT ADDRESS...
3116 027722  077270                          SOB     R2,2$           ;...AND LOOP 'TIL DONE.
3117 027724  005014                          CLR     (R4)            ; CLEAR QBX...
3118 027726  000402                          BR      10$             ;...AND EXIT.
3119
3120 027730                          9$:
     027730  104423  027506                  ERROR.  T113            ;; *** BUS TIME-OUT -- QBX (R4) OR RAM (R1) ***
3121
3122 027734  012767  000006  150042  10$:    MOV     @#ERRVEC+2,ERRVEC ; RESET ERROR VECTOR.
3123 027742  004767  005110                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
3124                                                                  ;...RETURN AND FALL THRU IF NOT.
3125                                  .SBTTL
```

SECTION 4 -- LOCAL PROM/RAM TESTS (EMPTY SOCKETS).

```
3127                                    .SBTTL  SECTION 4 -- LOCAL PROM/RAM TESTS (EMPTY SOCKETS).
3128
3129 027746  012767  036405  006352  S4:     MOV     #SEG4,SEGN      ; SEGMENT 4 TEXT.
3130                                  ;       MOV     #ROMADR,TEMP    ; SAVE ROM ADDRESS...           ;JSD REV B
3131                                  ;       MOV     #ROMSIZ,TEMP1   ;...AND SIZE.                   ;JSD REV B
3132 027754  012767  170000  151024          MOV     #ROMAD1,TEMP    ; ASSUME FALCON; SAVE ROM ADDR  ;JSD REV B
3133 027762  012767  002000  151020          MOV     #ROMSZ1,TEMP1   ;...AND SIZE                    ;JSD REV B
3134 027770  005767  177066                  TST     FPLUS           ; IS THIS FALCON+?               ;JSD REV B
3135 027774  001406                          BEQ     5$              ; NO - FALCON - READY TO START   ;JSD REV B
3136 027776  012767  164000  151002          MOV     #ROMAD2,TEMP    ; SAVE FALCON+ ROM ADDRESS       ;JSD REV B
3137 030004  012767  004000  150776          MOV     #ROMSZ2,TEMP1   ;...AND SIZE                    ;JSD REV B
3138 030012                          5$:                                                             ;JSD REV B
3139
3140                                    ;*****************************************************************
                                        .SBTTL      T114 -- LOCAL PROM/RAM ADDRESS TEST
                                        ;*****************************************************************
     030012  104420  000114           T114:   BEGINTEST, 114          ;; CHECK SWITCH REGISTER OPTIONS.
3141                                    ;
3142                                    ; PROM/RAM SOCKETS CONTAIN THE MACRO ODT PROM SET.
3143                                    ; ATTEMPT TO WRITE (123456) INTO EACH LOCATION.
3144                                    ; IF AN ADDRESS HAPPENS TO CONTAIN THAT VALUE, THEN WRITE AGAIN
3145                                    ; WITH THE COMPLIMENT OF THAT VALUE.
3146                                    ;
3147 030016  012706  003776                  MOV     #STACK,SP
3148 030022  012767  030074  147754          MOV     #3$,ERRVEC      ; SET TIME-OUT VECTOR.
3149 030030  016700  150752                  MOV     TEMP,R0         ; SET ADDRESS...
3150 030034  016701  150750                  MOV     TEMP1,R1        ;...AND SIZE.
3151
3152 030040  012702  123456          1$:     MOV     #123456,R2
3153 030044  010210                          MOV     R2,(R0)         ; TRY TO WRITE DATA (SHOULDN'T TRAP).
3154 030046  021002                          CMP     (R0),R2         ; DID IT WRITE ??
3155 030050  001006                          BNE     2$              ; BR IF NOT.
3156 030052  005102                          COM     R2              ; MAYBE...
3157 030054  010210                          MOV     R2,(R0)         ;...TRY COMPLIMENT.
3158 030056  021002                          CMP     (R0),R2         ; NOW NOW ??
3159 030060  001002                          BNE     2$              ; BR IF OK.
3160 030062  104423  030040                  ERROR,  1$              ;; *** WRITE TO ROM ALTERED DATA ????? ***
3161
3162 030066  005720                  2$:     TST     (R0)+           ; BUMP POINTER.
3163 030070  077115                          SOB     R1,1$           ; LOOP 'TIL DONE...
3164 030072  000402                          BR      4$              ;...AND PROCEED.
3165
3166 030074                          3$:
     030074  104423  030012                  ERROR,  T114            ;; *** BUS TIME-OUT -- ROM ADDRESS IN R0 ***
3167
3168 030100  012767  000006  147676  4$:     MOV     #ERRVEC+2,ERRVEC
3169 030106  004767  004744                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
3170                                                                  ;...RETURN AND FALL THRU IF NOT.
3171                                    .SBTTL
```

```
                                        .SBTTL  SECTIONS 5 AND 6 -- SERIAL LINES 1 AND 2 TESTS.
                                        ;
                                        ; DLART BIT ASSIGNMENTS.
                                        ;
            000001                      XMTBRK= 001
            000100                      XMTIE=  100
            000200                      XMTRDY= 200

            000100                      RCVIE=  000100
            000200                      RCVDUN= 000200
            004000                      RCVACT= 004000
            004000                      RCVBRK= 004000
            020000                      FRERR=  020000
            040000                      ORERR=  040000
            100000                      RCVERR= 100000
                                        ;
                                        ; THE FOLLOWING DLART TESTS ARE RUN TWICE PER PASS.
                                        ; FIRST, ON SLU1 (THE CONSOLE SLOT).
                                        ;
030112  012767  036417  006206  S5:     MOV     #SEG5,SEGN      ; SEGMENT 5 TEXT.
030120  012700  177560                  MOV     #SLU1,R0        ; 1ST UNIT CSR...
030124  012701  000060                  MOV     #SLU1V,R1       ;...AND VECTOR(S).
030130  012702  000200                  MOV     #SLU1P,R2       ; SLU1 INTERRUPTS AT LEVEL 4.
030134  012703  030300                  MOV     #BADU1,R3       ; TIME-OUT VECTOR.
030140  000413                          BR      DLSET           ; SET-UP POINTERS AND XCT.
                                        ;
                                        ; AND THEN ON SLU2.
                                        ;
030142  012767  036435  006156  S6:     MOV     #SEG6,SEGN      ; SEGMENT 6 TEXT.
030150  012700  176540                  MOV     #SLU2,R0        ; 2ND UNIT CSR...
030154  012701  000120                  MOV     #SLU2V,R1       ;...AND VECTOR.
030160  012702  000240                  MOV     #SLU2P,R2       ; SLU2 INTERRUPTS AT LEVEL 5.
030164  012703  030310                  MOV     #BADU2,R3       ; TIME-OUT VECTOR.

030170                          DLSET:
                                ;*******************************************************************
                                .SBTTL      T115 -- TEST SERIAL LINE REGISTER ADDRESSES
                                ;*******************************************************************
030170  104420  000115          T115:   BEGINTEST, 115          ;; CHECK SWITCH REGISTER OPTIONS.
030174  012706  003776                  MOV     #STACK,SP       ; INIT THE STACK.
030200  010367  147600                  MOV     R3,ERRVEC       ; SET BUS TIME-OUT TRAP CATCHER.
030204  010067  150634                  MOV     R0,RCSR         ; LOAD UP THE REGISTER POINTERS.
030210  005720                          TST     (R0)+           ; IF ANY OF THESE TRAP, WE'RE DEAD !!
030212  010067  150630                  MOV     R0,RBUF
030216  005720                          TST     (R0)+
030220  010067  150624                  MOV     R0,XCSR
030224  005720                          TST     (R0)+
030226  010067  150620                  MOV     R0,XBUF
030232  010167  150616                  MOV     R1,RVEC         ; SAVE RCVR VECTOR POINTER...
030236  012721  030320                  MOV     #BADRI,(R1)+    ;...AND INIT VECTOR.
030242  010221                          MOV     R2,(R1)+
030244  010167  150606                  MOV     R1,XVEC         ; SAVE XMTR VECTOR POINTER...
030250  012721  030330                  MOV     #BADTI,(R1)+    ;...AND INT VECTOR.
030254  010221                          MOV     R2,(R1)+
030256  017700  150566                  MOV     @XCSR,R0        ; NOW GET THE DEFAULT XMIT STATUS...
030262  042700  177705                  BIC     #^C72,R0        ;...AND STRIP THE BAUD RATE BITS.
030266  052700  000002                  BIS     #2,R0           ; INSURE "PBRE" IS SET...
```

F6

T115 -- TEST SERIAL LINE REGISTER ADDRESSES

```
 3227 030272  010067  150542                    MOV     R0,DFPBR        ;...AND SAVE AS THE DEFAULT BAUD RATE.
 3228 030276  000432                            BR      DLTSTS          ; START 'EM UP.
 3229                                   ;
 3230                                   ; COME HERE IF BUS TIME-OUT OR UNEXPECTED INTERRUPT.
 3231                                   ;
 3232 030300                            BADU1:
      030300  104423  030112                    ERROR,  S5              ;; *** BUS TIME-OUT ON SLU1 ADDRESS ***
 3233 030304  000167  003270                    JMP     DLDUN
 3234 030310                            BADU2:
      030310  104423  030142                    ERROR,  S6              ;; *** BUS TIME-OUT ON SLU2 ADDRESS ***
 3235 030314  000167  003260                    JMP     DLDUN
 3236 030320                            BADRI:
      030320  104423  030364                    ERROR,  DLTSTS          ;; *** UNEXPECTED RCVR INTERRUPT. ***
 3237 030324  000167  003250                    JMP     DLDUN
 3238 030330                            BADTI:
      030330  104423  030364                    ERROR,  DLTSTS          ;; *** UNEXPECTED XMTR INTERRUPT. ***
 3239 030334  000167  003240                    JMP     DLDUN
 3240                                   ;
 3241                                   ; FLOATING 1 AND 0 DATA TABLE, USED IN VARIOUS TESTS.
 3242                                   ;
 3243 030340     015     012            SYNC:   .BYTE   015, 012                        ; SYNC PAIR.
 3244 030342     001     002     004    FLT1:   .BYTE   001, 002, 004, 010, 020, 040, 100, 200
      030345     010     020     040
      030350     100     200
 3245 030352     376     375     373    FLT0:   .BYTE   376, 375, 373, 367, 357, 337, 277, 177
      030355     367     357     337
      030360     277     177
 3246 030362  000000                            .WORD   0                               ; TERMINATOR.
 3247
 3248 030364                            DLTSTS:
 3249                                   ;*************************************************************************
                                        .SBTTL      T116 -- TEST THAT BUS-RESET CLEARS THE RIGHT BITS
                                        ;*************************************************************************
      030364  104420  000116            T116:   BEGINTEST, 116          ;; CHECK SWITCH REGISTER OPTIONS.
 3250                                   ;
 3251                                   ; SET ALL WRITABLE BITS AND RESET.  CHECK THAT ONLY
 3252                                   ; RCV-IE, XMIT-IE, MAINT, AND XMIT-BRK GET CLEARED.
 3253                                   ;
 3254 030370  106427  000300                    MTPS    #PRI6           ; RAISE CPU
 3255 030374  012777  000177  150446            MOV     #177,@XCSR      ; SET WRITABLE BITS IN XCSR...
 3256 030402  012777  000377  150442            MOV     #377,@XBUF      ;...AND XBUF...
 3257 030410  012777  000100  150426            MOV     #100,@RCSR      ;...AND RCSR.
 3258 030416  000240  000240                    240,240                 ; IN CASE WE NEED A DELAY.
 3259 030422  000005                            RESET
 3260 030424  000240  000240                    240,240
 3261 030430  017700  150410                    MOV     @RCSR,R0        ; GET RCSR...
 3262 030434  017701  150406                    MOV     @RBUF,R1        ;...RBUF (DUMMY READ)...
 3263 030440  017702  150404                    MOV     @XCSR,R2        ;...XCSR...
 3264 030444  017703  150402                    MOV     @XBUF,R3        ;...AND XBUF.
 3265 030450  016777  150364  150372            MOV     DFPBR,@XCSR     ; RESTORE DEFAULT XMIT...
 3266 030456  005077  150362                    CLR     @RCSR           ;...AND RCVR STATUS.
 3267 030462  032700  000100                    BIT     #100,R0         ; NOW, DID RCV-IE CLEAR ??
 3268 030466  001402                            BEQ     1$
 3269 030470  104423  030364                    ERROR,  T116            ;; *** BUS-RESET DIDN'T CLEAR RCV-IE BIT IN RCSR ***
 3270 030474  032702  000105            1$:     BIT     #105,R2         ; DID XIE, MAINT AND XBRK CLEAR ??
 3271 030500  001402                            BEQ     2$
 3272 030502  104423  030364                    ERROR,  T116            ;; *** BUS-RESET DIDN'T CLEAR XIE, MAINT, AND/OR XBRK ***
```

```
PC 30502 - BUS-RESET DIDN'T CLEAR XIE, MAINT, AND/OR XBRK.

3273 030506  120227  000272          2$:     CMPB    R2,#272         ; ALL OTHERS SET (INCLUDING XMTRDY) ??
3274 030512  001402                          BEQ     3$
3275 030514  104423  030364                  ERROR,  T116            ;; *** BUS-RESET CLEARED WRONG BITS IN XCSR ***
3276 030520  120327  177777          3$:     CMPB    R3,#-1          ; ALL BITS STILL IN XBUF ??
3277 030524  001402                          BEQ     4$
3278 030526  104423  030364                  ERROR,  T116            ;; *** BUS-RESET CLEARED BITS IN XBUF ***
3279 030532  106427  000000          4$:     MTPS    #PRIO           ; LOWER THE CPU.
3280
3281                                         ;*************************************************************************
                                             .SBTTL      T117 -- TEST THAT ALL UNDEFINED BITS ARE ZERO
                                             ;*************************************************************************
     030536  104420  000117          T117:   BEGINTEST, 117          ;; CHECK SWITCH REGISTER OPTIONS.
3282 030542  017700  150276                  MOV     @RCSR,R0        ; GET REGISTERS AGAIN.
3283 030546  017701  150274                  MOV     @RBUF,R1
3284 030552  017702  150272                  MOV     @XCSR,R2
3285 030556  017703  150270                  MOV     @XBUF,R3
3286 030562  032700  173477                  BIT     #173477,R0      ; RCSR, UNDEFINED BITS SET ??
3287 030566  001402                          BEQ     1$
3288 030570  104423  030536                  ERROR,  T117            ;; *** UNDEFINED BITS SET IN RCVR STATUS ***
3289 030574  032701  013400          1$:     BIT     #13400,R1       ; RBUF, UNDEFINED BITS SET ??
3290 030600  001402                          BEQ     2$
3291 030602  104423  030536                  ERROR,  T117            ;; *** UNDEFINED BITS SET IN RCVR DATA BUFFER ***
3292 030606  032702  177400          2$:     BIT     #177400,R2      ; XCSR, UNDEFINED BITS SET ??
3293 030612  001402                          BEQ     3$
3294 030614  104423  030536                  ERROR,  T117            ;; *** UNDEFINED BITS SET IN XMTR STATUS ***
3295 030620  032703  177400          3$:     BIT     #177400,R3      ; XBUF, UNDEFINED BITS SET ??
3296 030624  001402                          BEQ     4$
3297 030626  104423  030536                  ERROR,  T117            ;; *** UNDEFINED BITS SET IN XMTR DATA BUFFER ***
3298 030632  105777  150212          4$:     TSTB    @XCSR           ; XMTR READY ??
3299 030636  100402                          BMI     5$
3300 030640  104423  030536                  ERROR,  T117            ;; *** XMITTER NOT READY ***
3301                                 ;       .IIF P2, .PRINT .       ; *** DEBUG, NOP ***            ;JSD REV B
3302 030644  000402                  5$:     SKP2
3303 030646  000167  002726                  JMP     DLDUN
3304                                         .SBTTL
```

```
                                        .SBTTL  TRANSMITTERS
                                        XMTRS:                          ; TRANSMITTER TEST SECTION.
                                        ;*******************************************************************************
                                        .SBTTL      T120 -- FLOAT 1 AND 0 THRU THE XMIT DATA BUFFER
                                        ;*******************************************************************************
     030652  104420  000120             T120:   BEGINTEST, 120          ;; CHECK SWITCH REGISTER OPTIONS.
3309 030656  012700  030342                     MOV     #FLT1,R0        ; TABLE POINTER.
3310 030662  112001                             MOVB    (R0)+,R1        ; GET 1ST CHARACTER.
3311 030664  010177  150162             1$:     MOV     R1,@XBUF        ;LOAD DATA INTO "XMIT DATA BUF"...
3312 030670  017702  150156                     MOV     @XBUF,R2        ;...AND READ IT BACK.
3313 030674  120102                             CMPB    R1,R2
3314 030676  001402                             BEQ     2$
3315 030700  104423  030664                     ERROR,  1$              ;; *** FLOATING 1 OR 0 INCORRECT IN XBUF ***
3316 030704  112001                     2$:     MOVB    (R0)+,R1        ; GET NEXT CHAR...
3317 030706  001366                             BNE     1$              ;...AND LOOP 'TIL DONE.
3318
3319                                    ;*******************************************************************************
                                        .SBTTL      T121 -- TEST THAT "XMIT-BRK" CAN BE SET AND CLEARED
                                        ;*******************************************************************************
     030710  104420  000121             T121:   BEGINTEST, 121          ;; CHECK SWITCH REGISTER OPTIONS.
3320 030714  106427  000340                     MTPS    #PRI7           ; RAISE CPU.
3321 030720  052777  000001  150122             BIS     #1,@XCSR        ;SET "XMIT-BRK" BIT...
3322 030726  017700  150116                     MOV     @XCSR,R0        ;...AND SAVE IN R0.
3323 030732  042777  000001  150110             BIC     #1,@XCSR        ; CLEAR IT...
3324 030740  017701  150104                     MOV     @XCSR,R1        ;...AND SAVE IN R1.
3325 030744  016777  150070  150076             MOV     DFPBR,@XCSR     ; RESTORE DEFAULT XMTR.
3326 030752  106427  000000                     MTPS    #PRI0           ; AND PRIORITY.
3327 030756  032700  000001                     BIT     #1,R0
3328 030762  001002                             BNE     1$              ;BR IF IT SET
3329 030764  104423  030710                     ERROR,  T121            ;; *** XMIT-BRK BIT FAILED TO SET ***
3330 030770  032701  000001             1$:     BIT     #1,R1
3331 030774  001402                             BEQ     2$              ;BR IF IT CLEARED
3332 030776  104423  030710                     ERROR,  T121            ;; *** XMIT-BRK BIT FAILED TO CLEAR ***
3333 031002                             2$:
3334
3335                                    ;*******************************************************************************
                                        .SBTTL      T122 -- TEST THAT "XMIT-MAINT" CAN BE SET AND CLEARED
                                        ;*******************************************************************************
     031002  104420  000122             T122:   BEGINTEST, 122          ;; CHECK SWITCH REGISTER OPTIONS.
3336 031006  052777  000004  150034             BIS     #4,@XCSR        ;SET "XMIT-MAINT" BIT...
3337 031014  017700  150030                     MOV     @XCSR,R0        ;...AND SAVE IN R0.
3338 031020  042777  000004  150022             BIC     #4,@XCSR        ; CLEAR IT...
3339 031026  017701  150016                     MOV     @XCSR,R1        ;...AND SAVE IN R1.
3340 031032  016777  150002  150010             MOV     DFPBR,@XCSR
3341 031040  032700  000004                     BIT     #4,R0
3342 031044  001002                             BNE     1$              ;BR IF IT SET
3343 031046  104423  031002                     ERROR,  T122            ;; *** XMIT-MAINT BIT FAILED TO SET ***
3344 031052  032701  000004             1$:     BIT     #4,R1
3345 031056  001402                             BEQ     2$              ;BR IF IT CLEARED
3346 031060  104423  031002                     ERROR,  T122            ;; *** XMIT-MAINT BIT FAILED TO CLEAR ***
3347 031064                             2$:
3348
3349                                    ;*******************************************************************************
                                        .SBTTL      T123 -- TEST THAT "XMIT-PBRE" CAN BE SET AND CLEARED
                                        ;*******************************************************************************
     031064  104420  000123             T123:   BEGINTEST, 123          ;; CHECK SWITCH REGISTER OPTIONS.
3350 031070  052777  000002  147752             BIS     #2,@XCSR        ;SET "XMIT-PBRE" BIT...
```

T123 -- TEST THAT "XMIT-PBRE" CAN BE SET AND CLEARED

```
 3351 031076  017700  147746                MOV     @XCSR,R0        ;...AND SAVE IN R0.
 3352 031102  042777  000002  147740        BIC     #2,@XCSR        ; CLEAR IT...
 3353 031110  017701  147734                MOV     @XCSR,R1        ;...AND SAVE IN R1.
 3354 031114  016777  147720  147726        MOV     DFPBR,@XCSR
 3355 031122  032700  000002                BIT     #2,R0
 3356 031126  001002                        BNE     1$              ;BR IF IT SET
 3357 031130  104423  031064                ERROR.  T123            ;; *** XMIT-PBRE BIT FAILED TO SET ***
 3358 031134  032701  000002          1$:   BIT     #2,R1
 3359 031140  001402                        BEQ     2$              ;BR IF IT CLEARED
 3360 031142  104423  031064                ERROR.  T123            ;; *** XMIT-PBRE BIT FAILED TO CLEAR ***
 3361 031146                          2$:
 3362
 3363                                ;*****************************************************************************
                                     .SBTTL      T124 -- TEST THAT "XMIT-PBR0" CAN BE SET AND CLEARED
                                     ;*****************************************************************************
      031146  104420  000124        T124:   BEGINTEST, 124          ;; CHECK SWITCH REGISTER OPTIONS.
 3364 031152  052777  000010  147670        BIS     #10,@XCSR       ;SET "XMIT-PBR0" BIT...
 3365 031160  017700  147664                MOV     @XCSR,R0        ;...AND SAVE IN R0.
 3366 031164  042777  000010  147656        BIC     #10,@XCSR       ; CLEAR IT...
 3367 031172  017701  147652                MOV     @XCSR,R1        ;...AND SAVE IN R1.
 3368 031176  016777  147636  147644        MOV     DFPBR,@XCSR
 3369 031204  032700  000010                BIT     #10,R0
 3370 031210  001002                        BNE     1$              ;BR IF IT SET
 3371 031212  104423  031146                ERROR.  T124            ;; *** XMIT-PBR0 BIT FAILED TO SET ***
 3372 031216  032701  000010          1$:   BIT     #10,R1
 3373 031222  001402                        BEQ     2$              ;BR IF IT CLEARED
 3374 031224  104423  031146                ERROR.  T124            ;; *** XMIT-PBR0 BIT FAILED TO CLEAR ***
 3375 031230                          2$:
 3376
 3377                                ;*****************************************************************************
                                     .SBTTL      T125 -- TEST THAT "XMIT-PBR1" CAN BE SET AND CLEARED
                                     ;*****************************************************************************
      031230  104420  000125        T125:   BEGINTEST, 125          ;; CHECK SWITCH REGISTER OPTIONS.
 3378 031234  052777  000020  147606        BIS     #20,@XCSR       ;SET "XMIT-PBR1" BIT...
 3379 031242  017700  147602                MOV     @XCSR,R0        ;...AND SAVE IN R0.
 3380 031246  042777  000020  147574        BIC     #20,@XCSR       ; CLEAR IT...
 3381 031254  017701  147570                MOV     @XCSR,R1        ;...AND SAVE IN R1.
 3382 031260  016777  147554  147562        MOV     DFPBR,@XCSR
 3383 031266  032700  000020                BIT     #20,R0
 3384 031272  001002                        BNE     1$              ;BR IF IT SET
 3385 031274  104423  031230                ERROR.  T125            ;; *** XMIT-PBR1 BIT FAILED TO SET ***
 3386 031300  032701  000020          1$:   BIT     #20,R1
 3387 031304  001402                        BEQ     2$              ;BR IF IT CLEARED
 3388 031306  104423  031230                ERROR.  T125            ;; *** XMIT-PBR1 BIT FAILED TO CLEAR ***
 3389 031312                          2$:
 3390
 3391                                ;*****************************************************************************
                                     .SBTTL      T126 -- TEST THAT "XMIT-PBR2" CAN BE SET AND CLEARED
                                     ;*****************************************************************************
      031312  104420  000126        T126:   BEGINTEST, 126          ;; CHECK SWITCH REGISTER OPTIONS.
 3392 031316  052777  000040  147524        BIS     #40,@XCSR       ;SET "XMIT-PBR2" BIT...
 3393 031324  017700  147520                MOV     @XCSR,R0        ;...AND SAVE IN R0.
 3394 031330  042777  000040  147512        BIC     #40,@XCSR       ; CLEAR IT...
 3395 031336  017701  147506                MOV     @XCSR,R1        ;...AND SAVE IN R1.
 3396 031342  016777  147472  147500        MOV     DFPBR,@XCSR
 3397 031350  032700  000040                BIT     #40,R0
 3398 031354  001002                        BNE     1$              ;BR IF IT SET
```

PC 31356 = XMIT-PBR2 BIT FAILED TO SET.

```
 3399 031356  104423  031312                ERROR,  T126            ;; *** XMIT-PBR2 BIT FAILED TO SET ***
 3400 031362  032701  000040          1$:   BIT     #40,R1
 3401 031366  001402                       BEQ     2$              ;BR IF IT CLEARED
 3402 031370  104423  031312                ERROR,  T126            ;; *** XMIT-PBR2 BIT FAILED TO CLEAR ***
 3403 031374                          2$:
 3404
 3405                                 ;*************************************************************************
                                      .SBTTL      T127 -- TEST THAT "XMIT-IE" CAN BE SET AND CLEARED
                                      ;*************************************************************************
      031374  104420  000127         T127:  BEGINTEST, 127           ;; CHECK SWITCH REGISTER OPTIONS.
 3406 031400  106427  000300                MTPS    #PRI6           ;RAISE CPU PRIORITY
 3407 031404  052777  000100  147436        BIS     #100,@XCSR      ;SET "XMIT-IE" BIT...
 3408 031412  017700  147432                MOV     @XCSR,R0        ;...AND SAVE IN R0.
 3409 031416  042777  000100  147424        BIC     #100,@XCSR      ; CLEAR IT...
 3410 031424  017701  147420                MOV     @XCSR,R1        ;...AND SAVE IN R1.
 3411 031430  016777  147404  147412        MOV     DFPBR,@XCSR     ; RESTORE XMTR...
 3412 031436  106427  000000                MTPS    #PRI0           ;...AND CPU TOO.
 3413 031442  032700  000100                BIT     #100,R0
 3414 031446  001002                       BNE     1$              ;BR IF IT SET
 3415 031450  104423  031374                ERROR,  T127            ;; *** XMIT-IE BIT FAILED TO SET ***
 3416 031454  032701  000100          1$:   BIT     #100,R1
 3417 031460  001402                       BEQ     2$              ; BR IF IT CLEARED.
 3418 031462  104423  031374                ERROR,  T127            ;; *** XMIT-IE BIT FAILED TO CLEAR ***
 3419 031466                          2$:
 3420
 3421                                 ;*************************************************************************
                                      .SBTTL      T130 -- TEST THAT "XMIT-RDY" CAN BE SET AND CLEARED
                                      ;*************************************************************************
      031466  104420  000130         T130:  BEGINTEST, 130           ;; CHECK SWITCH REGISTER OPTIONS.
 3422 031472  016777  147342  147350        MOV     DFPBR,@XCSR
 3423 031500  005001                       CLR     R1              ; DELAY COUNT.
 3424 031502  005002                       CLR     R2              ; DITTO
 3425 031504  105777  147340          1$:   TSTB    @XCSR           ;WAIT FOR "XMIT-RDY" TO SET
 3426 031510  100403                       BMI     2$
 3427 031512  077104                       SOB     R1,1$           ;KEEP TRYING.
 3428 031514  104423  031466                ERROR,  T130            ;; *** XMIT-RDY FAILED TO SET ***
 3429 031520  012777  000000  147324  2$:   MOV     #0,@XCBUF       ;TRANSMIT A NULL CHARACTER
 3430 031526  017700  147316                MOV     @XCSR,R0        ;READ XMIT STATUS REG
 3431 031532  032700  000200                BIT     #XMTRDY,R0      ; READY SHOULD BE GONE.
 3432 031536  001402                       BEQ     3$              ; BR IF SO.
 3433 031540  104423  031466                ERROR,  T130            ;; *** XMIT-RDY BIT FAILED TO CLEAR ***
 3434 031544  105777  147300          3$:   TSTB    @XCSR           ;WAIT FOR "XMIT-RDY" TO SET
 3435 031550  100403                       BMI     4$
 3436 031552  077204                       SOB     R2,3$           ; KEEP TRYING.
 3437 031554  104423  031466                ERROR,  T130            ;; *** XMIT-RDY FAILED TO SET AFTER XMITTING CHARACTER ***
 3438 031560                          4$:
 3439
 3440                                 ;*************************** *********************************************
                                      .SBTTL      T131 -- TEST THAT WE CAN INTERRUPT ON "XMT-RDY"
                                      ;*************************************************************************
      031560  104420  000131         T131:  BEGINTEST, 131           ;; CHECK SWITCH REGISTER OPTIONS.
 3441 031564  016777  147250  147256        MOV     DFPBR,@XCSR
 3442 031572  016701  147260                MOV     XVEC,R1         ; GET VECTOR POINTER.
 3443 031576  106461  000002                MTPS    2(R1)           ; RAISE CPU TO XMTR LEVEL.
 3444 031602  012711  031626                MOV     #2$,(R1)        ; SET XMITTER VECTOR.
 3445 031606  105777  147236          1$:   TSTB    @XCSR           ;WAIT FOR "XMIT-RDY"
 3446 031612  100375                       BPL     1$
```

T131 -- TEST THAT WE CAN INTERRUPT ON "XMT-RDY"

```
3447 031614  052777  000100  147226          BIS     #100,@XCSR      ;SET ENABLE "XMIT-IE" BIT
3448 031622  000240                          240                     ; INTERRUPT SHOULD BE HELD OFF...
3449 031624  000406                          BR      3$              ;...BR IF SO.
3450 031626  022626                  2$:     CMP     (SP)+,(SP)+     ; IT WASN'T.
3451 031630  042777  000100  147212          BIC     #100,@XCSR      ; REMOVE "XMIT-IE"
3452 031636  104423  031560                  ERROR.  T131            ;; *** XMT-RDY INTERRUPT AT WRONG LEVEL (TOO HIGH) ***
3453
3454 031642  012711  031670          3$:     MOV     #4$,(R1)        ;CHANGE THE VECTOR.
3455 031646  106427  000000                  MTPS    #PRIO           ;LOWER CPU PRIORITY
3456 031652  000240                          240                     ; INTERRUPT SHOULD HAVE COME IN.
3457 031654  042777  000100  147166          BIC     #100,@XCSR      ; IT DIDN'T, REMOVE "XMIT-IE" BIT
3458 031662  104423  031560                  ERROR.  T131            ;; *** XMT-RDY INTERRUPT NOT RECEIVED AT CPU LEVEL 0 ***
3459 031666  000415                          BR      6$
3460                                 ;
3461                                 ; NOW, ON INTERRUPT, WE'RE HERE AT CPU LEVEL 4 (5 IF SLU2).
3462                                 ; LOWER CPU AGAIN AND CHECK THAT WE DON'T GET ANOTHER INTERRUPT.
3463                                 ;
3464 031670  022626                  4$:     CMP     (SP)+,(SP)+
3465 031672  012711  031706                  MOV     #5$,(R1)        ; CHANGE THE VECTOR.
3466 031676  106427  000000                  MTPS    #PRIO           ;LOWER THE CPU PRIORITY AGAIN.
3467 031702  000240                          240                     ; SHOULDN'T GET ANOTHER...
3468 031704  000406                          BR      6$              ;...OK, EXIT.
3469 031706  022626                  5$:     CMP     (SP)+,(SP)+     ; 2ND INTERRUPT -- BAD NEWS.
3470 031710  042777  000100  147132          BIC     #100,@XCSR      ;REMOVE "XMIT-IE"
3471 031716  104423  031560                  ERROR.  T131            ;; *** XMT INTERRUPT ACK FAILED TO REMOVE THE REQUEST ***
3472
3473 031722  042777  000100  147120  6$:     BIC     #100,@XCSR      ; BE SURE THE ENABLE IS REMOVED.
3474 031730  012711  030330                  MOV     #BADTI,(R1)     ;RESET FALSE INTR VECTOR
3475
3476                                 ;*************************************************************************
                                     .SBTTL      T132 -- TEST THAT EACH XMIT BAUD RATE SETS A DIFFERENT SPEED
                                     ;*************************************************************************
     031734  104420  000132        T132:   BEGINTEST, 132          ;; CHECK SWITCH REGISTER OPTIONS.
3477                                 ;
3478                                 ; FIRST DETERMINE THE RELATIVE SPEED OF EACH BAUD RATE.
3479                                 ; TRANSMIT A CHARACTER AND COUNT CPU BUS TRANSACTION CYCLES UNTIL
3480                                 ; TRANSMISSION IS COMPLETE (XMT-RDY ASSERTS).
3481                                 ; SAVE THE OBSERVED CYCLE COUNT IN THE SPEED TABLE.
3482                                 ; IF NO FLAG BY THE TIME THE COUNT OVERFLOWS (65536. CYCLES), SAVE
3483                                 ; THE ZERO, WHICH IMPLIES XMT RDY NEVER SET AT THAT SPEED.
3484                                 ;
3485                                 ; NOTE:  ALL FOLLOWING RECEIVER TESTS WILL USE THE SPEED VALUES
3486                                 ; OBSERVED HERE AS LOOP CONTROL COUNTERS.
3487                                 ; NOTE:NOTE:  IF THERE'S A TTY ATTACHED, YOU'LL SEE SOME GARBAGE
3488                                 ; CHARACTERS ON THE SCREEN.
3489                                 ;
3490 031740  012700  000002                  MOV     #2,R0           ; START WITH SPEED 0 (300 BAUD).
3491 031744  012701  001016                  MOV     #SPD0,R1        ; TABLE POINTER.
3492 031750  005002                  1$:     CLR     R2              ; CLEAR THE ACCUMULATOR.
3493 031752  010077  147072                  MOV     R0,@XCSR        ; SET PBR BITS.
3494 031756  012777  000000  147066          MOV     #0,@XBUF        ; XMIT A CHAR.
3495 031764  105777  147060          2$:     TSTB    @XCSR           ; ON READY, CHAR HAS BEEN COPIED...
3496 031770  100375                          BPL     2$              ;...INTO THE SERIAL OUTPUT REGISTER.
3497 031772  012777  000000  147052          MOV     #0,@XBUF        ; XMIT AGAIN AND MEASURE THE TIME...
3498                                                                 ;...UNTIL READY RETURNS WHICH IS THE...
3499                                                                 ;...SERIAL OUTPUT TIME.
3500 032000  105777  147044          3$:     TSTB    @XCSR           ; [ 4 ]
```

T132 -- TEST THAT EACH XMIT BAUD RATE SETS A DIFFERENT SPEED

```
3501 032004  100403                        BMI     4$              ; [ 1 ] WE'RE DONE ON XMT-RDY.
3502 032006  062702  000010                ADD     #8.,R2          ; [ 2 ] COUNT 8 CYCLES/LOOP...
3503 032012  001372                        BNE     3$              ; [ 1 ] ...UNITL DONE (OR OVERFLOW).
3504 032014  010221                 4$:    MOV     R2,(R1)+        ; OK, R2 = TX TIME IN BUS CYCLES...
3505                                                               ;...AT APPROX 3 USEC/COUNT...
3506                                                               ;...SAVE IT IN THE SPEED TABLE.
3507 032016  020067  147016                CMP     R0,DFPBR        ; CURRENT RATE = DEFAULT ???
3508 032022  001002                        BNE     5$              ; SKIP IF NOT.
3509 032024  010267  147012                MOV     R2,DFSPD        ; YES, SAVE AS DEFAULT SPEED.
3510 032030  062700  000010          5$:   ADD     #10,R0          ; INCREMENT PBR BITS.
3511 032034  020027  000072                CMP     R0,#72
3512 032040  101743                        BLOS    1$              ; LOOP FOR ALL 8 SETTINGS.
3513 032042  016777  146772  147000        MOV     DFPBR,@XCSR     ; DONE, RESTORE DEFAULT XMTTR...
3514                                                               ;...AND FALL THRU.
3515                                ;
3516                                ; NOW THE RELATIVE SPEED AT EACH SETTING HAS BEEN DETERMINED.
3517                                ; CHECK THAT THE DIFFERENTIAL OF EACH ONE TO THE NEXT LOWER
3518                                ; IS 2:1 (MORE OR LESS).  THIS IS A GROSS SPEED CHECK SINCE
3519                                ; WE CAN'T RELY ON A REAL CLOCK FOR TIMING.
3520                                ;
3521                                ; DIFFERENTIAL IS CALCULATED BY:
3522                                ;       (LOWER/2)-HIGHER = 0 +/- 25.0% OF THE LOWER.
3523                                ;
3524                                ; ON ERROR, R0 DISPLAYS THE HIGHER OF THE TWO RATES UNDER TEST.
3525                                ;
3526 032050  012700  000072                MOV     #72,R0          ; WORK FROM THE HIGHEST DOWN.
3527 032054  012701  001034                MOV     #SPD7,R1        ; TABLE POINTER.
3528 032060  011102                 10$:   MOV     (R1),R2         ; GET HIGHER (SMALLER) VALUE.
3529 032062  001002                        BNE     11$
3530 032064  104423  031734                ERROR.  T132            ;!! *** XMT-RDY NEVER SET AT PBR IN R0 ***
3531 032070  020127  001016          11$:  CMP     R1,#SPD0        ; IF PBR 0, THERE'S NOTHING LOWER...
3532 032074  001421                        BEQ     14$             ;...SO JUST QUIT.
3533
3534 032076  014103                        MOV     -(R1),R3        ; GET NEXT LOWER (LARGER) VALUE.
3535 032100  010304                        MOV     R3,R4           ; SAVE A COPY FOR THE TOLERANCE.
3536 032102  006203                        ASR     R3              ; NOW, 1/2 THE LARGER...
3537 032104  160203                        SUB     R2,R3           ;...MINUS THE SMALLER = DELTA.
3538 032106  100001                        BPL     12$
3539 032110  005403                        NEG     R3              ; MAKE ABSOLUTE IF NECESSARY.
3540
3541 032112  006204                 12$:   ASR     R4
3542 032114  006204                         ASR    R4              ; SET ALLOWABLE TOLERANCE...
3543 032116  000240                          NOP                   ;... = 25.0% OF THE LARGER.
3544 032120  000240                           NOP
3545 032122  020304                        CMP     R3,R4           ; DELTA <= 25.0% ???
3546 032124  003402                        BLE     13$             ; BR IF SO.
3547 032126  104423  031734                ERROR.  T132            ;!! *** SPEED DIFFERENTIAL ERROR ***
3548 032132  162700  000010          13$:  SUB     #10,R0          ; DEC RATE COUNT...
3549 032136  100350                        BPL     10$             ;...AND LOOP 'TIL DONE.
3550 032140                         14$:
3551                                .SBTTL
```

```
3553                                    .SBTTL  RECEIVERS
3554 032140                             RCVRS:                          ; RECEIVER TEST SECTION.
3555                                    ;*****************************************************************************
                                        .SBTTL      T133 -- TEST THAT "RCV-IE" CAN BE SET AND CLEARED
                                        ;*****************************************************************************
     032140  104420  000133           T133:   BEGINTEST, 133          ;; CHECK SWITCH REGISTER OPTIONS.
3556 032144  106427  000300                   MTPS    #PRI6           ;RAISE CPU PRIORITY
3557 032150  012777  000100  146666           MOV     #100,@RCSR      ;SET "RCV-IE" BIT...
3558 032156  017700  146662                   MOV     @RCSR,R0        ;...AND SAVE IN R0.
3559 032162  005077  146656                   CLR     @RCSR           ; CLEAR IT...
3560 032166  017701  146652                   MOV     @RCSR,R1        ;...AND SAVE IN R1.
3561 032172  032700  000100                   BIT     #100,R0
3562 032176  001002                           BNE     1$              ; BR IF IT SET.
3563 032200  104423  032140                   ERROR,  T133            ;; *** RCV-IE BIT FAILED TO SET ***
3564 032204  032701  000100           1$:     BIT     #100,R1
3565 032210  001402                           BEQ     2$              ; BR IF IT CLEARED.
3566 032212  104423  032140                   ERROR,  T133            ;; *** RCV-IE BIT FAILED TO CLEAR ***
3567 032216  106427  000000           2$:     MTPS    #PRI0
3568
3569                                    ;*****************************************************************************
                                        .SBTTL      T134 -- TEST THAT "RCV-ACT" AND "RCV-DONE" CAN SET AND CLEAR
                                        ;*****************************************************************************
     032222  104420  000134           T134:   BEGINTEST, 134          ;; CHECK SWITCH REGISTER OPTIONS.
3570                                    ;
3571                                    ;THE TEST ENSURES THAT:
3572                                    ;       1. THE INTERNAL "MAINT" PATH IS FUNCTIONAL
3573                                    ;       2. RCV-ACT CLEARS BEFORE RCV-DONE SETS.
3574                                    ;       3. RCV-ACT CAN SET AND CLEAR
3575                                    ;       4. READING RCV DATA REGISTER CLEARS RCV DONE
3576                                    ;
3577 032226  016777  146606  146614           MOV     DFPBR,@XCSR
3578 032234  052777  000004  146606           BIS     #4,@XCSR        ; SET MAINT BIT.
3579 032242  105777  146602           1$:     TSTB    @XCSR           ;INSURE WE'RE "XMIT-RDY"
3580 032246  100375                           BPL     1$
3581 032250  016700  146566                   MOV     DFSPD,R0
3582 032254  006300                           ASL     R0              ; SET A 2 CHAR (20 BITS) DELAY COUNT...
3583 032256  010001                           MOV     R0,R1
3584 032260  010002                           MOV     R0,R2           ;...AND SAVE 3 COPIES FOR LATER.
3585 032262  010003                           MOV     R0,R3
3586 032264  077001                           SOB     R0,.            ; WAIT 20 BIT TIMES...
3587 032266  017700  146554                   MOV     @RBUF,R0        ;...AND DUMMY READ.
3588 032272  017700  146546                   MOV     @RCSR,R0        ; GET RCVR STATUS.
3589 032276  032700  004200                   BIT     #RCVACT!RCVDUN,R0 ; BOTH ACTIVE AND DONE SHOULD BE CLEAR
3590 032302  001402                           BEQ     2$
3591 032304  104423  032222                   ERROR,  T134            ;; *** RCV-ACT AND/OR RCV-DUN ARE SET IN ERROR ***
3592
3593 032310  012777  000000  146534   2$:     MOV     #0,@XBUF        ;NOW, XMIT A NULL CHARACTER
3594 032316  000240                           240
3595 032320  017700  146520           3$:     MOV     @RCSR,R0        ;GET RCVR STATUS
3596 032324  032700  004000                   BIT     #RCVACT,R0
3597 032330  001003                           BNE     4$              ;BR IF AND WHEN ACTIVE SETS.
3598 032332  077106                           SOB     R1,3$
3599 032334  104423  032222                   ERROR,  T134            ;; *** RCV-ACT FAILED TO SET ***
3600 032340  032700  000200           4$:     BIT     #RCVDUN,R0      ;TEST THAT "RCV-DUN" IS NOT SET YET.
3601 032344  001402                           BEQ     5$
3602 032346  104423  032222                   ERROR,  T134            ;; *** RCV-DUN SET WHEN RCV-ACT WAS SET ***
3603 032352  017700  146466           5$:     MOV     @RCSR,R0        ;READ RCVR STATUS AGAIN.
```

PC 32346 - RCV-DUN SET WHEN RCV-ACT WAS SET.

```
3604 032356  032700  004000                BIT     #RCVACT,R0
3605 032362  001403                        BEQ     6$                ; BR IF AND WHEN ACTIVE CLEARS.
3606 032364  077206                        SOB     R2,5$
3607 032366  104423  032222                ERROR.  T134              ;; *** RCV-ACT SET BUT FAILED TO CLEAR ***
3608 032372  017700  146446          6$:   MOV     @RCSR,R0          ;READ RCVR STATUS
3609 032376  032700  000200                BIT     #RCVDUN,R0
3610 032402  001003                        BNE     7$                ; BR IF AND WHEN DONE SETS.
3611 032404  077306                        SOB     R3,6$
3612 032406  104423  032222                ERROR.  T134              ;; *** RCV-DUN FAILED TO SET WHEN RCV-ACT CLEARED ***
3613 032412  017700  146430          7$:   MOV     @RBUF,R0          ; OK, READ SHOULD LOWER DONE FLAG.
3614 032416  017700  146422                MOV     @RCSR,R0          ; GET STATUS.
3615 032422  032700  000200                BIT     #RCVDUN,R0
3616 032426  001402                        BEQ     8$                ; BR IF DONE IS CLEAR.
3617 032430  104423  032222                ERROR.  T134              ;; *** RCV-DUN FAILED TO CLEAR AFTER READING DATA ***
3618 032434                          8$:
3619
3620                                ;*************************************************************************
                                    .SBTTL      T135 -- TEST THE "RCV-BRK", "RCV-ERR", AND "FR-ERR" BITS
                                    ;*************************************************************************
     032434  104420  000135        T135:   BEGINTEST, 135            ;; CHECK SWITCH REGISTER OPTIONS.
3621                                ;
3622                                ; GENERATE A FRAMING ERROR BY SETTING THE RCV-BRK BIT.
3623                                ; ON SLU1 -- EXPECT THAT BRK INVOKES A MASKABLE LEVEL 7 HALT REQUEST.
3624                                ;
3625 032440  013746  000140                MOV     @#BHALT,-(SP)     ; SAVE BREAK VECTOR.
3626 032444  106427  000340          1$:   MTPS    #PRI7             ; RAISE CPU.
3627 032450  016777  146364  146372        MOV     DFPBR,@XCSR
3628 032456  052777  000004  146364        BIS     #4,@XCSR          ; SET MAINT BIT.
3629 032464  017700  146356                MOV     @RBUF,R0          ; DUMMY READ.
3630 032470  012767  032624  145442        MOV     #6$,BHALT         ; SET THE HALT VECTOR (FOR SLU1).
3631 032476  012737  000340  000142        MOV     #PRI7,@#BHALT+2
3632 032504  016701  146332                MOV     DFSPD,R1
3633 032510  006301                        ASL     R1                ; SET A 2 CHAR (20 BITS) DELAY.
3634 032512  052777  000001  146330        BIS     #XMTBRK,@XCSR     ; SET "XBRK"...
3635 032520  012777  000025  146324        MOV     #25,@XBUF         ;...START XMITTER...
3636 032526  077101                        SOB     R1,.              ;...AND DELAY.  IF SLU1, INTERRUPT...
3637                                                                  ;...SHOULD BE HELD OFF (CPU IS AT 7).
3638 032530  105777  146310                TSTB    @RCSR             ; DONE SHOLD BE THERE BY NOW.
3639 032534  100402                        BMI     2$                ; BR IF SO.
3640 032536  104423  032444                ERROR.  1$                ;; *** NO RCV-DUN AFTER XMIT WITH XMT-BRK SET ***
3641 032542  026727  146276  176540  2$:   CMP     RCSR,#SLU2        ; IF SLU2...
3642 032550  001430                        BEQ     7$                ;...JUST VERIFY STATUS/DATA.
3643                                ;
3644                                ; SLU1, VERIFY THAT "BHALT" INTERRUPT DOES THE RIGHT THINGS.
3645                                ;
3646 032552  012767  032574  145360  3$:   MOV     #4$,BHALT         ; SO FAR, SO GOOD, CHANGE THE VECTOR...
3647 032560  106427  000300                MTPS    #PRI6             ;...AND LOWER CPU TO 6...
3648 032564  000240                        240                       ;...INTERRUPT SHOULD COME IN.
3649 032566  104423  032444                ERROR.  1$                ;; *** HALT INTERRUPT NOT RECEIVED AT CPU LEVEL 6 ***
3650 032572  000417                        BR      7$
3651 032574  022626                4$:   CMP     (SP)+,(SP)+       ; OK, INT RECEIVED, FIX STACK.
3652 032576  012767  032614  145334        MOV     #5$,BHALT         ; NOW, CHANGE THE VECTOR...
3653 032604  106427  000300                MTPS    #PRI6             ;...LOWER CPU AGAIN...
3654 032610  000240                        240                       ;...AND SEE THAT WE DON'T GET ANOTHER.
3655 032612  000407                        BR      7$                ; TERRIFIC -- PROCEED.
3656 032614  022626                5$:   CMP     (SP)+,(SP)+       ; 2ND HALT TRAP -- TOUGH LUCK !!
3657 032616  104423  032444                ERROR.  1$                ;; *** DOUBLE HALT TRAP RECEIVED ***
```

```
PC 32616 = DOUBLE HALT TRAP RECEIVED.

 3658 032622  000403                          BR      7$
 3659 032624  022626                  6$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF HALT WASN'T MASKED !!
 3660 032626  104423  032444                  ERROR.  1$              ;; *** HALT TRAP INTERRUPT NOT MASKED BY CPU LEVEL 7 ***
 3661                                         ;
 3662                                         ; BOTH LINES VERIFY CORRECT STATUS AND DATA BITS.
 3663                                         ;
 3664 032632  017700  146210          7$:     MOV     @#RBUF,R0       ; READ (IF SLU1, LOWERS BRK REQ)...
 3665 032636  012637  000140                  MOV     (SP)+,@#HALT    ;...AND RESTORE BREAK VECTOR.
 3666 032642  032700  004000                  BIT     #RCVBRK,R0      ; RCV-BRK SHOULD BE SET.
 3667 032646  001002                          BNE     8$
 3668 032650  104423  032444                  ERROR.  1$              ;; *** RCV-BRK BIT DIDN'T SET ***
 3669 032654  032700  100000          8$:     BIT     #RCVERR,R0      ; SO SHOULD RCV-ERR...
 3670 032660  001403                          BEQ     9$
 3671 032662  032700  020000                  BIT     #FRERR,R0       ;...AND FR-ERR.
 3672 032666  001002                          BNE     10$
 3673 032670                          9$:
      032670  104423  032444                  ERROR.  1$              ;; *** RCV-ERR AND/OR FR-ERR BITS DIDN'T SET ***
 3674 032674  105700                  10$:    TSTB    R0              ; RCVD DATA SHOULD BE ZERO.
 3675 032676  001402                          BEQ     11$
 3676 032700  104423  032444                  ERROR.  1$              ;; *** RECEIVED DATA NON-ZERO ON "BRK" ***
 3677                                         ;
 3678                                         ; AND FINALLY, WE SHOULD BE ABLE TO RESTORE A NORMAL RECEIVER.
 3679                                         ;
 3680 032704  042777  000001  146136  11$:     BIC     #XMTBRK,@#XCSR  ; NOW, CLEAR THE BREAK BIT...
 3681 032712  016700  146124                  MOV     DFSPD,R0
 3682 032716  077001                          SOB     R0,.            ;...DELAY...
 3683 032720  017700  146122                  MOV     @#RBUF,R0       ;...DUMMY READ.
 3684 032724  012777  000025  146120          MOV     #25,@#XBUF      ; XMIT A CHAR...
 3685 032732  105777  146106          12$:    TSTB    @#RCSR
 3686 032736  100375                          BPL     12$             ;...WAIT FOR RCVR DONE...
 3687 032740  017700  146102                  MOV     @#RBUF,R0       ;...AND READ.
 3688 032744  032700  124000                  BIT     #RCVERR!FRERR!RCVBRK,R0 ; ERRORS SHOULD BE GONE.
 3689 032750  001402                          BEQ     13$             ; BR IF SO.
 3690 032752  104423  032704                  ERROR.  11$             ;; *** RCV-ERR, FR-ERR, AND/OR BRK BITS DIDN'T CLEAR ***
 3691 032756  120027  000025          13$:    CMPB    R0,#25
 3692 032762  001402                          BEQ     14$             ; AND CORRECT DATA RECEIVED.
 3693 032764  104423  032704                  ERROR.  11$             ;; *** DATA INCORRECT AFTER "BRK" SEQUENCE ***
 3694 032770  106427  000000          14$:    MTPS    #PRIO           ; DONE, LOWER THE CPU.
 3695
 3696                                         ;*****************************************************************************
                                              .SBTTL      T136 -- TEST THAT "RCV-ERR" AND "OR-ERR" BITS SET AND CLEAR
                                              ;*****************************************************************************
      032774  104420  000136          T136:   BEGINTEST, 136          ;; CHECK SWITCH REGISTER OPTIONS.
 3697 033000  016777  146034  146042          MOV     DFPBR,@#XCSR
 3698 033006  052777  000004  146034          BIS     #4,@#XCSR       ; SET MAINT BIT.
 3699 033014  017700  146026                  MOV     @#RBUF,R0       ; DUMMY READ.
 3700 033020  012701  000004                  MOV     #4,R1
 3701 033024  012777  000000  146020  1$:     MOV     #0,@#XBUF       ;XMIT A CHARACTER
 3702 033032  105777  146012          2$:     TSTB    @#XCSR          ;WAIT FOR XMIT-RDY
 3703 033036  100375                          BPL     2$
 3704 033040  077107                          SOB     R1,1$           ; LOOP 4 TIMES...
 3705                                                                 ;...SHOULD CAUSE AN OVER-RUN ERROR.
 3706 033042  016700  145774                  MOV     DFSPD,R0
 3707 033046  077001                          SOB     R0,.            ; WAIT 1 ADDITIONAL CHAR TIME...
 3708 033050  017700  145772                  MOV     @#RBUF,R0       ;...AND READ RCV DATA BUFFER.
 3709 033054  100003                          BPL     3$              ; BR IF NO "RCV-ERR".
 3710 033056  032700  040000                  BIT     #ORERR,R0       ; "OR-ERR" SHOULD BE SET AS WELL.
```

T136 -- TEST THAT "RCV-ERR" AND "OR-ERR" BITS SET AND CLEAR

```
 3711 033062  001002                        BNE     4$              ; BOTH SET -- TERRIFIC.
 3712 033064                          3$:
      033064  104423  032774                ERROR.  T136            ;; *** RCV-ERR AND/OR OR-ERR NOT SET ON FORCED OVER-RUN ***
 3713
 3714 033070  012777  000000  145754  4$:   MOV     #0,@XBUF        ; NOW XMIT AGAIN...
 3715 033076  105777  145742        5$:   TSTB    @RCSR           ;...WAIT FOR RCVR DUN...
 3716 033102  100375                        BPL     5$
 3717 033104  017700  145736              MOV     @RBUF,R0        ;...AND READ, ERROR SHOULD BE GONE.
 3718 033110  032700  140000              BIT     #RCVERR!ORERR,R0
 3719 033114  001402                        BEQ     6$              ; BR IF SO.
 3720 033116  104423  032774              ERROR.  T136            ;; *** RCV-ERR AND/OR OR-ERR BITS FAILED TO CLEAR ***
 3721 033122                          6$:
 3722
 3723                                 ;*************************************************************************
                                      .SBTTL      T137 -- TEST THAT WE CAN INTERRUPT ON "RCV-DUN"
                                      ;*************************************************************************
      033122  104420  000137         T137:   BEGINTEST, 137          ;; CHECK SWITCH REGISTER OPTIONS.
 3724 033126  016777  145706  145714        MOV     DFPBR,@XCSR
 3725 033134  052777  000004  145706        BIS     #4,@XCSR        ; SET MAINT BIT.
 3726 033142  016701  145706              MOV     RVEC,R1         ; GET RCVR VECTOR POINTER.
 3727 033146  106461  000002              MTPS    2(R1)           ; RAISE CPU TO RCVR LEVEL.
 3728 033152  012711  033204              MOV     #2$,(R1)        ; SET RCVR VECTOR.
 3729 033156  012777  000000  145666        MOV     #0,@XBUF        ;TRANSMIT A CHARACTER
 3730 033164  105777  145654        1$:   TSTB    @RCSR           ;WAIT FOR "RCV-DUN"
 3731 033170  100375                        BPL     1$
 3732 033172  052777  000100  145644        BIS     #100,@RCSR      ; SET ENABLE "RCV-IE" BIT
 3733 033200  000240                        240                     ; INTERRUPT SHOULD BE HELD OFF...
 3734 033202  000406                        BR      3$              ;...BR IF SO.
 3735 033204  022626                  2$:   CMP     (SP)+,(SP)+     ; BUT IT WASN'T.
 3736 033206  042777  000100  145630        BIC     #100,@RCSR      ;REMOVE "RCV-IE"
 3737 033214  104423  033122              ERROR.  T137            ;; *** RCV-DUN INTERRUPT AT WRONG LEVEL (TOO HIGH) ***
 3738
 3739 033220  012711  033246        3$:   MOV     #4$,(R1)        ; CHANGE VECTOR.
 3740 033224  106427  000000              MTPS    #PRIO           ;LOWER CPU PRIORITY
 3741 033230  000240                        240                     ; INTERRUPT SHOULD COME IN.
 3742 033232  042777  000100  145604        BIC     #100,@RCSR      ; BUT IT DIDN'T, REMOVE "RCV-IE" BIT
 3743 033240  104423  033122              ERROR.  T137            ;; *** RCV-DUN INTERRUPT NOT RECEIVED AT ***
 3744 033244  000415                        BR      6$
 3745                                 ;
                                      ; NOW, ON INTERRUPT, WE'RE HERE AT LEVEL 4 (5 IF SLU2).
                                      ; LOWER THE CPU AND VERIFY THAT WE DON'T GET ANOTHER ONE.
                                      ;
 3749 033246  022626                  4$:   CMP     (SP)+,(SP)+
 3750 033250  012711  033264              MOV     #5$,(R1)        ; CHANGE THE VECTOR.
 3751 033254  106427  000000              MTPS    #PRIO           ;LOWER THE CPU PRIORITY AGAIN.
 3752 033260  000240                        240                     ; SHOULDN'T GET ANOTHER...
 3753 033262  000406                        BR      6$              ;...OK, EXIT.
 3754 033264  022626                  5$:   CMP     (SP)+,(SP)+     ; 2ND INTERRUPT -- TOO BAD !!
 3755 033266  042777  000100  145550        BIC     #100,@RCSR      ;REMOVE "RCV-IE"
 3756 033274  104423  033122              ERROR.  T137            ;; *** INTERRUPT ACK FAILED TO REMOVE THE REQUEST ***
 3757
 3758 033300  042777  000100  145536  6$:   BIC     #100,@RCSR      ; INSURE THE ENABLE IS OFF.
 3759 033306  017700  145534              MOV     @RBUF,R0        ; READ
 3760 033312  012711  030320              MOV     #BADRI,(R1)     ;RESET FALSE INTR VECTOR
 3761
 3762                                 ;*************************************************************************
                                      .SBTTL      T140 -- FLOAT 1 AND 0 THRU THE "MAINT" DATA LOOP (INTERNAL)
```

```
                                ;*****************************************************************************
     033316  104420  000140     T140:   BEGINTEST, 140          ;; CHECK SWITCH REGISTER OPTIONS.
3763                            ;
3764                            ; NOTE:  IF THERE'S A TERMINAL ATTACHED, YOU'LL SEE GARBAGE
3765                            ; CHARACTERS ON THE SCREEN -- DON'T WORRY ABOUT IT !!
3766                            ;
3767 033322  016777  145512  145520        MOV     DFPBR,@XCSR
3768 033330  052777  000004  145512        BIS     #4,@XCSR        ; SET MAINT BIT.
3769 033336  012700  030340                MOV     #SYNC,R0        ; DATA TABLE POINTER.
3770 033342  112077  145504          1$:   MOVB    (R0)+,@XBUF     ; XMIT THE SYNC PAIR <CR>...
3771 033346  105777  145476          2$:   TSTB    @XCSR
3772 033352  100375                       BPL     2$
3773 033354  112077  145472                MOVB    (R0)+,@XBUF     ;...AND <LF>.
3774 033360  105777  145460          3$:   TSTB    @RCSR           ; WAIT FOR RCVR DONE...
3775 033364  100375                       BPL     3$
3776 033366  127727  145454  000012        CMPB    @RBUF,#12       ; AND PROCEED ON THE RCVD <LF>.
3777 033374  001371                       BNE     3$
3778
3779 033376  112001                4$:   MOVB    (R0)+,R1        ; NOW FLOAT DATA THRU THE LOOP.
3780 033400  010177  145446          5$:   MOV     R1,@XBUF        ; XMIT A BYTE...
3781 033404  105777  145434          6$:   TSTB    @RCSR
3782 033410  100375                       BPL     6$
3783 033412  017702  145430                MOV     @RBUF,R2        ;...AND READ IT BACK.
3784 033416  120102                       CMPB    R1,R2
3785 033420  001402                       BEQ     7$
3786 033422  104423  033400                ERROR.  5$              ;; *** RCVD DATA INCORRECT, MAINT DATA PATH FAILURE ***
3787 033426  105701                7$:   TSTB    R1              ; TERMINATOR DONE ??
3788 033430  001362                       BNE     4$              ; NOT YET, LOOP.
3789
3790                            ;*****************************************************************************
                                .SBTTL      T141 -- FLOAT 1 AND 0 THRU THE LOOP-BACK (EXTERNAL, SLU2 ONLY)
                                ;*****************************************************************************
     033432  104420  000141     T141:   BEGINTEST, 141          ;; CHECK SWITCH REGISTER OPTIONS.
3791 033436  026727  145402  177560        CMP     RCSR,#SLU1      ; CONSOLE LINE ??
3792 033444  001455                       BEQ     DLDUN           ; DON'T IF SO.
3793
3794 033446  016777  145366  145374        MOV     DFPBR,@XCSR     ; DEFAULT XMIT STATUS (NO MAINT).
3795 033454  017700  145366                MOV     @RBUF,R0        ; DUMMY READ.
3796 033460  016700  145356                MOV     DFSPD,R0
3797 033464  006300                        ASL     R0
3798 033466  006300                        ASL     R0              ; 4 CHAR LOOP TIMER.
3799 033470  010001                        MOV     R0,R1           ; SAVE A COPY.
3800 033472  112777  000033  145352        MOVB    #33,@XBUF       ; XMIT AN "ESC".
3801 033500  105777  145340          1$:   TSTB    @RCSR           ; IF RCVR-DUN SETS NOW, WE CAN ASSUME...
3802 033504  100413                       BMI     3$              ;...THAT THE LOOP-BACK IS INSTALLED.
3803 033506  077004                       SOB     R0,1$
3804 033510  112777  000132  145334        MOVB    #'Z,@XBUF       ; IF NOT, XMIT "Z".
3805 033516  105777  145322          2$:   TSTB    @RCSR           ; IF IT SETS NOW, WE HAVE A TTY...
3806 033522  100426                       BMI     DLDUN           ;...RESPONDING TO THE "ESC Z" SEQUENCE.
3807 033524  077104                       SOB     R1,2$           ; IF NEITHER, THE LOOP MUST BE OPEN.
3808 033526  104423  033432                ERROR.  T141            ;; *** NO RCV-DUN -- IS LOOP CONNECTOR INSTALLED ??? ***
3809 033532  000422                       BR      DLDUN           ; QUIT (WHILE WE'RE AHEAD) !!
3810
3811 033534  017700  145306          3$:   MOV     @RBUF,R0        ; OK, DUMMY READ...
3812 033540  012700  030342                MOV     #FLT1,R0        ;...SET TABLE POINTER...
3813 033544  112001                4$:   MOVB    (R0)+,R1        ;...AND FLOAT DATA THRU THE LOOP.
3814 033546  010177  145300          5$:   MOV     R1,@XBUF        ; XMIT A BYTE...
```

PC 33526 = NO RCV-DUN -- IS LOOP CONNECTOR INSTALLED ???.

```
3815 033552  105777  145266          6$:     TSTB    @RCSR
3816 033556  100375                  BPL     6$
3817 033560  017702  145262                  MOV     @RBUF,R2        ;...AND READ IT BACK.
3818 033564  120102                  CMPB    R1,R2
3819 033566  001402                  BEQ     7$
3820 033570  104423  033546                  ERROR.  5$              ;; *** RCVD DATA INCORRECT, EXTERNAL DATA PATH FAILURE ***
3821 033574  105701                  7$:     TSTB    R1              ; TERMINATOR DONE ??
3822 033576  001362                  BNE     4$              ; NOT YET, LOOP.
3823
3824 033600  016777  145234  145242  DLDUN:  MOV     DFPBR,@XCSR     ; INSURE A DEFAULT XMITTER...
3825 033606  017700  145234                  MOV     @RBUF,R0        ;...AND RECEIVER.
3826 033612  012767  000006  144164          MOV     @ERRVEC+2,ERRVEC
3827 033620  004767  001232                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
3828                                                                  ;...RETURN AND FALL THRU IF NOT.
3829 033624  026727  145214  176540          CMP     RCSR,@SLU2      ; LINE 2 ??
3830 033632  001402                  BEQ     S7              ; BR IF SO...
3831 033634  000167  174302                  JMP     S6              ;...ELSE, GO 'ROUND ONCE.
3832                                  .SBTTL
```

SECTION 7 -- PARALLEL I/O PORT (8255) TESTS.

```
                                        .SBTTL  SECTION 7 -- PARALLEL I/O PORT (8255) TESTS.
                                        ;
                                        ; PORTS A, B, AND C MUST BE INTERCONNECTED THRU A SPECIAL
                                        ; LOOP-BACK CONNECTOR.  IF NOT THESE TESTS BREAK.
                                        ;
            000233                      PPM0=   233             ; CONTROL WORD TO SET MODE 0 (ALL INPUT).
            000264                      PPM1=   264             ; ANOTHER FOR MODE 1, A IN, B OUT, C<7:6> OUT.
                                        ;
            000200                      LEDBIT= 200             ; LED CONTROL BIT (0 = ON, 1 = OFF).
            000040                      INRDY=  040             ; INPUT DATA READY ( IBF ).
            000020                      INENA=  020             ; ENABLE INTERRUPT ON "INRDY"
            000010                      ININT=  010             ; INPUT INTERRUPT BIT.
                                        ;
            000004                      OUTENA= 004             ; ENABLE INTERRUPT ON "OUTRDY".
            000002                      OUTRDY= 002             ; OUTPUT PORT READY ( -OBF ).
            000001                      OUTINT= 001             ; OUTPUT INTERRUPT BIT.
                                        ;
033640  012767  036453  002460  S7:     MOV     #SEG7,SEGN      ; SEGMENT 7 TEXT.

                                        ;*****************************************************************
                                        .SBTTL      T142 -- TEST PIO REGISTER ADDRESSES
                                        ;*****************************************************************
033646  104420  000142                  T142:   BEGINTEST, 142          ;; CHECK SWITCH REGISTER OPTIONS.
033652  000240  000240                          240,240                 ; *** TRY A RESET HERE ***
033656  012700  176200                          MOV     #PIOA,R0        ; A IS 1ST OF 4 REGISTERS.
033662  012701  000130                          MOV     #PIOBV,R1       ; B IS 1ST OF 2 VECTOR PAIRS.
033666  012702  000240                          MOV     #PIOP,R2        ; PRIORITY (5) FOR BOTH.
033672  012767  033762  144104                  MOV     #BADPIO,ERRVEC  ; SET TIME-OUT VECTOR.
033700  012706  003776                          MOV     #STACK,SP
033704  010067  145150                          MOV     R0,PPA          ; SET REGISTER POINTERS.
033710  005720                                  TST     (R0)+           ; IF ANY TRAP -- TROUBLE !@!
033712  010067  145144                          MOV     R0,PPB
033716  005720                                  TST     (R0)+
033720  010067  145140                          MOV     R0,PPC
033724  005720                                  TST     (R0)+
033726  010067  145134                          MOV     R0,PPCW
033732  005710                                  TST     (R0)
033734  010167  145132                          MOV     R1,PPBV         ; SAVE B VECTOR POINTER.
033740  012721  033772                          MOV     #BADBI,(R1)+    ; SET TO CATCH UNEXPECTED INTERRUPTS.
033744  010221                                  MOV     R2,(R1)+
033746  010167  145116                          MOV     R1,PPAV         ; DITTO FOR PORT A VECTOR.
033752  012721  034002                          MOV     #BADAI,(R1)+
033756  010221                                  MOV     R2,(R1)+
033760  000414                                  BR      PIOTST
                                        ;
                                        ; COME HERE IF TIME-OUT OR UNEXPECTED PIO INTERRUPTS.
                                        ;
033762                                  BADPIO:
033762  104423  033646                          ERROR,  T142            ;; *** BUS TIME-OUT ON PIO ADDRESS ***
033766  000167  001050                          JMP     PIODUN
033772                                  BADBI:
033772  104423  034012                          ERROR,  PIOTST          ;; *** UNEXPECTED PORT B INTERRUPT ***
033776  000167  001040                          JMP     PIODUN
034002                                  BADAI:
034002  104423  034012                          ERROR,  PIOTST          ;; *** UNEXPECTED PORT A INTERRUPT ***
034006  000167  001030                          JMP     PIODUN
034012                                  PIOTST:
```

PC 34002 = UNEXPECTED PORT A INTERRUPT.

```
3885                                    ;*******************************************************************************
                                        .SBTTL      T143 -- TEST THAT RESET INVOKES MODE 0 AND CLEARS THE CHIP
                                        ;*******************************************************************************
     034012  104420  000143          T143:   BEGINTEST, 143          ;; CHECK SWITCH REGISTER OPTIONS.
3886                                    ;
3887                                    ; VERIFY THAT PORT A (AND IT'S BUFFER) IS IN THE "RESET" STATE.
3888                                    ; PORTS B AND C WILL BE INDETERMINATE DUE TO THE LOOP-BACK INTERCONNECTS
3889                                    ; AND THE CONTROL WORD PORT IS "WO", SO DON'T BOTHER WITH THOSE.
3890                                    ;
3891 034016  000240  000240                    240,240
3892 034022  000005                     RESET                   ; LED (ON C<7:) SHOULD BE OFF.
3893 034024  005000                     CLR     R0              ; DELAY, CHIP SEEMS TO TAKE QUITE...
3894 034026  077001                     SOB     R0,.            ;...A WHILE (200USEC) TO GET "RESET".
3895 034030  117700  145024            MOVB    @#PPA,R0        ; READ PORT A (INPUT).
3896 034034  120027  177777            CMPB    R0,#-1          ; LO BYTE ONLY -- SHOULD BE HIGH.
3897 034040  001402                     BEQ     1$
3898 034042  104423  034012            ERROR,  T143            ;; *** PORT A RESET STATE INCORRECT ***
3899 034046                         1$:
3900
3901                                    ;*******************************************************************************
                                        .SBTTL      T144 -- TEST MODE 1 I/O THRU THE LOOP CONNECTOR (FLAG MODE)
                                        ;*******************************************************************************
     034046  104420  000144          T144:   BEGINTEST, 144          ;; CHECK SWITCH REGISTER OPTIONS.
3902 034052  112777  000264  145006        MOVB    #PPM1,@#PPCW    ; SET MODE 1, A IN, B OUT, C<7:6> OUT.
3903                                ;       MOVB    #<7_1.>!0,@#PPCW ; CLEAR C(7) TO TURN ON THE TINY LED.   ;JSD REV B
3904 034060  112777  000016  145000        MOVB    #16.,@#PPCW     ; CLEAR C(7) TO TURN ON THE TINY LED.   ;JSD REV B
3905                                ;       MOVB    #<6_1.>!0,@#PPCW ; CLEAR C(6) JUST FOR THE HELL OF IT.   ;JSD REV B
3906 034066  112777  000014  144772        MOVB    #14.,@#PPCW     ; CLEAR C(6) JUST FOR THE HELL OF IT.   ;JSD REV B
3907 034074  012700  000012                MOV     #10.,R0
3908 034100  077001                     SOB     R0,.            ; DELAY ABOUT 50 USEC...
3909 034102  117703  144752            MOVB    @#PPA,R3        ;...AND DUMMY READ TO DROP "INRDY".
3910 034106  117703  144752            MOVB    @#PPC,R3        ; NOW, GET STATUS.
3911 034112  120327  000002            CMPB    R3,#OUTRDY      ; SHOULD HAVE THIS STATUS SET.
3912 034116  001402                     BEQ     1$
3913 034120  104423  034046            ERROR,  T144            ;; *** PORT C MODE 1 STATUS INCORRECT ***
3914
3915 034124  005001                 1$: CLR     R1              ; SEND A BINARY COUNT FROM R1...
3916 034126  012702  177777            MOV     #-1,R2          ;...TO R2 THRU THE 8255.
3917 034132  005000                 2$: CLR     R0              ; LOOP DELAY.
3918 034134  110177  144722            MOVB    R1,@#PPB        ; XMIT DATA OUTPUT (THRU B).
3919 034140  117703  144720        3$: MOVB    @#PPC,R3        ; GET STATUS.
3920 034144  032703  000002            BIT     #OUTRDY,R3      ; OUTRDY SHOULD BE GONE.
3921 034150  001402                     BEQ     4$
3922 034152  104423  034132            ERROR,  2$              ;; *** OUTRDY STILL SET AFTER OUTPUT XFER ***
3923 034156  117703  144702        4$: MOVB    @#PPC,R3        ; GET STATUS AGAIN...
3924 034162  120327  000042            CMPB    R3,#INRDY!OUTRDY ;...AND WAIT FOR THIS.
3925 034166  001403                     BEQ     5$              ; WONDERFUL, MOVE ON.
3926 034170  077006                     SOB     R0,4$
3927 034172  104423  034132            ERROR,  2$              ;; *** STATUS INCORRECT AFTER OUTPUT XFER ***
3928 034176  117702  144656        5$: MOVB    @#PPA,R2        ; DATA INPUT (THRU A).
3929 034202  117703  144656            MOVB    @#PPC,R3        ; GET STATUS
3930 034206  120327  000002            CMPB    R3,#OUTRDY      ; EXPECT THIS STATUS.
3931 034212  001402                     BEQ     6$              ; BR IF SO.
3932 034214  104423  034132            ERROR,  2$              ;; *** STATUS INCORRECT AFTER INPUT XFER ***
3933 034220  120201                 6$: CMPB    R2,R1           ; DATA LOOP INTACT ??
3934 034222  001402                     BEQ     7$              ; BR IF SO.
3935 034224  104423  034132            ERROR,  2$              ;; *** DATA INCORRECT AFTER OUT/IN EXCHANGE ***
```

PC 34224 = DATA INCORRECT AFTER OUT/IN EXCHANGE.

```
3936 034230  105201                  7$:     INCB    R1
3937 034232  001337                          BNE     2$                      ; LOOP 'TIL DONE.
3938
3939                                 ;*****************************************************************************
                                     .SBTTL      T145 -- TEST THE PORT B (OUTPUT) INTERRUPT
                                     ;*****************************************************************************
     034234  104420  000145          T145:   BEGINTEST, 145                  ;; CHECK SWITCH REGISTER OPTIONS.
3940 034240  016704  144626                  MOV     PPBV,R4                 ; GET B VECTOR POINTER.
3941 034244  106464  000002                  MTPS    2(R4)                   ; RAISE CPU TO PIO LEVEL...
3942 034250  012714  034266                  MOV     #1$,(R4)                ;...AND SET VECTOR.
3943                                 ;       MOVB    #<2_1.>!1,@PPCW         ; SET C<2> B INTR ENABLE.       ;JSD REV B
3944 034254  112777  000005  144604          MOVB    #5,@PPCW                ; SET C<2> B INTR ENABLE.       ;JSD REV B
3945 034262  000240                          240                             ; OUTRDY IS ALREADY THERE, SO...
3946 034264  000403                          BR      2$                      ;...INTERRUPT REQUEST SHOULD BE THERE...
3947                                                                         ;...BUT HELD OFF BY THE CPU LEVEL.
3948 034266  022626                  1$:     CMP     (SP)+,(SP)+             ; BUT IT WASN'T !!
3949 034270  104423  034234                  ERROR,  T145                    ;; *** PORT B INTERRUPT LEVEL INCORRECT ***
3950 034274  117703  144564          2$:     MOVB    @PPC,R3                 ; GET STATUS.
3951 034300  120327  000007                  CMPB    R3,#OUTENA!OUTRDY!OUTINT
3952 034304  001402                          BEQ     3$
3953 034306  104423  034234                  ERROR,  T145                    ;; *** STATUS INCORRECT WITH PORT B INTERRUPT PENDING ***
3954
3955 034312  012714  034332          3$:     MOV     #4$,(R4)                ; OK, CHANGE VECTOR.
3956 034316  106427  000200                  MTPS    #PRI4                   ; LOWER CPU TO LEVEL 4...
3957 034322  000240                          240                             ;...INTERRUPT SHOULD COME IN.
3958 034324  104423  034234                  ERROR,  T145                    ;; *** PORT B INTERRUPT NOT RECEIVED ***
3959 034330  000401                          SKP1
3960
3961 034332  022626                  4$:     CMP     (SP)+,(SP)+             ; WE'RE HERE ON INTERRUPT, FIX STACK.
3962 034334  012714  034350                  MOV     #5$,(R4)                ; CHANGE THE VECTOR...
3963 034340  106427  000200                  MTPS    #PRI4                   ;...AND LOWER CPU AGAIN...
3964 034344  000240                          240                             ;...SHOULDN'T GET ANOTHER INTERRUPT.
3965 034346  000403                          BR      6$
3966 034350  022626                  5$:     CMP     (SP)+,(SP)+             ; BUT WE DID -- TROUBLE.
3967 034352  104423  034234                  ERROR,  T145                    ;; *** DOUBLE PORT B INTERRUPT ***
3968
3969                                 ;6$:    MOVB    #<2_1.>!0,@PPCW         ; OK, CLEAR THE ENABLE...       ;JSD REV B
3970 034356  112777  000004  144502  6$:     MOVB    #4,@PPCW                ; OK, CLEAR THE ENABLE...       ;JSD REV B
3971 034364  112777  000001  144470          MOVB    #1,@PPB                 ;...WRITE -- SHOULD LOWER INT REQUEST.
3972 034372  005000                          CLR     R0
3973 034374  117703  144464          7$:     MOVB    @PPC,R3                 ; GET STATUS
3974 034400  032703  000001                  BIT     #OUTINT,R3
3975 034404  001403                          BEQ     8$                      ; BR WHEN REQUEST CLEARS.
3976 034406  077006                          SOB     R0,7$
3977 034410  104423  034234                  ERROR,  T145                    ;; *** PORT B INT REQUEST DIDN'T CLEAR AFTER WRITE ***
3978
3979 034414  132777  000040  144442  8$:     BITB    #INRDY,@PPC
3980 034422  001001                          BNE     9$                      ; BR WHEN "INRDY" COMES UP...
3981 034424  077005                          SOB     R0,8$                   ;...BUT DON'T WAIT FOREVER.
3982 034426  117700  144426          9$:     MOVB    @PPA,R0                 ; DUMMY READ.
3983 034432  012714  033772                  MOV     #BADBI,(R4)             ; RESET VECTOR.
3984
3985                                 ;*****************************************************************************
                                     .SBTTL      T146 -- TEST THE PORT A (INPUT) INTERRUPT
                                     ;*****************************************************************************
     034436  104420  000146          T146:   BEGINTEST, 146                  ;; CHECK SWITCH REGISTER OPTIONS.
3986 034442  016704  144422                  MOV     PPAV,R4                 ; GET PORT A VECTOR POINTER.
```

T146 -- TEST THE PORT A (INPUT) INTERRUPT

```
3987 034446  106464  000002                MTPS    2(R4)              ; RAISE CPU TO PIO LEVEL.
3988                              ;         MOVB    #<4_1.>!1,@PPCW    ; SET C<4> A INTR ENABLE.       ;JSD REV B
3989 034452  112777  000011  144406        MOVB    #11,@PPCW          ; SET C<4> A INTR ENABLE.       ;JSD REV B
3990 034460  117703  144400               MOVB    @PPC,R3
3991 034464  120327  000022               CMPB    R3,#INENA!OUTRDY   ; EXPECT TO SEE THIS.
3992 034470  001402                       BEQ     1$
3993 034472  104423  034436               ERROR,  T146               ;; *** STATUS INCORRECT ***
3994
3995 034476  012714  034520          1$:  MOV     #2$,(R4)           ; SET THE VECTOR.
3996 034502  005000                       CLR     R0
3997 034504  012701  000123               MOV     #123,R1
3998 034510  110177  144346               MOVB    R1,@PPB            ; SEND DATA (B => A) THRU THE LOOP.
3999 034514  077001                       SOB     R0,.               ; INTERRUPT SHOULD BE HELD OFF.
4000 034516  000403                       BR      3$
4001 034520  022626                  2$:  CMP     (SP)+,(SP)+        ; BUT IT WASN'T.
4002 034522  104423  034436               ERROR,  T146               ;; *** PORT A INTERRUPT LEVEL INCORRECT ***
4003
4004 034526  117703  144332          3$:  MOVB    @PPC,R3            ; GET STATUS.
4005 034532  120327  000072               CMPB    R3,#INRDY!INENA!ININT!OUTRDY
4006 034536  001402                       BEQ     4$                 ; BR IF STATUS IS OK.
4007 034540  104423  034436               ERROR,  T146               ;; *** STATUS INCORRECT WITH PORT A INTERRUPT PENDING ***
4008
4009 034544  012714  034564          4$:  MOV     #5$,(R4)           ; CHANGE VECTOR.
4010 034550  106427  000200               MTPS    #PRI4              ; LOWER CPU TO LEVEL 4...
4011 034554  000240                       240                        ;...INTERRUPT SHOULD COME IN.
4012 034556  104423  034436               ERROR,  T146               ;; *** PORT A INTERRUPT NOT RECEIVED ***
4013 034562  000401                       SKP1
4014
4015 034564  022626                  5$:  CMP     (SP)+,(SP)+        ; WE'RE HERE ON INTERRUPT, FIX STACK.
4016 034566  012714  034602               MOV     #6$,(R4)           ; CHANGE VECTOR...
4017 034572  106427  000200               MTPS    #PRI4              ;...AND LOWER AGAIN...
4018 034576  000240                       240                        ;...SHOULDN'T GET ANOTHER.
4019 034600  000403                       BR      7$
4020 034602  022626                  6$:  CMP     (SP)+,(SP)+        ; 2ND INTERRUPT, FIX STACK.
4021 034604  104423  034436               ERROR,  T146               ;; *** DOUBLE PORT A INTERRUPT ***
4022
4023                              ;7$:     MOVB    #<4_1.>!0,@PPCW    ; OK, CLEAR THE ENABLE...       ;JSD REV B
4024 034610  112777  000010  144250   7$:  MOVB    #10,@PPCW          ; OK, CLEAR THE ENABLE...       ;JSD REV B
4025 034616  117702  144236               MOVB    @PPA,R2            ;...READ DATA (SHOULD LOWER FLAGS).
4026 034622  005000                       CLR     R0
4027 034624  117703  144234          8$:  MOVB    @PPC,R3            ; READ STATUS.
4028 034630  032703  000010               BIT     #ININT,R3
4029 034634  001403                       BEQ     9$                 ; BR WHEN INT REQUEST CLEARS.
4030 034636  077006                       SOB     R0,8$
4031 034640  104423  034436               ERROR,  T146               ;; *** PORT A INT REQUEST DIDN'T CLEAR AFTER READ ***
4032
4033 034644  012714  034002          9$:  MOV     #BADAI,(R4)        ; RESET PORT A VECTOR.
4034
4035
                                  ;*************************************************************************
                                  .SBTTL      T147 -- TEST THAT B PRIORITY IS HIGHER THAN A
                                  ;*************************************************************************
     034650  104420  000147          T147:   BEGINTEST, 147             ;; CHECK SWITCH REGISTER OPTIONS.
4036 034654  016704  144212               MOV     PPBV,R4            ; GET B VECTOR POINTER.
4037 034660  106464  000002               MTPS    2(R4)              ; RAISE CPU TO DEVICE LEVEL.
4038 034664  012714  034760               MOV     #3$,(R4)           ; SET B VECTOR.
4039 034670  012764  034750  000004       MOV     #2$,4(R4)          ; SET A VECTOR.
4040                              ;         MOVB    #<2_1.>!1,@PPCW    ; ENABLE B...                   ;JSD REV B
```

T147 -- TEST THAT B PRIORITY IS HIGHER THAN A

```
4041 034676  112777  000005  144162          MOVB    #5,@PPCW        ; ENABLE B...                   ;JSD REV B
4042                                 ;         MOVB    #<4_1.>!1,@PPCW ;...AND A INTERRUPTS.          ;JSD REV B
4043 034704  112777  000011  144154          MOVB    #11,@PPCW       ;...AND A INTERRUPTS.           ;JSD REV B
4044 034712  112777  000001  144142          MOVB    #1,@PPB         ; OUTPUT DATA...
4045 034720  005000                          CLR     R0
4046 034722  132777  000040  144134  1$:     BITB    #INRDY,@PPC     ;...AND WAIT 'TIL DONE.
4047 034730  001001                          BNE     .+4
4048 034732  077005                          SOB     R0,1$           ; KEEP-ALIVE.
4049
4050 034734  106427  000200                  MTPS    #PRI4           ; LOWER CPU, B SHOULD COME IN FIRST.
4051 034740  000240                          240
4052 034742  104423  034650                  ERROR.  T147            ;; *** NEITHER ONE INTERRUPTED -- WHAT THE HELL !! ***
4053 034746  000415                          BR      4$
4054 034750  022626                  2$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF A CAME IN FIRST.
4055 034752  104423  034650                  ERROR.  T147            ;; *** A INTERRUPTED BEFORE B -- BAD NEWS !! ***
4056 034756  000411                          BR      4$
4057
4058 034760  022626                  3$:     CMP     (SP)+,(SP)+     ; WE'RE HERE IF B INTERRUPTED FIRST.
4059 034762  012764  035002  000004          MOV     #4$,4(R4)       ; CHANGE THE A VECTOR...
4060 034770  106427  000200                  MTPS    #PRI4           ;...AND LET IT IN NOW.
4061 034774  000240                          240
4062 034776  104423  034650                  ERROR.  T147            ;; *** A INT NOT RECEIVED AFTER B INT ***
4063
4064                                 ;4$:    MOVB    #<2_1.>!0,@PPCW ; CLEAR THE ENABLES.            ;JSD REV B
4065 035002  112777  000004  144056  4$:     MOVB    #4,@PPCW        ; CLEAR THE ENABLES.            ;JSD REV B
4066                                 ;         MOVB    #<4_1.>!0,@PPCW                                 ;JSD REV B
4067 035010  112777  000010  144050          MOVB    #10,@PPCW                                       ;JSD REV B
4068 035016  112777  000001  144036          MOVB    #1,@PPB         ; LOWER REQUESTS.
4069 035024  117700  144030                  MOVB    @PPA,R0
4070 035030  012714  033772                  MOV     #BADBI,(R4)     ; RESET VECTORS.
4071 035034  012764  034002  000004          MOV     #BADAI,4(R4)
4072
4073 035042  012767  000006  142734  PIODUN: MOV     #ERRVEC+2,ERRVEC
4074 035050  004767  000002                  CALL    ENDSEG          ; CHECK FOR LOOP-IN-SEGMENT...
4075 035054  000410                          BR      ENDPAS          ;...RETURN AND DO ENDPASS IF NOT.
4076                                 .SBTTL
```

END OF PASS ROUTINES.

```
                                .SBTTL  END OF PASS ROUTINES.
                                ;*****************************************************************************
                                ; AT LAST -- WE'VE HIT THE END OF THE TRAIL, PODNAH !!!
                                ; REPORT END-PASS, TAKE A LITTLE NAP, AND THEN DO IT ALL AGAIN.
                                ;
                                ; SOB R0,. EXECUTES IN 3.66 USEC IN KXT11.
                                ;          IF Q-BUS TRANSACTION REQUIRED, ADD 609 NSEC (CYCLE SLIP).
                                ;          THEREFORE, SOB FOR TIMING TAKES ABOUT 4.27 USEC/COUNT.
                                ;
035056  005767  000730          ENDSEG: TST     INSEG           ; LOOP-IN-SEGMENT ??
035062  001001                          BNE     1$              ; YES.
035064  000207                          RETURN                  ; JUST RETURN IF NOT.

035066  104007                  1$:     EMT+7                   ; <CRLF>
035070  016700  001232                  MOV     SEGN,R0         ; CURRENT SEGMENT DONE.
035074  000402                          SKP2
035076  012700  035256          ENDPAS: MOV     #EOP,R0         ; ALL SEGMENTS DONE.
035102  104003                          EMT+3
035104  012700  035267                  MOV     #EOP1,R0
035110  104003                          EMT+3                   ; END PASS.
035112  005267  146664                  INC     TPASS           ; BUMP TOTAL PASS COUNT.
035116  026727  146656                  CMP     ERRCNT,(PC)+    ; ANY ERRORS THIS PASS ??
035122  000000                  1$:     0
035124  001002                          BNE     2$              ; SKIP IF SO.
035126  005267  146652                  INC     GPASS           ; NO, BUMP CONSECUTIVE-GOOD PASS COUNT.
035132  016767  146642  177762  2$:     MOV     ERRCNT,1$       ; UPDATE SAVED COUNT.
035140  016700  146636                  MOV     TPASS,R0
035144  104424                          TRAP+24                 ; PASS COUNT...
035146  012700  035303                  MOV     #EOPE,R0
035152  104003                          EMT+3
035154  016700  146620                  MOV     ERRCNT,R0
035160  104424                          TRAP+2#                 ;...AND TOTAL ERROR COUNT.
035162  104007                          EMT+7                   ; <CRLF>
035164  032767  010000  143004          BIT     #10000,SWR      ; ERROR SUMMARY ??
035172  001402                          BEQ     3$
035174  004767  000124                  CALL    ERRSUM          ; PRINT IT IF SO.
035200  106427  000000          3$:     MTPS    #PRIO           ; LOWER THE CPU.
035204  005000                          CLR     R0
035206  012701  000004                  MOV     #4,R1
035212  077001                  4$:     SOB     R0,.            ; ENDPASS DELAY...
035214  077102                          SOB     R1,4$           ;...ABOUT 1 SECOND.

035216  013700  000042                  MOV     @#42,R0
035222  001405                          BEQ     5$              ; BR IF NOT CHAINING.
035224  000240                          NOP                     ; OR RESET.
035226  004710                          CALL    (R0)            ; SYSMAC STYLE CHAIN RETURN.
035230  000240  000240  000240          240,240,240

035236  016700  000550          5$:     MOV     INSEG,R0        ; GET ITERATION ADDRESS.
035242  001002                          BNE     6$
035244  012700  010336                  MOV     #DOALL,R0       ; IF NONE, BACK TO SQUARE 1.
035250  012706  003776          6$:     MOV     #STACK,SP       ; RESET STACK...
035254  000110                          JMP     (R0)            ;...AND LOOP FOREVER.

035256     015     012     116  EOP:    .ASCIZ  <15><12>'NKXABO'
035261     113     130     101
035264     102     060     000
```

END OF PASS ROUTINES.

```
4133 035267  054  040  105  EOP1:   .ASCIZ  ', END PASS '
     035272  116  104  040
     035275  120  101  123
     035300  123  040  000
4134 035303  054  040  124  EOPE:   .ASCIZ  ', TOTAL ERRORS '
     035306  117  124  101
     035311  114  040  105
     035314  122  122  117
     035317  122  123  040
     035322  000
4135                              .EVEN
4136
4137                         ;*****************************************************************************
4138                         ; ERROR SUMMARY REPORT.
4139                         ; CALLED IN "END-PASS" IF SWR<12> IS SET.
4140                         ;
4141 035324  012702  004006 ERRSUM: MOV     #EPCTBL,R2
4142 035330  005712                 TST     (R2)
4143 035332  001414                 BEQ     2$                  ; BR IF NO ERROR DATA LOGGED.
4144 035334  012700  035366         MOV     #ERHEAD,R0
4145 035340  104003                 EMT+3                       ; HEADER.
4146 035342                      1$:
4150 035342  104010                 EMT+10                      ;; <TAB>
     035344  012200                 MOV     (R2)+,R0
     035346  104424                 TRAP+24                     ;; PRINT (R2)+
     035350  104010                 EMT+10                      ;; <TAB>
     035352  012200                 MOV     (R2)+,R0
     035354  104424                 TRAP+24                     ;; PRINT (R2)+
4151 035356  104007                 EMT+7                       ; <CRLF>
4152 035360  005712                 TST     (R2)
4153 035362  001367                 BNE     1$                  ; LOOP 'TIL DONE.
4154 035364  000207                 RETURN
4155
4156 035366    015    012    011  ERHEAD: .ASCIZ  <15><12><11>'ERRPC'<11>' TIMES'<15><12>
     035371    105    122    122
     035374    120    103    011
     035377    040    124    111
     035402    115    105    123
     035405    015    012    000
4157                              .EVEN
```

```
END OF PASS ROUTINES.
   4158
   4159                                ;*****************************************************************************
   4160                                ; SUBROUTINE TO CHECK FOR Q-BUS EXERCISOR.
   4161                                ; RETURN QBX CSR1 ADDRESS IN R0 IF IT'S THERE.
   4162                                ;
   4163 035410  005767  143422        QBXIN:  TST     QBXA            ; QBX INSTALLED ??
   4164 035414  001020                        BNE     3$              ; BR IF SO.
   4165 035416  012767  035442  142360        MOV     #1$,ERRVEC      ; DON'T KNOW YET, SET TRAP CATCHER...
   4166 035424  012700  174020                MOV     #QBXCSR,R0      ;...AND A POINTER.
   4167 035430  005010                        CLR     (R0)            ; CLEAR QBX CSR1 (IF POSSIBLE)...
   4168 035432  000240                        240                     ;...TRAP TO 1$ IF NOT.
   4169 035434  010067  143376                MOV     R0,QBXA         ; IT'S THERE, SET FLAG = CSR ADDRESS...
   4170 035440  000403                        BR      2$              ;...AND RETURN.
   4171
   4172 035442  005067  143370        1$:     CLR     QBXA            ; NOT THERE, CLEAR FLAG.
   4173 035446  022626                        CMP     (SP)+,(SP)+     ; FIX STACK.
   4174 035450  012767  000006  142326  2$:     MOV     #ERRVEC+2,ERRVEC ; RESET ERROR VECTOR.
   4175 035456  016700  143354        3$:     MOV     QBXA,R0         ; RETURN CSR POINTER OR ZERO.
   4176 035462  000207                        RETURN
```

```
                                .SBTTL  TRAP HANDLER
                                ;*****************************************************************************
                                ; TRAP HANDLER
                                ;
035464  010046                  TRAPHAN: MOV     R0,-(SP)        ; SAVE CALLERS R0...
035466  010146                           MOV     R1,-(SP)        ;...AND R1.
035470  016600  000004                   MOV     4(SP),R0        ; GET TRAP PC.
035474  116000  177776                   MOVB    -2(R0),R0       ; GET TRAP CODE.
035500  010001                           MOV     R0,R1           ; WORKING COPY.
035502  162701  000017                   SUB     #17,R1
035506  100001                           BPL     1$              ; BR IF TRAP+20 OR HIGHER.
035510  005001                           CLR     R1              ; TRAP+0 THRU TRAP+17 ARE COMMON.
035512  006301                  1$:      ASL     R1              ; SHIFT UP TO WORD OFFSET...
035514  016107  035520                   MOV     2$(R1),PC       ;...AND DISPATCH THERE.

035520  035534                  2$:      CCBIT                   ; TRAP0 - TRAP17 = CC BIT CHECK.
035522  035564                           BGNTST                  ; TRAP20 = BEGIN NEW TEST.
035524  035606                           CHKSWR                  ; TRAP21 = CHECK SWR.
035526  035616                           NEWSWR                  ; TRAP22 = GET NEW SWR.
035530  036014                           LOGERR                  ; TRAP23 = LOG AND REPORT ERROR.
035532  036602                           ITOA                    ; TRAP24 = CONVERT INTEGER TO ASCII AND PRINT.
                                         ;
                                         ; ADD MORE AS NEEDED.
                                         ;

                                ;*****************************************************************************
                                ; CONDITION CODE BIT CHECKER.
                                ; CALL: TRAP+N, WHERE N IS THE CONDITION CODE TO EXPECT (0 - 17).
                                ;
                                ; PSW TO EXAMINE IS AT 6(SP), N IS IN R0.
                                ; RETURN TO CALL+2 IF CC N.E. N
                                ; RETURN TO CALL+6 OTHERWISE.
                                ;
035534  016601  000006          CCBIT:   MOV     6(SP),R1        ; GET PSW TO CHECK.
035540  042701  177760                   BIC     #^C17,R1        ; KEEP CC BITS ONLY.
035544  020001                           CMP     R0,R1
035546  001003                           BNE     1$              ; BR IF CC BITS INCORRECT.
035550  062766  000004  000004           ADD     #4,4(SP)        ; OK, ADJUST RETURN PC.
035556  012601                  1$:      MOV     (SP)+,R1
035560  012600                           MOV     (SP)+,R0
035562  000002                           RTI

                                ;*****************************************************************************
                                ; BEGIN TEST ROUTINE.
                                ; CALL VIA TRAP+20
                                ;
035564  104421                  BGNTST:  CHECKSWR
035566  012601                           MOV     (SP)+,R1        ; RESTORE CALLERS REGISTERS.
035570  012600                           MOV     (SP)+,R0
035572  017667  000000  000524           MOV     @(SP),TSTNUM    ; SAVE TEST NUMBER...
                                                                 ;...FOR THE ERROR HANDLER...
035600  062716  000002                   ADD     #2,(SP)         ;...BUMP RETURN PC...
035604  000002                           RTI                     ;...AND RETURN.
```

```
                                ;*****************************************************************************
                                ; SWITCH REGISTER HANDLER.
                                ; CALLED VIA TRAP+21 (CHECKSWR) AND TRAP+22 (GETSWR).
                                ;
                                        .ENABL  LSB
035606  104005                  CHKSWR: EMT+5                   ; CHECK KEYBOARD ( => R0 ).
035610  120027  000007                  CMPB    R0,#7           ; <^G> ??
035614  001024                          BNE     1$              ; JUST RETURN IF NOT.
                                ;
035616  005737  000042          NEWSWR: TST     @#42            ; CHAINING ??
035622  001021                          BNE     1$              ; DON'T IF SO.
035624  012700  035674                  MOV     #OSWR,R0
035630  104003                          EMT+3                   ; OLD SWR...
035632  016700  142340                  MOV     SWR,R0
035636  104424                          TRAP+24                 ;...= THIS.
035640  012700  035705                  MOV     #NSWR,R0
035644  104003                          EMT+3                   ; NEW SWR ...
035646  104000                          EMT+0                   ;...AND WAIT FOR INPUT.
035650  104011                          EMT+11                  ; PARSE OCTAL NUMBER.
035652  000405                          BR      1$              ; JUST EXIT IF NO CHANGE.
035654  120067  142316                  CMPB    R0,SWR          ; NEW SEGMENT SELECTED ??
035660  001016                          BNE     NEWSEG          ; BR IF SO.
035662  010067  142310                  MOV     R0,SWR          ; OTHERWISE, SAVE NEW SWR...
035666  012601                  1$:     MOV     (SP)+,R1        ;...RESTORE...
035670  012600                          MOV     (SP)+,R0
035672  000002                          RTI                     ;...AND RETURN TO CALLER.
                                        .DSABL  LSB
035674      015     012     123 OSWR:   .ASCIZ  <15><12>'SWR = '
035677      127     122     040
035702      075     040     000
035705      040     040     116 NSWR:   .ASCIZ  '  NEW = '
035710      105     127     040
035713      075     040     000
                                        .EVEN
                                ;
                                ; NEW SEGMENT REQUIRED.  GET ADDRESS AND DO A RESTART-IN-SEGMENT.
                                ;
035716  010067  142254          NEWSEG: MOV     R0,SWR          ; SAVE NEW SWR.
035722  042700  177770                  BIC     #^C7,R0         ; STRIP OFF SEGMENT NUMBER.
035726  006300                          ASL     R0              ; SHIFT NUMBER UP TO WORD OFFSET...
035730  016067  035772  000054        MOV     3$(R0),INSEG    ;...AND SET/CLEAR "INSEG" FLAG.
035736  012700  004000                MOV     #ERRCNT,R0
035742  012701  002000                MOV     #1024.,R1
035746  005020                  1$:     CLR     (R0)+           ; CLEAR THE ERROR TABLE AND COUNTERS.
035750  077102                          SOB     R1,1$
035752  016700  000034                MOV     INSEG,R0        ; GET LOOP-SEGMENT ADDRESS.
035756  001002                          BNE     2$
035760  012700  010336                MOV     #DOALL,R0       ; IF ZERO, RESTORE "RUN-ALL" MODE.
035764  012706  003776          2$:     MOV     #STACK,SP       ; NOW, RESET STACK...
035770  000110                          JMP     (R0)            ;...AND RESTART-IN-SEGMENT.
035772  000000                  3$:     0                       ; SEGMENT = 0 = RUN 'EM ALL...
035774  010336  024030  027052        S1,S2,S3                ;...OTHERWISE, RUN JUST ONE OF THESE.
036002  027746  030112  030142        S4,S5,S6
036010  033640                        S7
036012  000000          INSEG:  0                               ; ZERO = RUN ALL...
                                                                ;...NZ = ADDRESS FOR LOOP-IN-SEGMENT.
```

```
                                        ;*****************************************************************************
                                        ; ERROR HANDLER.
                                        ; CALLED VIA TRAP#23.
                                        ;
                                        ; ACCUMULATE ERROR DATA (PC AND NUMBER OF OCCURRANCES) IN THE ERROR LOG.
                                        ; PRINT THE STATE OF THE MACHINE AT TIME OF ERROR (UNLESS INHIBITED).
                                        ; TEST THE LOOP (SWR<9>) AND HALT (SWR<15>) SWITCHES
                                        ; AND EXIT ACCORDINGLY.
                                        ;
036014  012700  004000          LOGERR: MOV     #ERRCNT,R0      ; GET TABLE POINTER...
036020  016601  000004                  MOV     4(SP),R1        ;...AND CURRENT (ERROR) PC.
036024  005220                          INC     (R0)+           ; BUMP TOTAL ERROR COUNT.
036026  122020                          CMPB    (R0)+,(R0)+     ; SKIP OVER TOTAL PASS COUNT...
036030  005020                          CLR     (R0)+           ;...AND CLEAR "CONSECUTIVE-GOOD" COUNT.
036032  020110                  1$:     CMP     R1,(R0)         ; THIS PC ALREADY LOGGED ??
036034  001410                          BEQ     3$              ; JUST BUMP ITS COUNT IF SO.
036036  005710                          TST     (R0)            ; NO, END OF TABLE ??
036040  001405                          BEQ     2$              ; STORE PC HERE IF SO.
036042  020027  007772                  CMP     R0,#LOGEND-4    ; TABLE FULL ??
036046  103005                          BHIS    4$              ; YES, FORGET IT !!!
036050  022020                          CMP     (R0)+,(R0)+     ; NONE OF THE ABOVE, BUMP POINTER...
036052  000767                          BR      1$              ;...AND LOOP 'TIL WE FIND ONE...
                                                                ;...OR THE OTHER.
036054  010110                  2$:     MOV     R1,(R0)         ; STORE ERROR PC...
036056  005260  000002          3$:     INC     2(R0)           ;...AND BUMP ITS COUNTER...
036062                          4$:                             ;...AND FALL THRU.

036062  032767  020000  142106  PNTERR: BIT     #20000,SWR      ; ERROR MESSAGES INHIBITED ??
036070  001062                          BNE     1$              ; BR IF SO.
036072  010667  000224                  MOV     SP,ESP          ; GET STACK POINTER...
036076  062767  000010  000216          ADD     #10,ESP         ;...ADJUST TO SP AT TIME OF ERROR.
036104  104007                          EMT+7                   ; <CRLF>
036106  016700  000214                  MOV     SEGN,R0
036112  104003                          EMT+3                   ; CURRENT SEGMENT...
036114  012700  036475                  MOV     #FINTST,R0
036120  104003                          EMT+3                   ;...AND TEST...
036122  016700  000176                  MOV     TSTNUM,R0
036126  104424                          TRAP+24                 ;...NUMBER.
036130  012700  036515                  MOV     #HDR,R0
036134  104003                          EMT+3                   ;...HEADER.
036136  016600  000004                  MOV     4(SP),R0
036142  104424                          TRAP+24                 ;; PRINT 4(SP)
036144  104010                          EMT+10                  ;; <TAB>
036146  016600  000006                  MOV     6(SP),R0
036152  104424                          TRAP+24                 ;; PRINT 6(SP)
036154  104010                          EMT+10                  ;; <TAB>
036156  016600  000002                  MOV     2(SP),R0
036162  104424                          TRAP+24                 ;; PRINT 2(SP)
036164  104010                          EMT+10                  ;; <TAB>
036166  011600                          MOV     (SP),R0
036170  104424                          TRAP+24                 ;; PRINT (SP)
036172  104010                          EMT+10                  ;; <TAB>
036174  010200                          MOV     R2,R0
036176  104424                          TRAP+24                 ;; PRINT R2
036200  104010                          EMT+10                  ;; <TAB>
036202  010300                          MOV     R3,R0
036204  104424                          TRAP+24                 ;; PRINT R3
```

TRAP HANDLER

```
                036206  104010                          EMT+10                  ;; <TAB>
                036210  010400                          MOV     R4,R0
                036212  104424                          TRAP+24                 ;; PRINT R4
                036214  104010                          EMT+10                  ;; <TAB>
                036216  010500                          MOV     R5,R0
                036220  104424                          TRAP+24                 ;; PRINT R5
                036222  104010                          EMT+10                  ;; <TAB>
                036224  016700  000072                  MOV     ESP,R0
                036230  104424                          TRAP+24                 ;; PRINT ESP
                036232  104010                          EMT+10                  ;; <TAB>
     4330       036234  104007                          EMT+7                   ; <CRLF>
     4331       036236  012601                  1$:     MOV     (SP)+,R1        ; RESTORE REGISTERS.
     4332       036240  012600                          MOV     (SP)+,R0
     4333       036242  005767  141730                  TST     SWR             ; HALT-ON-ERROR ( SWR<15> ) ??
     4334       036246  100012                          BPL     2$              ; BR IF NOT.
     4335       036250  013767  177564  000042          MOV     @#177564,SAVTTY ; YES, SAVE CONSOLE TTY STATUS...
     4336       036256  042737  000004  177564          BIC     #4,@#177564     ;...AND INSURE "MAINT" IS NOT SET.
     4337       036264  000000                          EHALT                   ; ERROR HALT.
     4338       036266  016737  000026  177564          MOV     SAVTTY,@#177564 ; ON PROCEED, RESTORE TTY STATUS.
     4339       036274  032767  001000  141674  2$:     BIT     #1000,SWR       ; LOOP-ON-ERROR ( SWR<9> ) ??
     4340       036302  001003                          BNE     3$              ; BR IF SO.
     4341       036304  062716  000002                  ADD     #2,(SP)         ; NO, ADJUST RETURN PC...
     4342       036310  000002                          RTI                     ;...AND RETURN TO CALLER.
     4343       036312  017616  000000          3$:     MOV     @(SP),(SP)      ; REPLACE RETURN PC WITH LOOP ADDRESS...
     4344       036316  000002                          RTI                     ;...AND DO IT.
     4345
     4346       036320  000000                  SAVTTY: 0
     4347       036322  000000                  ESP:    0
     4348       036324  000000                  TSTNUM: 0
     4349       036326  036330                  SEGN:   SEG1                    ; POINTS TO ONE OF THE FOLLOWING:
     4350       036330     124     061     061  SEG1:   .ASCIZ  'T11 INSTRUCTION'
                036333     040     111     116
                036336     123     124     122
                036341     125     103     124
                036344     111     117     116
                036347     000
     4351       036350     124     061     061  SEG2:   .ASCIZ  'T11 TRAP/INTERRUPT'
                036353     040     124     122
                036356     101     120     057
                036361     111     116     124
                036364     105     122     122
                036367     125     120     124
                036372     000
     4352       036373     114     117     103  SEG3:   .ASCIZ  'LOCAL RAM'
                036376     101     114     040
                036401     122     101     115
                036404     000
     4353       036405     114     117     103  SEG4:   .ASCIZ  'LOCAL ROM'
                036410     101     114     040
                036413     122     117     115
                036416     000
     4354       036417     123     105     122  SEG5:   .ASCIZ  'SERIAL LINE 1'
                036422     111     101     114
                036425     040     114     111
                036430     116     105     040
                036433     061     000
     4355       036435     123     105     122  SEG6:   .ASCIZ  'SERIAL LINE 2'
```

```
                036440  111     101     114
                036443  040     114     111
                036446  116     105     040
                036451  062     000
    4356        036453  120     101     122     SEG7:   .ASCIZ  'PARALLEL I/O PORT'
                036456  101     114     114
                036461  105     114     040
                036464  111     057     117
                036467  040     120     117
                036472  122     124     000
    4357        036475  040     106     101     FINTST: .ASCIZ  ' FAILS IN TEST '
                036500  111     114     123
                036503  040     111     116
                036506  040     124     105
                036511  123     124     040
                036514  000
    4358        036515  015     012     040     HDR:    .ASCII  <15><12>' PC+2'<11>
                036520  120     103     053
                036523  062     011
    4359        036525  040     040     120             .ASCII  '  PSW'<11>
                036530  123     127     011
    4360        036533  040     040     122             .ASCII  '  R0'<11>
                036536  060     011
    4361        036540  040     040     122             .ASCII  '  R1'<11>
                036543  061     011
    4362        036545  040     040     122             .ASCII  '  R2'<11>
                036550  062     011
    4363        036552  040     040     122             .ASCII  '  R3'<11>
                036555  063     011
    4364        036557  040     040     122             .ASCII  '  R4'<11>
                036562  064     011
    4365        036564  040     040     122             .ASCII  '  R5'<11>
                036567  065     011
    4366        036571  122     066     050             .ASCIZ  'R6(SP)'<15><12>
                036574  123     120     051
                036577  015     012     000
    4367                                                .EVEN
```

```
                                ;*****************************************************************************
                                ; CONVERT INTEGER TO ASCII AND PRINT.
                                ; CALL VIA TRAP#24, NUMBER TO CONVERT IS 2ND WORD ON THE STACK.
                                ;
                                ; CODE ADDED BECAUSE EMT#30 (INTEGER TO ASCII) FUNCTIONALITY IS
                                ; DIFFERENT BETWEEN XXDP# V1.0 AND V1.1.  THIS ROUTINE MAKES THAT
                                ; DIFFERENCE TRANSPARENT TO THE CALLER.
                                ;
036602  016600  000002        ITOA:   MOV     2(SP),R0        ; NUMBER => R0.
036606  012701  036640                MOV     #2$,R1          ; IF V1.1, PACK THE ASCII STRING AT 2$.
036612  005011                        CLR     (R1)
036614  104030                        EMT#30                  ; CONVERT I TO A.
036616  005767  000016                TST     2$              ;
036622  001403                        BEQ     1$              ; IF V1.0, IT'S ALREADY BEEN PRINTED.
036624  012700  036640                MOV     #2$,R0          ; IF V1.1, PRINT IT NOW.
036630  104003                        EMT#3
036632  012601                1$:     MOV     (SP)#,R1        ; RESTORE...
036634  012600                        MOV     (SP)#,R0
036636  000002                        RTI                     ;...AND RETURN.

036640      061     062     063 2$:     .ASCIZ  /123456/
036643      064     065     066
036646      000
                                        .EVEN

036650  036650                $LSTAD: .

                                        .END    BEGIN
```

CNKXAB -- KXT-11 (FALCON(-PLUS) MACRO M1200 02-DEC-83 11:28 PAGE 16-1

SEQ 0097

SYMBOL TABLE

```
ADCB0   012536  EHALT = 000000  MODE0   015224  QBXCSR= 174020  SLU2P = 000240
ADCB1   020146  EMTTRP 010140  MODE1   015270  QBXIN   035410  SLU2V = 000120
ADC0    014142  EMTVEC= 000030  MODE2   015364  QBXVEC= 000110  SOB1    023354
ADC1    022122  ENDPAS 035076  MODE3   015506  RAMAD1= 160010  SPD0  = 001016
ADD0    014656  ENDSEG 035056  MODE4   015664  RAMAD2= 100000  SPD1  = 001020
ADD1    022672  EOP    035256  MODE5   016112  RAMSZ1= 003764  SPD2  = 001022
ADR   = 001000  EOPE   035303  MODE6   016322  RAMSZ2= 024000  SPD3  = 001024
ADR1  = 001002  EOP1   035267  MODE7   016456  RBUF  = 001046  SPD4  = 001026
ADR2  = 001004  EPCTBL= 004006  MOV0    013010  RCSR  = 001044  SPD5  = 001030
ALLCC   010604  ERHEAD 035366  MOV1    020430  RCVACT= 004000  SPD6  = 001032
ASLB0   012364  ERRCNT= 004000  NBIT    010434  RCVBRK= 004000  SPD7  = 001034
ASLB1   017760  ERROR = 104423  NEGB0   012144  RCVDUN= 000200  STACK = 003776
ASL0    013770  ERRSUM 035324  NEGB1   017524  RCVERR= 100000  START = 172000
ASL1    021734  ERRVEC= 000004  NEG0    013550  RCVIE = 000100  SUB0    015004
ASRB0   012436  ESP    036322  NEG1    021464  RCVRS   032140  SUB1    023156
ASRB1   020036  FINTST 036475  NEWSEG  035716  REGS    010630  SWAB0   014466
ASR0    014042  FLT0   030352  NEWSWR  035616  RESRVD= 000010  SWAB1   022462
ASR1    022012  FLT1   030342  NOBIT   010374  RESTAR= 172004  SWR   = 000176
BADAI   034002  FPLUS  027062  NOP1  = 000260  RESVEC= 000010  SXT0    014414
BADBI   033772  FRERR = 020000  NSWR    035705  ROLB0   012224  SXT1    022404
BADPI0  033762  GETSWR= 104422  ODT   = 172000  ROLB1   017610  SYNC    030340
BADRI   030320  GPASS = 004004  ODTBRK= 170000  ROL0    013630  S1      010336
BADTI   030330  HDR    036515  ODTIN = 000140  ROL1    021560  S2      024030
BADU1   030300  HELLO  010300  ODTRE = 172004  ROMAD1= 170000  S3      027052
BADU2   030310  INCB0  011714  ORERR = 040000  ROMAD2= 164000  S4      027746
BEGIN   010000  INCB1  017214  OSWR    035674  ROMSZ1= 002000  S5      030112
BEGINT= 104420  INC0   013320  OUTENA= 000004  ROMSZ2= 004000  S6      030142
BEVNT = 000100  INC1   021212  OUTINT= 000001  RORB0   012302  S7      033640
BGNTST  035564  INENA = 000020  OUTRDY= 000002  RORB1   017672  TEMP  = 001006
BHALT = 000140  ININT = 000010  PIOA  = 176200  RORO    013706  TEMP1 = 001010
BICB0   011610  INIT   010000  PIOAV = 000134  ROR1    021646  TEMP2 = 001012
BICB1   017006  INIT1  010072  PIOB  = 176202  RVEC  = 001054  TEMP3 = 001014
BIC0    013162  INRDY = 000040  PIOBV = 000130  R6    =%000006  TN$   = 000147
BIC1    020744  INSEG  036012  PIOC  = 176204  R6TST   024042  TPASS = 004002
BPTVEC= 000014  IOTVEC= 000020  PIODUN  035042  R7    =%000007  TPSW    015126
BRANCH  010350  ITOA   036602  PIOMC = 176206  SAVTTY  036320  TPSW1   023454
BTWRD   023572  JMP1   010712  PIOP  = 000240  SBCB0   012650  TRAPHA  035464
CBIT    010514  JMP2   010752  PIOTST  034012  SBCB1   020264  TRAP4 = 167776
CCBIT   035534  JMP3   011006  PNTERR  036062  SBC0    014254  TRPVEC= 000034
CHECKC= 104400  JMP4   011056  PPA   = 001060  SBC1    022240  TRTVEC= 000014
CHECKS= 104421  JMP5   011112  PPAV  = 001070  SEGN    036326  TSTB0   011422
CHKSWR  035606  JMP6   011162  PPB   = 001062  SEG1    036330  TSTB1   016604
CLNZ  = 000254  JMP7   011234  PPBV  = 001072  SEG2    036350  TSTNUM  036324
CMPB0   011476  JSRTST 011322  PPC   = 001064  SEG3    036373  T1      010344
CMPB1   016722  KXT11 = 000004  PPCW  = 001066  SEG4    036405  T10     011316
CMP0    013104  K0    = 001016  PPM0  = 000233  SEG5    036417  T100    025646
CMP1    020516  K1    = 001020  PPM1  = 000264  SEG6    036435  T101    025764
COMB0   012052  K2    = 001022  PRI0  = 000000  SEG7    036453  T102    026102
COMB1   017414  K3    = 001024  PRI1  = 000040  SENVC = 000273  T103    026276
COM0    013456  K4    = 001026  PRI2  = 000100  SEVC  = 000263  T104    026376
COM1    021362  K5    = 001030  PRI3  = 000140  SKP1  = 000401  T105    026460
DFPBR = 001040  K6    = 001032  PRI4  = 000200  SKP2  = 000402  T106    026526
DFSPD = 001042  K7    = 001034  PRI5  = 000240  SKP3  = 000403  T107    026614
DLDUN   033600  LEDBIT= 000200  PRI6  = 000300  SLU1  = 177560  T11     011416
DLSET   030170  LOGEND= 007776  PRI7  = 000340  SLU1P = 000200  T110    026742
DLTSTS  030364  LOGERR  036014  PWRVEC= 000024  SLU1V = 000060  T111    027200
DOALL   010336  MFPT  = 000007  QBXA  = 001036  SLU2  = 176540  T112    027316
```

```
CNKXAB -- KXT-11 (FALCON(-PLUS) MACRO M1200  02-DEC-83 11:28  PAGE 16-2
                                                                SEQ 0098
SYMBOL TABLE

T113    027506          T136    032774          T25     013452          T47     017410          T70     024210
T114    030012          T137    033122          T26     013624          T5      010746          T71     024466
T115    030170          T14     011710          T27     013764          T50     017604          T72     024652
T116    030364          T140    033316          T3      010624          T51     017754          T73     024750
T117    030536          T141    033432          T30     014136          T52     020142          T74     025122
T12     011472          T142    033646          T31     014410          T53     020424          T75     025240
T120    030652          T143    034012          T32     014652          T54     020512          T76     025356
T121    030710          T144    034046          T33     015122          T55     020740          T77     025474
T122    031002          T145    034234          T34     015220          T56     021206          VBIT    010462
T123    031064          T146    034436          T35     015360          T57     021356          XBUF  = 001052
T124    031146          T147    034650          T36     015502          T6      011052          XCSR  = 001050
T125    031230          T15     012046          T37     015660          T60     021554          XMTBRK= 000001
T126    031312          T16     012220          T4      010706          T61     021730          XMTIE = 000100
T127    031374          T17     012360          T40     016106          T62     022116          XMTRDY= 000200
T13     011604          T2      010600          T41     016316          T63     022400          XMTRS   030652
T130    031466          T20     012532          T42     016452          T64     022666          XOR0    014552
T131    031560          T21     013004          T43     016600          T65     023450          XOR1    022554
T132    031734          T22     013100          T44     016716          T66     023566          XVEC  = 001056
T133    032140          T23     013156          T45     017002          T67     024036          ZBIT    010544
T134    032222          T24     013314          T46     017210          T7      011156          $LSTAD  036650
T135    032434

. ABS.  036652      000
        000000      001
ERRORS DETECTED:  3

VIRTUAL MEMORY USED:  8608 WORDS  ( 34 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:04:28
CNKXAB.BIC,CNKXAB.LST/-SP=CNKXAB.P11
```